Sparse Approximate LU Decompositions

Thomas Huckle + Jürgen Bräckle

TUM

19.6.2015

Overview

1. Parallel ILU
2. SAI and MSPAI
3. Similarities of ILU and SAI → Merge
4. Parallel triangular solves

ILU Parallel for Sparse Matrices

Meijerink, van der Vorst 1977

Problem: $Ax=b$ iteratively, A sparse.

Preconditioning: $P^{-1}Ax=P^{-1}b$ with P easy and $P \approx A$

ILU classical: Reduce Gaussian Elimination to pattern of A

Incomplete LU factorization = ILU

```
FOR k=1,...,n-1 DO
  FOR j=k+1,...,n DO
    FOR  i=k+1,...,n DO
         l(j,k) = a(j,k)/a(k,k);
         a(j,i) = a(j,i) - l(j,k)a(k,i);
    ENDFOR
  ENDFOR
ENDFOR
```

```
FOR k=1,...,n-1 DO
   FOR j>k in pattern of A(:,k) DO
       FOR i>k in pattern of A(i,:) DO
            l(j,k) = a(j,k)/a(k,k);
            a(j,i) = a(j,i) - l(j,k)a(k,i);
       ENDFOR
   ENDFOR
ENDFOR
```

Good preconditioning, but fully sequential. Furthermore: $Lx=b$.

New parallel Approach

Chow, Patel, 2015

$$(L \cdot U)_{i,j} = A_{i,j}, \qquad (i,j) \in P(A), \quad P(L), P(U) \subset P(A)$$

Nonlinear system of equations for nonzero entries
in L and U at allowed positions.

Everything reduced e.g. to pattern of A, with
- L lower triangular matrix with main diagonal 1 and
- U upper triangular matrix.

Consider as fixed point problem, solve via **A**lternating **L**inear **S**ystems:
- Choose U, gives linear system for computing new L.
- Computed L gives new linear system for U

etc.

Parallel Solution of the Fixpoint Iteration

$$l_{i,j} = \frac{1}{\boxed{u_{jj}}}\left(a_{i,j} - \sum_{k=1}^{j-1} l_{i,k}\boxed{u_{k,j}}\right) \qquad sparse, \quad (i,j) \in P(L)$$

$$u_{i,j} = a_{i,j} - \sum_{k=1}^{i-1} \boxed{l_{i,k}} u_{k,j} \qquad sparse, \quad (i,j) \in P(U)$$

Ordering important! Jacobi- or Gauss-Seidel-like!

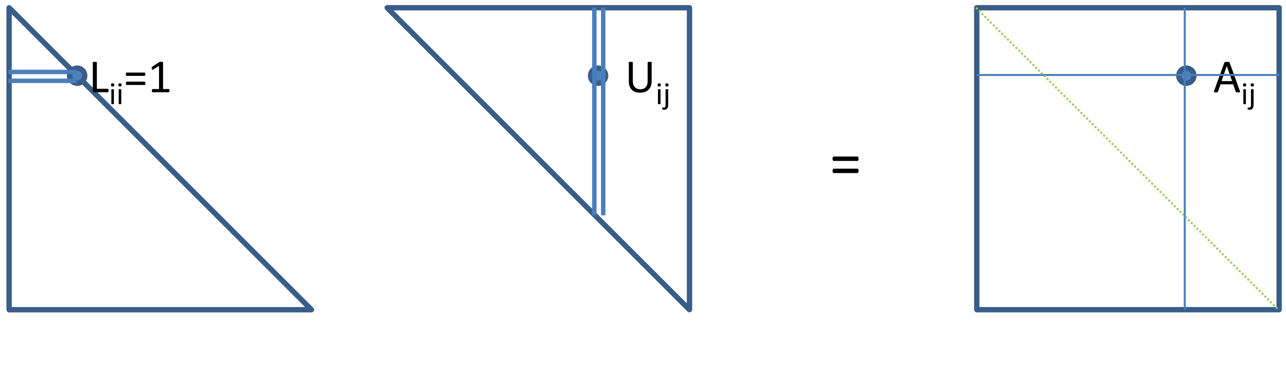
Extreme cases: like Gauss-Seidel, always newest information

→ ILU

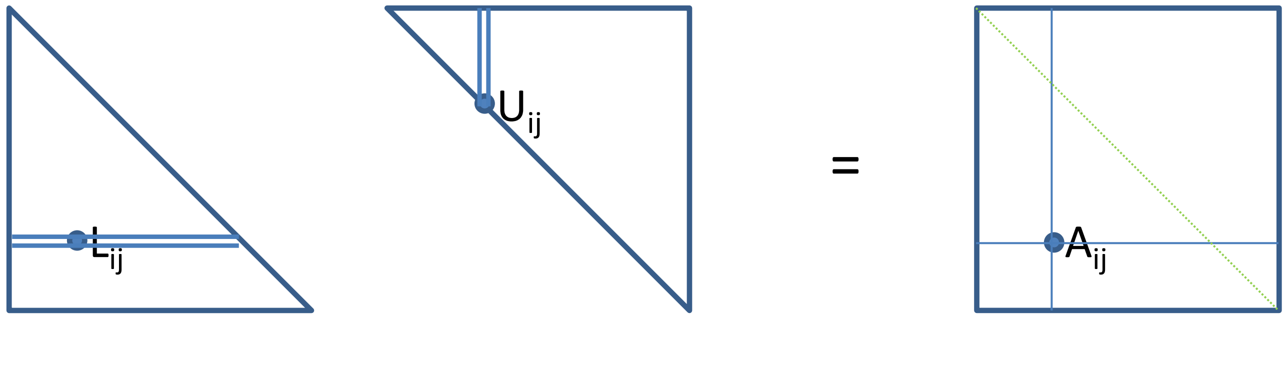
like Jacobi, always old information

→ fully parallel

Linear Equations:



U_{ij} equation
for $i <= j$



L_{ij} equation
for $i > j$

Parallel ILU

Set initial values for L and U (e.g. via Gauss-Seidel L and U)
For sweep 1,2,…. until convergence
    parallel for (i,j) ϵ S
        if i > j then

$$l_{i,j} = \frac{1}{u_{jj}}\left(a_{i,j} - \sum_{k=1}^{j-1} l_{i,k} u_{k,j} \right)$$

        else

$$u_{i,j} = a_{i,j} - \sum_{k=1}^{i-1} l_{i,k} u_{k,j}$$

        endif
    endfor
endfor

p independent threads, distribute S on p threads.
Always using new information if available.

2^{nd} Problem: Triangular Solves

Solve Lx=b, again strongly sequential!

Resort: Iterative solver, Jacobi:

$$L = D + L_0 :$$

$$b = Lx = (D + L_0)x \Rightarrow x^{k+1} = D^{-1}b - D^{-1}L_0 x^k$$

$$x^{k+1} = x^k - D^{-1}(b - Dx^k - L_0 x^k) = x^k - D^{-1}(b - Lx^k) =$$

$$= x^k - D^{-1} r_k$$

Ev. bad (no or very slow) convergence!

Associated unavoidable discrepancy for iterative solvers:

ILU well-conditioned → LU bad preconditioner!

ILU ill-conditioned → Jacobi for Lx=b or Uy=c shows slow convergence!

SAI = **S**parse **A**pproximate **I**nverse

Benson, Frederickson, Kolotilina, Yeremin, Grote, H.,Chow,…

Demands for Preconditioner M:

(i) Good parallel setup of M

(ii) Good parallel application of M

(iii) Faster convergence, $M \approx A^{-1}$

Ansatz: min || A*M – I || with respect to some norm ||.||

$$\min_G \|GA - I\|_W = \min tr\left((GA - I)W(GA - I)^T\right)$$

A spd: $W = A^{-1}$: $\quad (GA)_{i,j} = I_{i,j}, \quad (i, j) \in S(A)$

Factorized SAI: $\quad (LA)_{i,j} = L_{A,i,j}, \quad (i, j) \in S(L)$

SAI

General Ansatz:

$$\min_M \left\| AM - I \right\|_F^2 = \sum_{k=1}^{n} \min \left\| AM_k - e_k \right\|_2^2$$

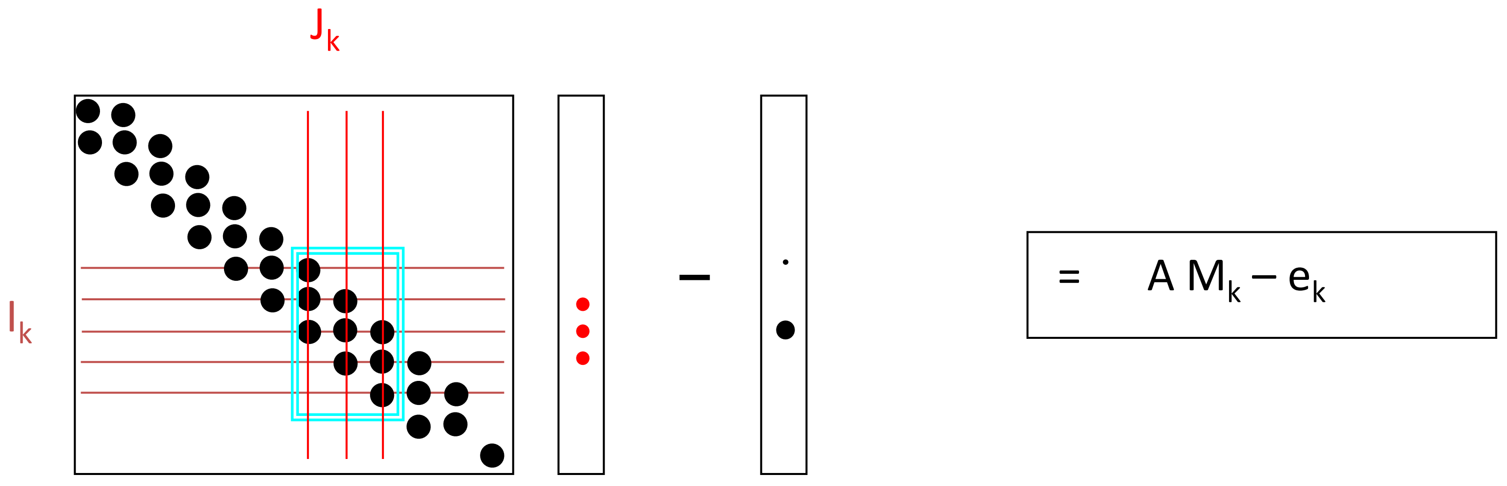
Allows parallel columnwise computation of M_k.

Solve n independent Least Squares Problems

$$\min \left\| A(I_k, J_k) M_k(J_k) - e_k(I_k) \right\|$$

On reduced index set I_k, J_k because of sparsity.

Computing M_k



Delete superfluous zeros in Least Squares Problem:

For index set J_k in M_k keep only $A(: , J_k)$

In $A(: , J_k)$ keep only nonzero rows $A(I_k , J_k)$

Solve small Least Squares problem in $A(I_k , J_k)$,
e.g. by QR-decomposition, Householder method.

Generalization

MSPAI, including target matrix (Wathen, Holland, Shaw 2005) and probing (Kallischko,H. 2007):

$$\min_M \|AM - P\|_F^2 \qquad\qquad \min_M \left\| \begin{pmatrix} A \\ \rho F \end{pmatrix} M - \begin{pmatrix} I \\ \rho E \end{pmatrix} \right\|_F^2$$

e.g. with probing conditions E·A = F → F · A^{-1} ≈ E → F· M ≈ E.
On certain subspace E the preconditioner should be very good!

More general:

$$\min_M \|BM - C\|_F^2$$ with general sparse (rectangular) B and C.

Useful feature for smoothing in MG,
seminorm in regularization,
preconditioning of Schur complement or
dense matrices

MSPAI Application

$$\min_M \left\| \begin{pmatrix} A \\ \rho v^T A \end{pmatrix} M - \begin{pmatrix} I \\ \rho v^T \end{pmatrix} \right\|_F$$

$v^T=(1,-1,1,-1,\ldots,)$ for smoother M

$v^T=(1,1,1,1,\ldots,)$ for regularizing preconditioner M

Preconditioning implicit A with sparse M:
Use sparse approximation $A_{sparse} \approx A$.
Improve A_{sparse} via

$$\min_M \left\| \begin{pmatrix} I \\ \rho v^T \end{pmatrix} M - \begin{pmatrix} A_{sparse} \\ \rho v^T A \end{pmatrix} \right\|_F$$

MSPAI for approx. LU-Factorization

Kallischko, H. 2007

$$\min \left\| L \cdot U - A \right\|_F$$ e.g. with pattern of A

Choose Start L_0=L, z.B. Gauss-Seidel tril(A)

Compute new U, with this U new L etc.

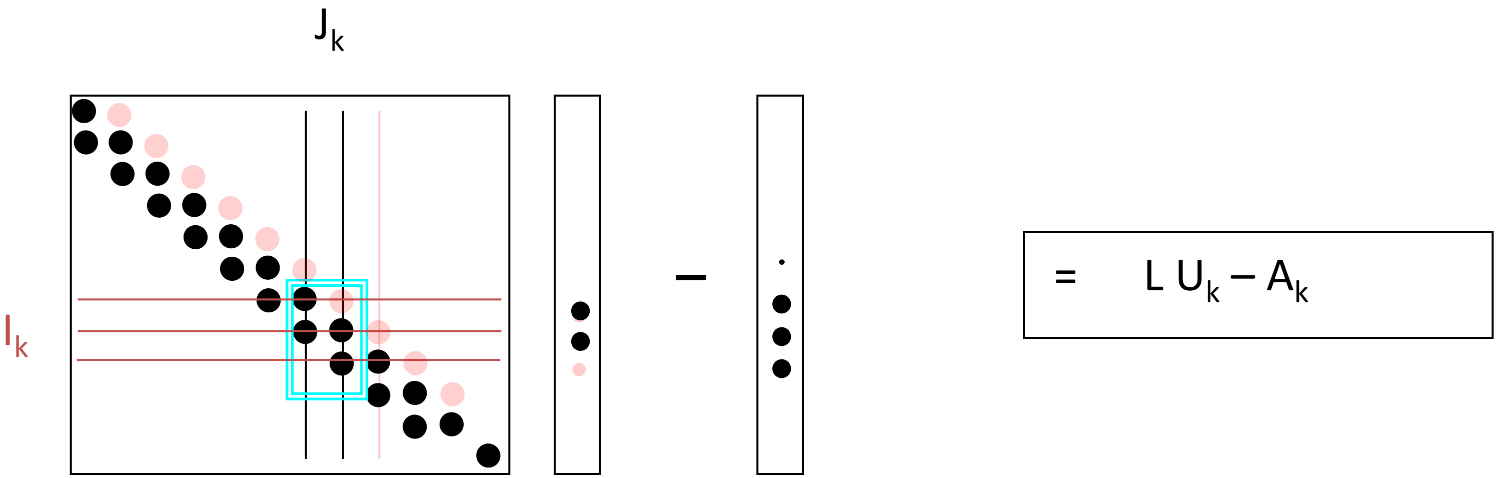
For k=1,2,...:

$$U_k = \arg\min_U \left\| L_{k-1} U - A \right\|_F$$

$$L_k = \arg\min_L \left\| L U_k - A \right\|_F$$

Advantages: - fully parallel, relatively cheap
- any pattern
- possible additional conditions (probing)
- blockwise
- symmetrisable

Computing M_k



I_k should include the pattern of A_k.

Solve small Least Squares problem in $L(I_k , J_k)U_k - A_k$,

$L(I_k,J_k)$ lower triangular!

Optimality of SALU

$$\min\left\|LU - A\right\|_F^2 = \min trace\left(\left(U^T L^T - A^T\right)\left(LU - A\right)\right) =$$

$$= \min trace\left(U^T L^T LU - 2U^T L^T A + A^T A\right) =$$

$$= \min \sum_{i=1}^{n} u_i^T L^T L u_i - 2\sum_{i=1}^{n} u_i^T L^T A + const =$$

$$= \min \sum_{i=1}^{n} u_i^T (I_i)\left(L^T(:,I_i)L(:,I_i)\right)u_i(I_i) - 2\sum_{i=1}^{n} u_i^T(I_i)L^T(:,I_i)A + const$$

Derivative = 0: $\frac{d}{d\hat{u}_i} : L^T(:,I_i)L(:,I_i)u_i(I_i) - L^T(:,I_i)A_i = 0$

$$L^T(:,I_i)\cdot\left(L(:,I_i)u_i(I_i) - A(:,i)\right) = 0$$

$$L^T(J_i,I_i)\cdot\left(L(J_i,I_i)u_i(I_i) - A(J_i,i)\right) = 0$$

Similarly for d/dL.

I_i and J_i are nonzero indices

Criteria

$$\left[L^T \cdot (LU - A)\right](i, j) = 0 \qquad \textit{für} \quad U(i, j) \neq 0$$
$$\left[(LU - A) \cdot U^T\right](i, j) = 0 \qquad \textit{für} \quad L(i, j) \neq 0$$

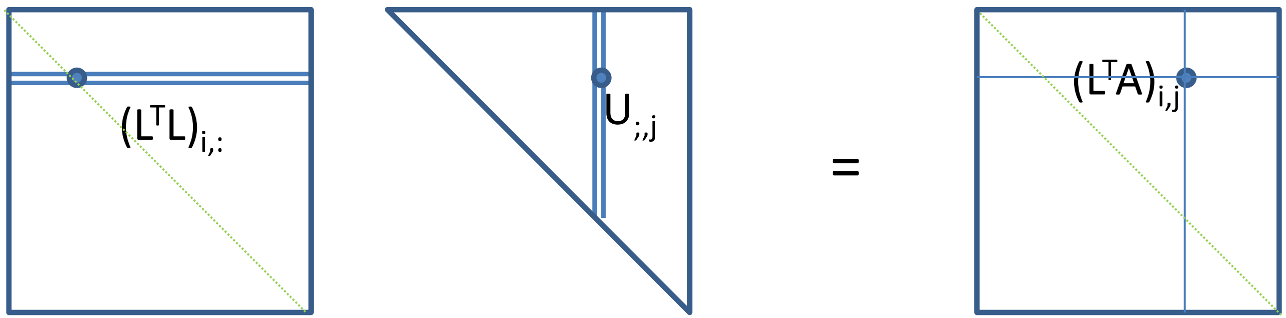
instead of

$$\left[LU - A\right](i, j) = 0 \qquad \textit{für} \quad A(i, j) \neq 0$$

$$\left(L^T LU\right)_{i,j} = \left(L^T A\right)_{i,j} \qquad \textit{für} \quad U(i, j) \neq 0 \qquad \Rightarrow U$$
$$\left(UU^T L^T\right)_{i,j} = \left(UA^T\right)_{i,j} \qquad \textit{für} \quad L(i, j) \neq 0 \qquad \Rightarrow L$$

$$\left(L^T L U\right)_{ij} = \left(L^T A\right)_{ij} \qquad for \quad U_{ij} \neq 0$$

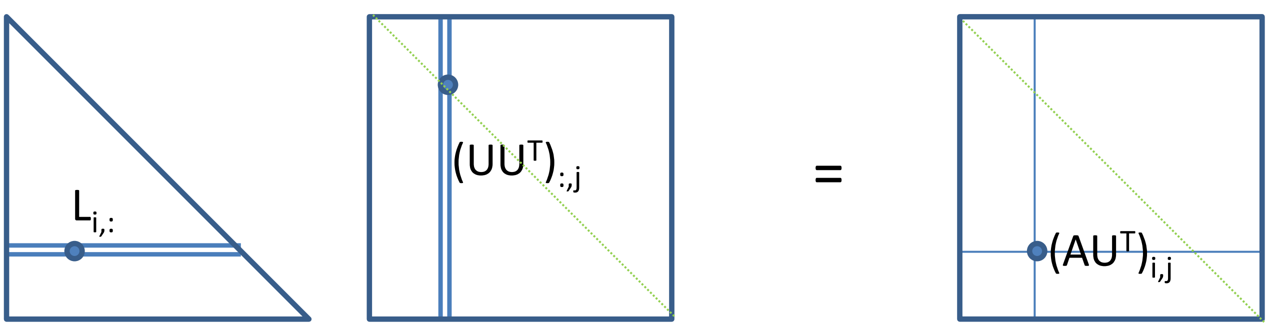
$$\left(L^T L\right)_{i,:} U_{:,j} = \left(L_{:,i}\right)^T A_{:,j} = L^T_{i,:} A_{:,j}$$



U_{ij} equation

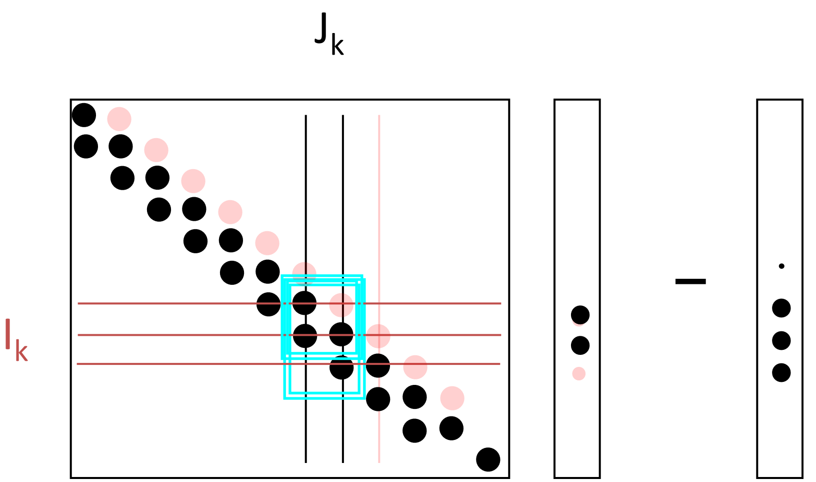
$$\left(L U U^T\right)_{ij} = \left(A U^T\right)_{ij} \qquad for \quad L_{ij} \neq 0$$

$$L_{i,:} \left(U U^T\right)_{:,j} = A_{i,:} \left(U^T\right)_{:,j} = A_{i,:} U^T_{j,:}$$



L_{ij} equation

Simplifying LS:



$$= L\,U_k - A_k$$

Instead of rectangular LS problem $L(I_k, J_k)U_k(J_k) - A_k(I_k)$

use smaller quadratic triangular nonsingular equations

$$L(J_k, J_k)U_k(J_k) - A_k(J_k)$$

like in the iterative ILU ALS.

Iterative ILU ←→ SAI

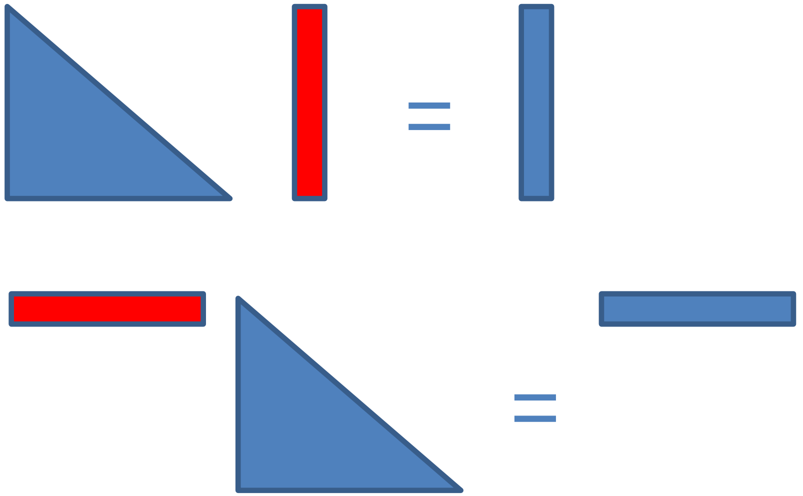
Iterate between updating
U columnwise and
L rowwise.
Special ordering/blocking/
parallelization

$$L(J_k, J_k)U_k(J_k) = A_k(J_k)$$

$$L_k(J_k)^T U(J_k, J_k) = A_k(J_k)^T$$

$$L_{J_k,J_k} U_{k,J_k} = A_{k,J_k}$$

$$L_{J_k,k} U_{J_k,J_k} = A_{J_k,k}$$



General Incomplete SAI
Combination of parallel ILU and SAI

Replace LS problem $\min \| A(I_k, J_k) M_k(J_k) - e_k(I_k) \|$

by quadratic problem $\min \| A(J_k, J_k) M_k(J_k) - e_k(J_k) \|$

if $A(J_k, J_k)$ is regular → quadratic linear system.

Special case: A triangular.

Stable version of iterative ILU by “regularizing”:

$$\min \left\| \begin{pmatrix} A(J,J) \\ \rho \cdot A(I \setminus J, J) \end{pmatrix} M_k(J) - \begin{pmatrix} e_k(J) \\ \rho e_k(I \setminus J) \end{pmatrix} \right\|$$

Further Generalizations

Inverse LU:

For k=1,2,...:

$$U_k = \arg\min_U \|(L_{k-1}A)U - I\|_F$$

$$L_k = \arg\min_L \|L(AU_k) - I\|_F$$

Mixed LU:

$$\min \|AU_{new,j} - L_{old,j}\| \rightarrow A(J,J)U_{new,j}(J) = L_{old,j}(J)$$

Symmetrized version:

$$\min \|U_{old}^T U_{new,j} - A_j\|, \qquad U = (U_{old} + U_{new})/2, \ldots$$

Matrix-free: min||AM_j-e_j|| : A(J,J)→ M_j : $$Mb = \sum_{j=1}^{n} M_j b_j$$

More Generalizations

Adding dynamic SPAI pattern finding for simulating ILUT.

Adding probing conditions like in MILU:

$$\min \left\| \begin{pmatrix} L(J,J) \\ \rho_1 \cdot L(I \setminus J, J) \\ \rho_2 v^T L \end{pmatrix} U_k(J) - \begin{pmatrix} e_k(J) \\ \rho_1 e_k(I \setminus J) \\ \rho_2 v^T A \end{pmatrix} \right\|$$

← Incomplete or LS

← probing

with $v=(1,\dots,1)^T$

Triangular Solves + Preconditioning

$$L = D + L_0 :$$

$$b = Lx = (D + L_0)x \Rightarrow x^{k+1} = D^{-1}b - D^{-1}L_0 x^k = x_k + D^{-1}r_k$$

Compute parallel triangular (incomplete) SAI Preconditioner M for L

$$Mb = MLx = (I - I + ML)x \Rightarrow$$

$$x^{k+1} = Mb - (ML - I)x^k = x^k + M\left(b - Lx^k\right)$$

$$x^{k+1} = x^k + Mr_k$$

Stationary Iteration, like Jacobi

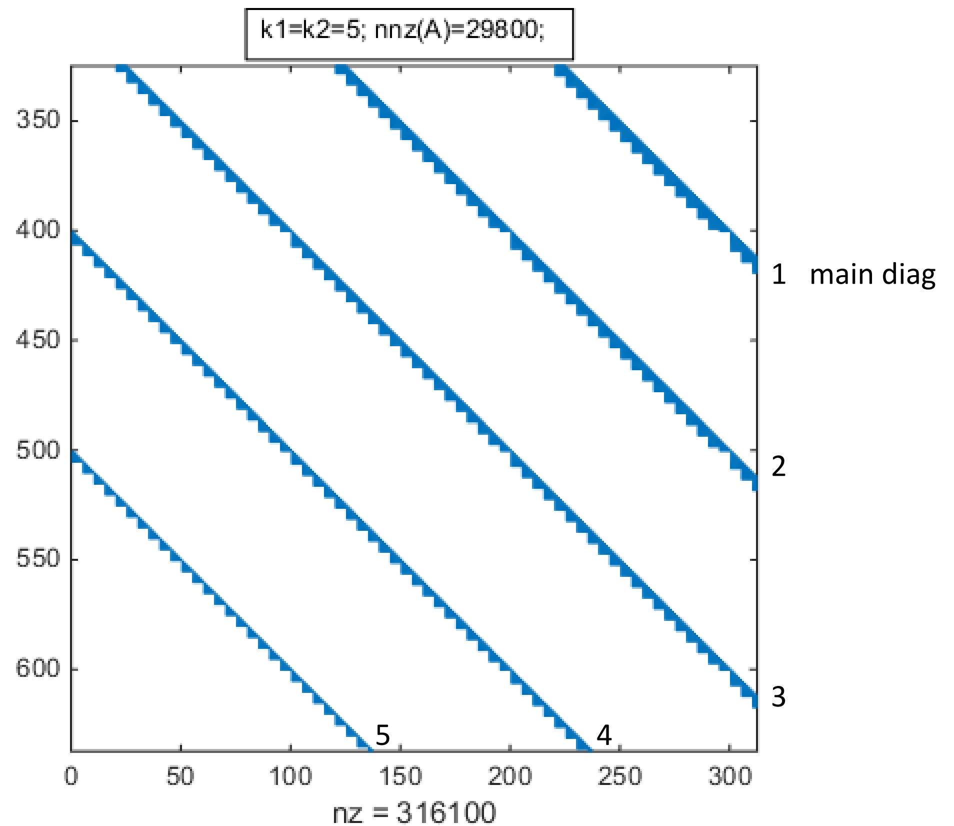
Advantages: - fully parallel

- any pattern, also blockwise, more dense
- faster convergence

Numerical example: Triangular Solve

Lower triangular part of 2D Laplace band(2,-1,….,-1,0…0)
N=100x100, iteration to relres<=10^{-12}.

Jacobi iteration or preconditioner
SAI or ISAI (=incomplete) with pattern
$|A|^{k1}$ and
blockwise k2



Iteration, Density

Jacobi: 199 iterations, density 1

SAI, k1=1	k2=1	5	10	20
It	202	179	173	169
Density	1	3.6	7	13.4

k1=2	k2=1	10	20
It	142	125	121
Dens.	2	11	20.7

k1=5	k2=1	2	5	10
It	81	82	81	76
Dens.	7	9	14.5	24

ISAI, k1=1	k2=1	5	10	20
It	100	68	59	54
Density	1	2.3	4	7.3

k1=2	k2=1	10	20
It	67	42	38
Dens.	2	6.4	11.3

k1=5	k2=1	2	5	10
It	34	33	29	24
Dens.	7	7.8	10.6	15.3

For allowed density factor around 10 the lowest iteration count 29 results for ISAI and k1=k2=5.

Comparison ILU - SALU

2D Laplacian, 5-point, 30x30

ILU	It=1	2	5
Cond	36.42	35.26	35.225
Error ichol	0.0026	7.4e-5	1.9e-9

$$(L \cdot U)_{i,j} = A_{i,j}, \qquad (i,j) \in P(A),$$

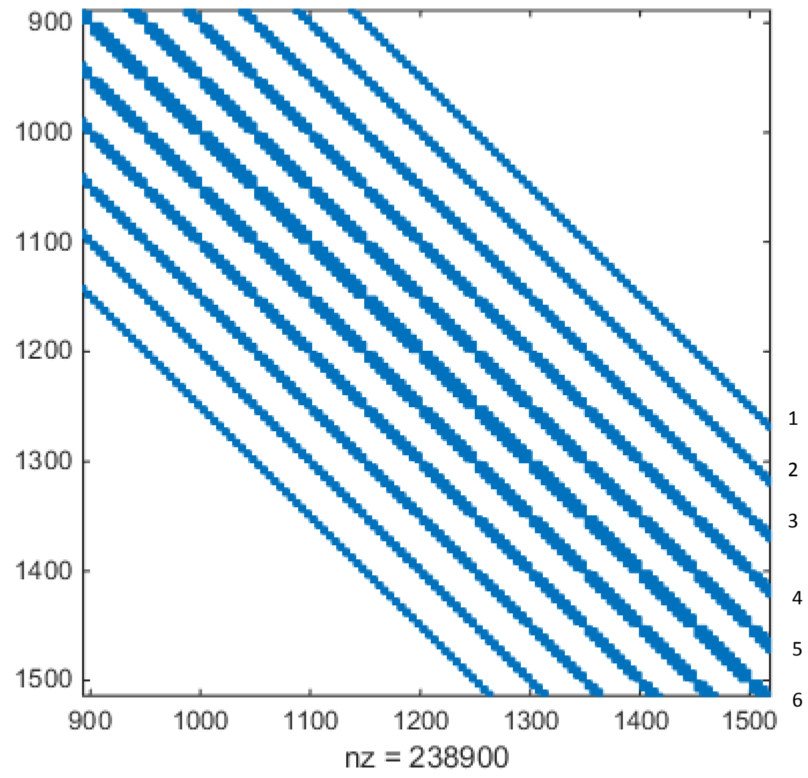
SALU	It=1	2	5
Cond	45.78	45.41	45.399
||LU-A||_frob	1.3495	1.3490	1.3490

$$\min \|L \cdot U - A\|_F$$

Numerical example: Preconditioner

2D Laplace   5-point stencil
N=50x50, condition unpreconditioned 1053, nnz(A)=12300

SAI or ISAI (=incomplete)
preconditioner
with pattern
$|A|^{k1}$ and
blockwise k2



k2=1	k1=1	2	3	4	5	10
SAI	319	181	162	69.7	50	16.4
ISAI	263	188	76	70	36	14.4
Density	12k	31k	60k	97k	142k	477k

k2=2	k1=1	2	3	4	5	10
SAI	270	145	88.9	59.9	44	15
ISAI	223	115	64	49	35	15.1
Density	20k	43k	76k	117k	166k	519k

k2=5	k1=1	2	3	4	5	10
SAI	214	108	65.6	45.1	33.9	12.4
ISAI	175	81.5	45.4	33.6	23.7	9
Density	41k	79k	124k	178k	239k	643k

k2=10	k1=1	2	3	4	5	10
SAI	193	97.6	59.2	39.8	28.6	10
ISAI	161	73.3	40	27	18.5	6.8
Density	78k	138k	204k	277k	357k	843k

For allowed density factor around 10 the lowest condition number 36 results for ISAI and k1=5,k2=1.

Comparison Modified ILU - ISALU

2D Laplacian, 5-point, 30x30, it=1

Prec.	ILU	SAI	MILU,ρ=1,ρ_1=1	MILU, ρ=0.1,ρ_1=10
Cond	36.42	45.8	34.7	25.6

$$\min \left\| \begin{pmatrix} L(J,J) \\ \rho \cdot L(I \setminus J, J) \\ \rho\rho_1 v^T L \end{pmatrix} U_k(J) - \begin{pmatrix} e_k(J) \\ \rho e_k(I \setminus J) \\ 0 \end{pmatrix} \right\|$$

← Incomplete or LS

← probing

with v=(1,…,1)^T

Conclusions

Comparing iterative parallel ILU and SAI leads to new preconditioners:

Incomplete Sparse Approximate …

Advantages:
- fully parallel
- cheaper
- include probing
- allows dynamic pattern finding
- denser preconditioner via blockwise and A^k

Problem with ILU: Solving ill-conditioned triangular systems!

A lot of open problems

End