Coarsening-based Algebraic Multi-level preconditioners

Yousef Saad
Department of Computer Science and Engineering

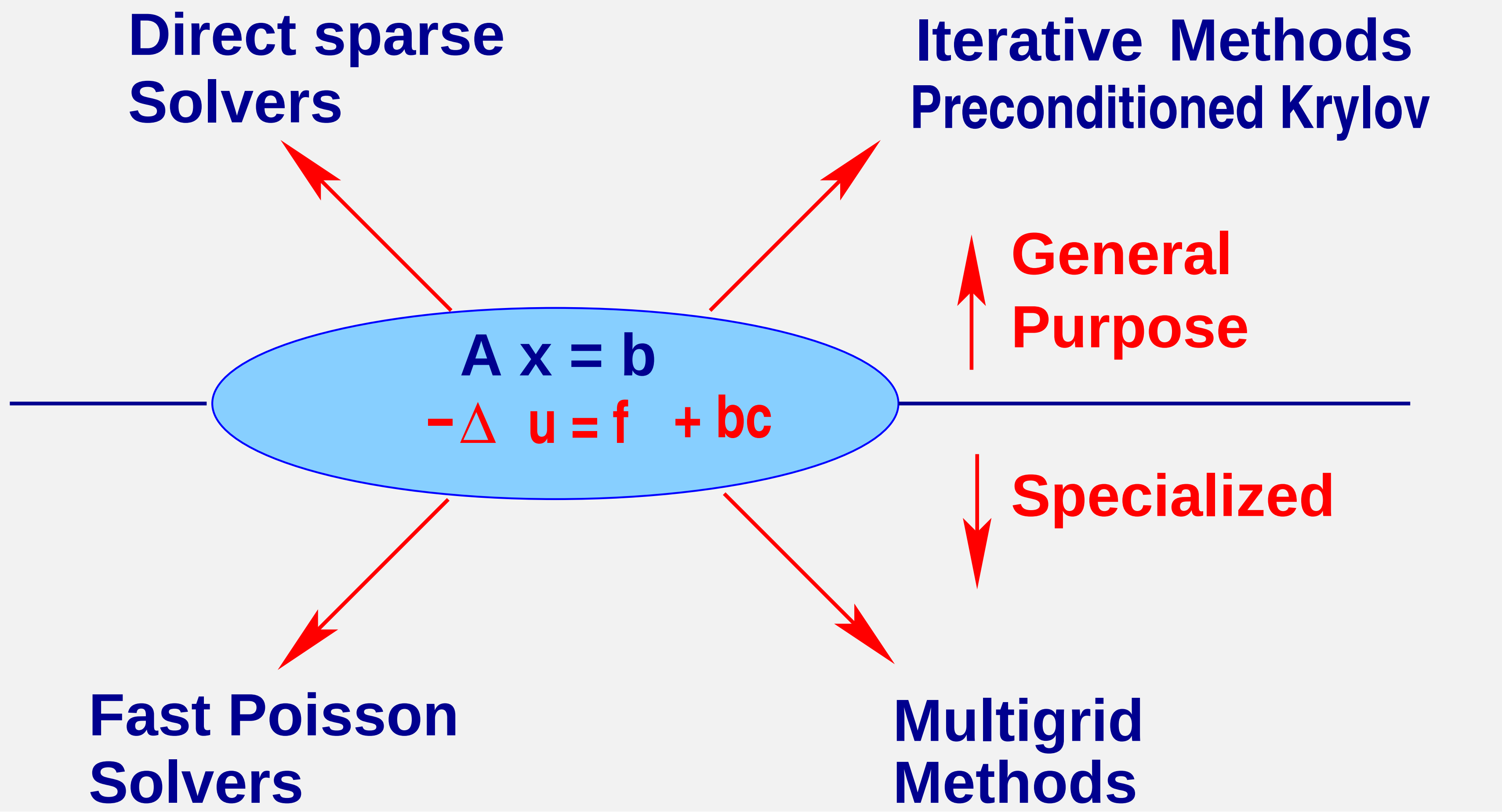
University of Minnesota

Hong-Kong, Aug 24th – 27th, 2009

Alternative Title:

Summer musings in coarse territories...

Introduction: Linear System Solvers



A few observations

- Problems are getting harder for Sparse Direct methods (more 3-D models, much bigger problems,..)
- Problems are also getting difficult for iterative methods Cause: more complex models - away from Poisson
- Researchers in iterative methods are borrowing techniques from direct methods: $\longrightarrow$ preconditioners
- The inverse is also happening: Direct methods are being adapted for use as preconditioners
- A recent trend: AMG or AMG-like, multilevel solvers of various kinds.

THE MULTILEVEL FRAMEWORK

Background: Independent sets, ILUM, ARMS

Independent set orderings permute a matrix into the form

$$\begin{pmatrix} B & F \\ E & C \end{pmatrix}$$

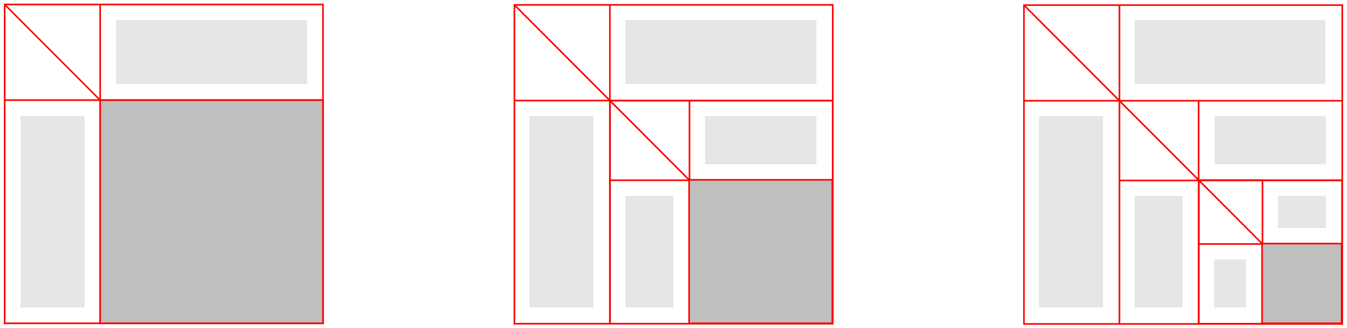
where B is a diagonal matrix.

- Unknowns associated with the B block form an independent set (IS).
- IS is maximal if it cannot be augmented by other nodes
- Finding a maximal independent set is inexpensive

Main observation: Reduced system obtained by eliminating the unknowns associated with the IS, is still sparse since its coefficient matrix is the Schur complement

$$S = C - EB^{-1}F$$

- Idea: apply IS set reduction recursively.
- When reduced system small enough solve by any method
- ILUM: ILU factorization based on this strategy. YS '92-94.

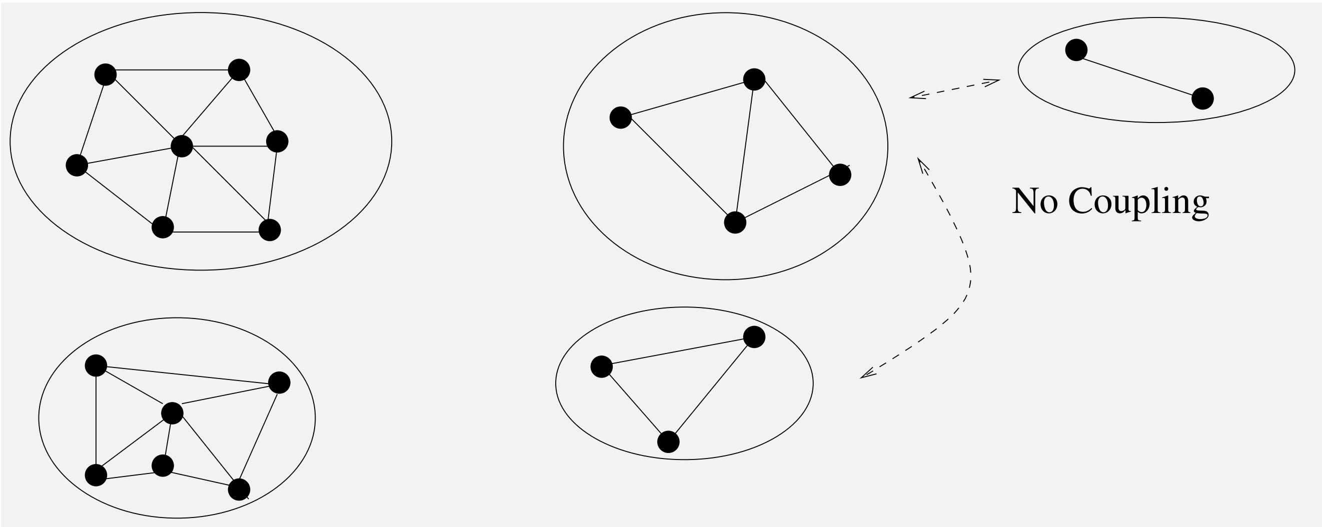


- See work by [Botta-Wubbs '96, '97, YS'94, '96, Leuze '89,..]

Group Independent Sets / Aggregates

Main goal: generalize independent sets to improve robustness

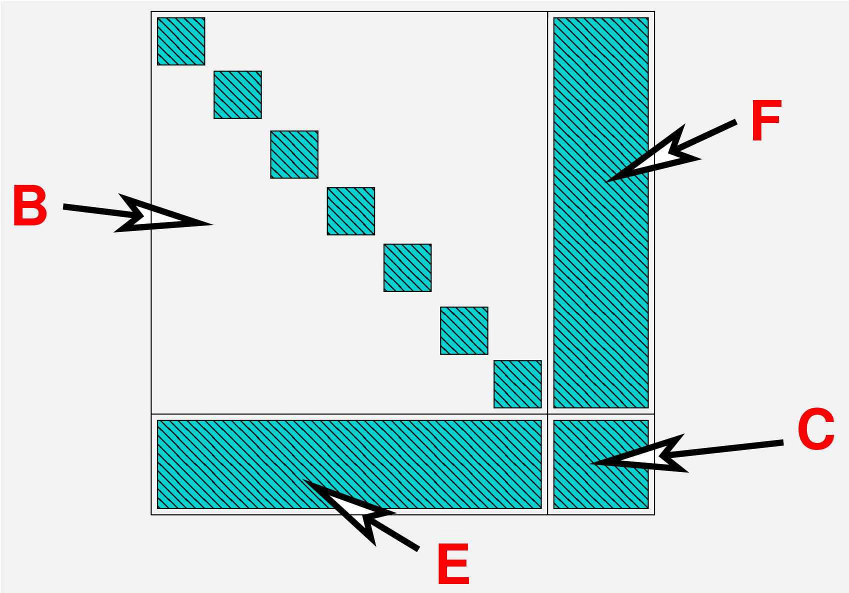
Main idea: use "cliques", or "aggregates". No coupling between the aggregates.



- Label nodes of independent sets first

Algebraic Recursive Multilevel Solver (ARMS)

➤ Typical shape of reordered matrix:



$$PAP^T = \begin{pmatrix} B & F \\ E & C \end{pmatrix} =$$

➤ Block factorize: $\begin{pmatrix} B & F \\ E & C \end{pmatrix} = \begin{pmatrix} L & 0 \\ EU^{-1} & I \end{pmatrix} \begin{pmatrix} U & L^{-1}F \\ 0 & S \end{pmatrix}$

➤ $S = C - EB^{-1}F$ = Schur complement + dropping to reduce fill

➤ Next step: treat the Schur complement recursively

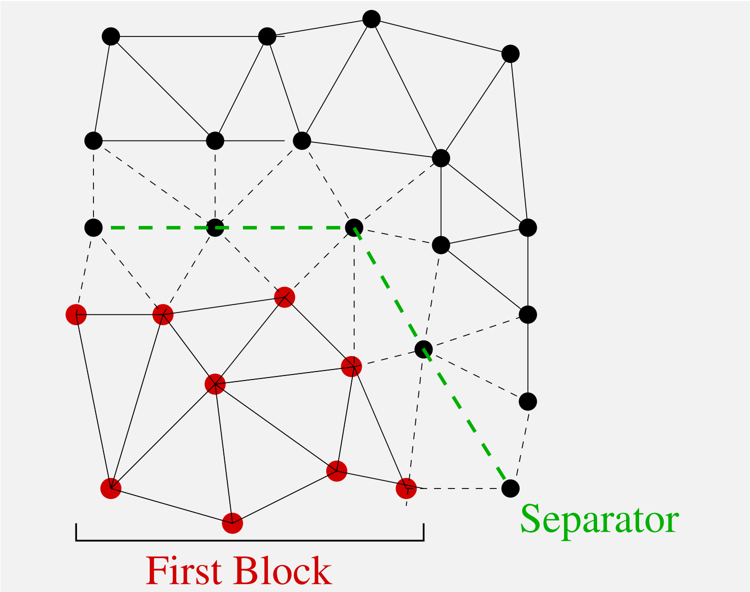
Algebraic Recursive Multilevel Solver (ARMS)

Level l Factorization:

$$\begin{pmatrix} B_l & F_l \\ E_l & C_l \end{pmatrix} \approx \begin{pmatrix} L_l & 0 \\ E_l U_l^{-1} & I \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & A_{l+1} \end{pmatrix} \begin{pmatrix} U_l & L_l^{-1} F_l \\ 0 & I \end{pmatrix}$$

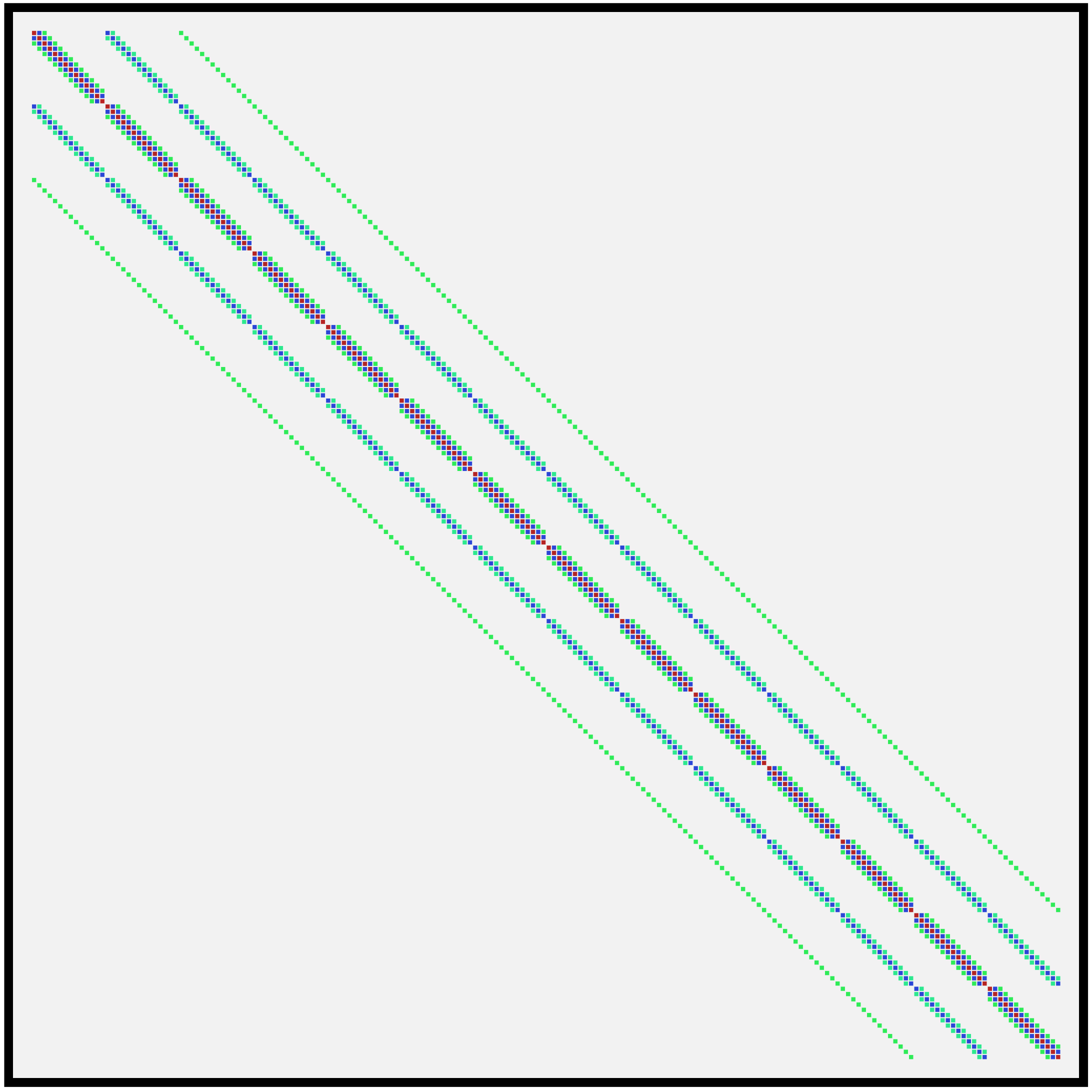
- L-solve $\sim$ restriction; U-solve $\sim$ prolongation.
- Perform above block factorization recursively on A_{l+1}

Group Independent Set reordering

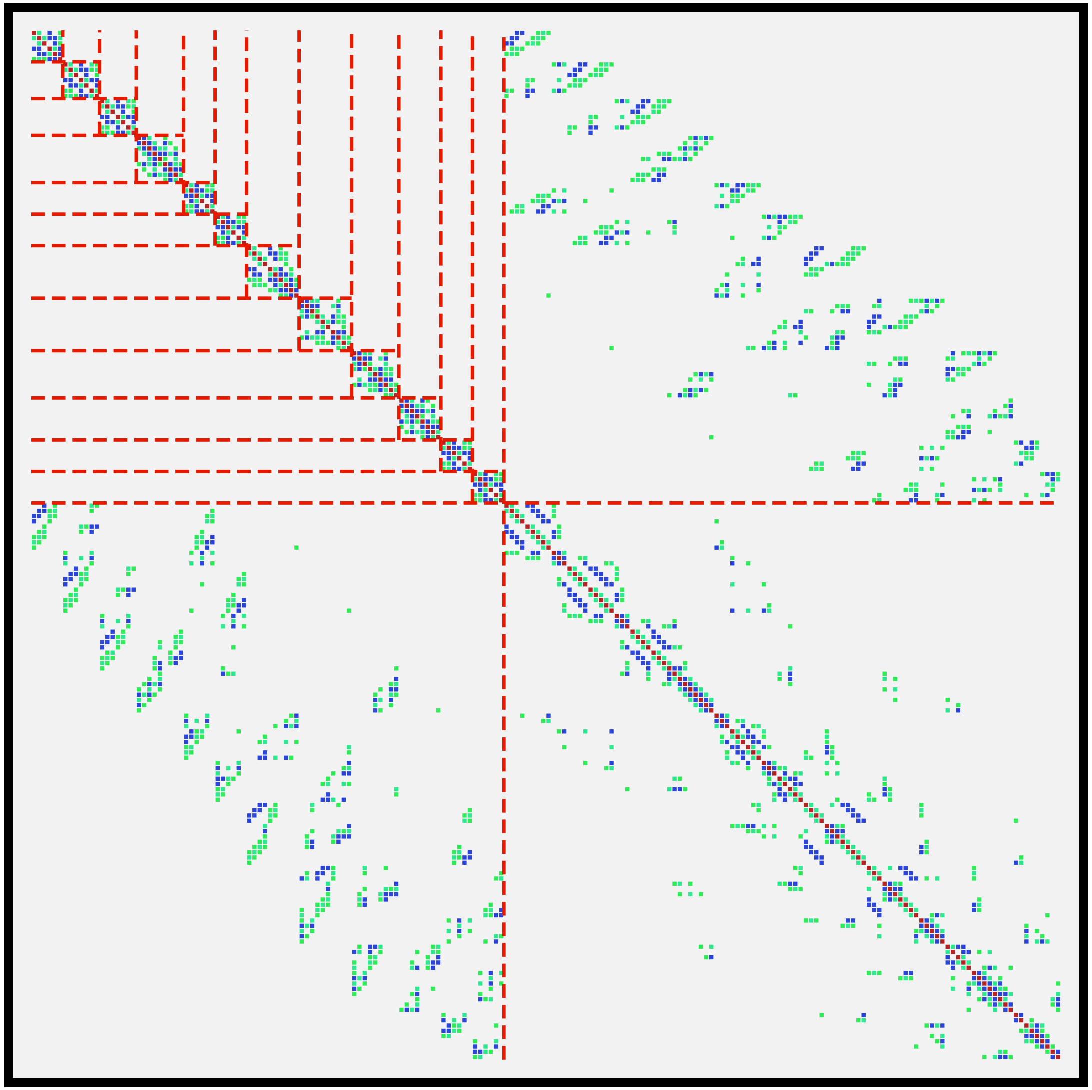


Simple strategy: Level taversal until there are enough points to form a block. Reverse ordering. Start new block from non-visited node. Continue until all points are visited. Add criterion for rejecting “not sufficiently diagonally dominant rows.”

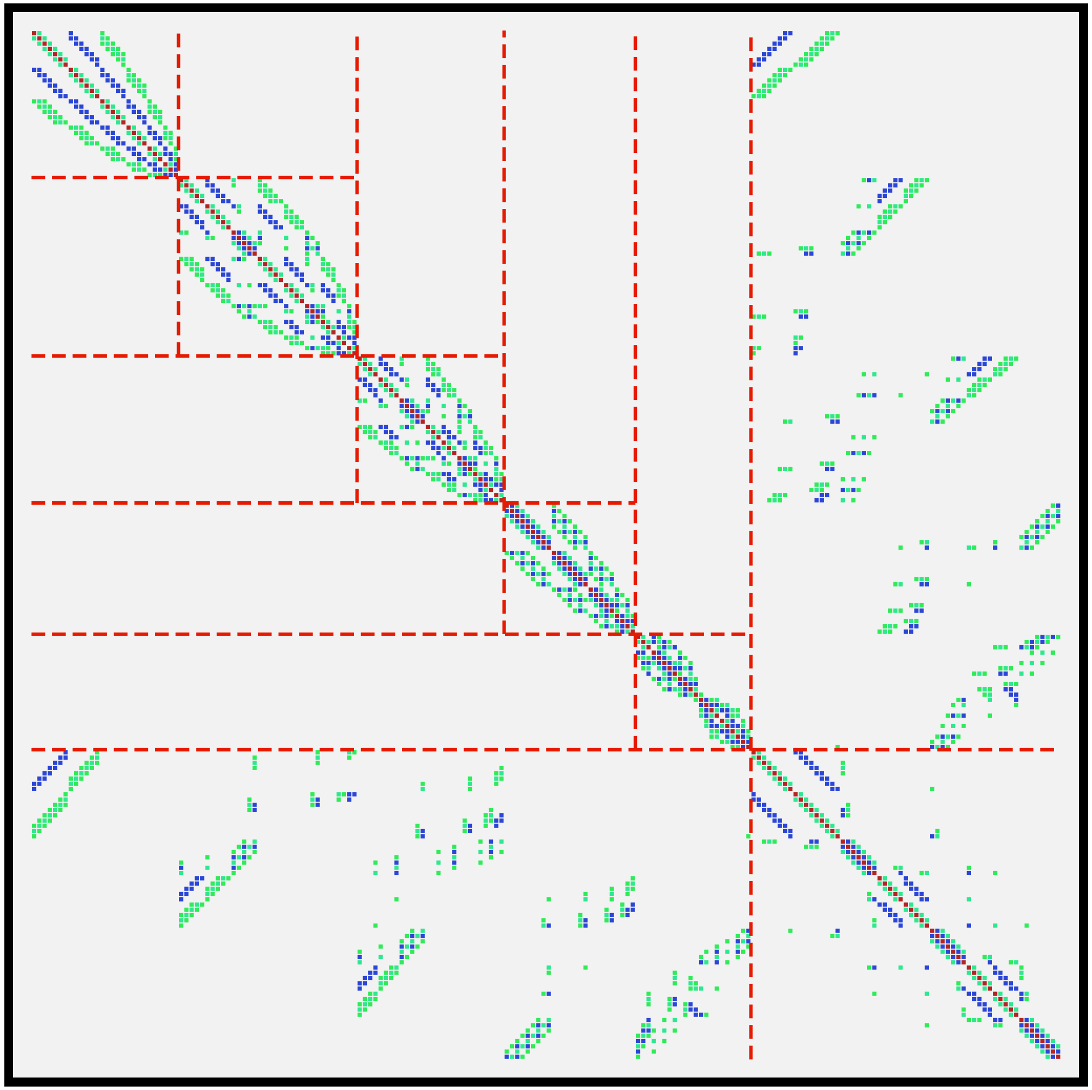
Original matrix



Block size of 6



Block size of 20



Related ideas

- See *Y. Notay*, Algebraic Multigrid and algebraic multilevel techniques, a theoretical comparison, NLAA, 2005.
- Some of these ideas are related to work by Axelsson and co-workers [e.g., AMLI] – see Axelson's book
- Work by Bank & Wagner on MLILU quite similar to ARMS – but uses AMG framework: [*R. E. Bank and C. Wagner*, Multilevel ILU decomposition, Numer. Mat. (1999)]
- Main difference with AMG framework: block ILU-type factorization to obtain Coarse-level operator. + use of relaxation.
- In AMG $S = P^T A P$ with P of size $(n_F + n_C) \times n_C$

Two-sided permutations with diag. dominance

Idea: ARMS + exploit nonsymmetric permutations

- No particular structure or assumptions for B block

- Permute rows * and * columns of A. Use two permutations P (rows) and Q (columns) to transform A into

$$PAQ^T = \begin{pmatrix} B & F \\ E & C \end{pmatrix}$$

P, Q is a pair of permutations (rows, columns) selected so that the B block has the 'most diagonally dominant' rows (after nonsym perm) and few nonzero elements (to reduce fill-in).

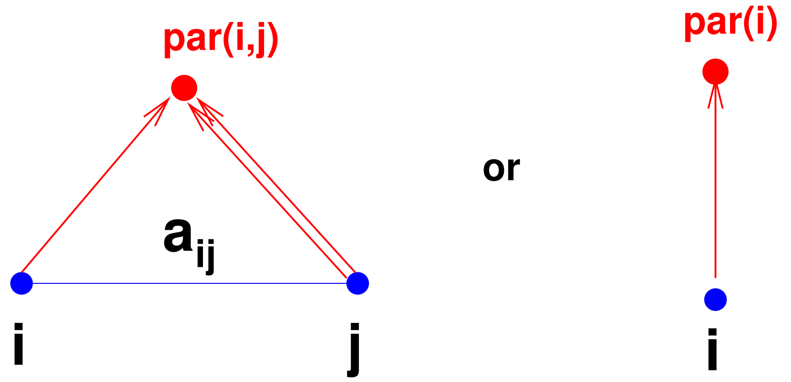
Coarsening

“Divide et imperia”, (Julius Caesar, 100BC-44BC)

Divide and conquer and coarsening

➤ Want to mix ideas from AMG with purely algebraic strategies based on graph coarsening

First step: Coarsen. We use matching: coalesce two nodes into one 'coarse' node

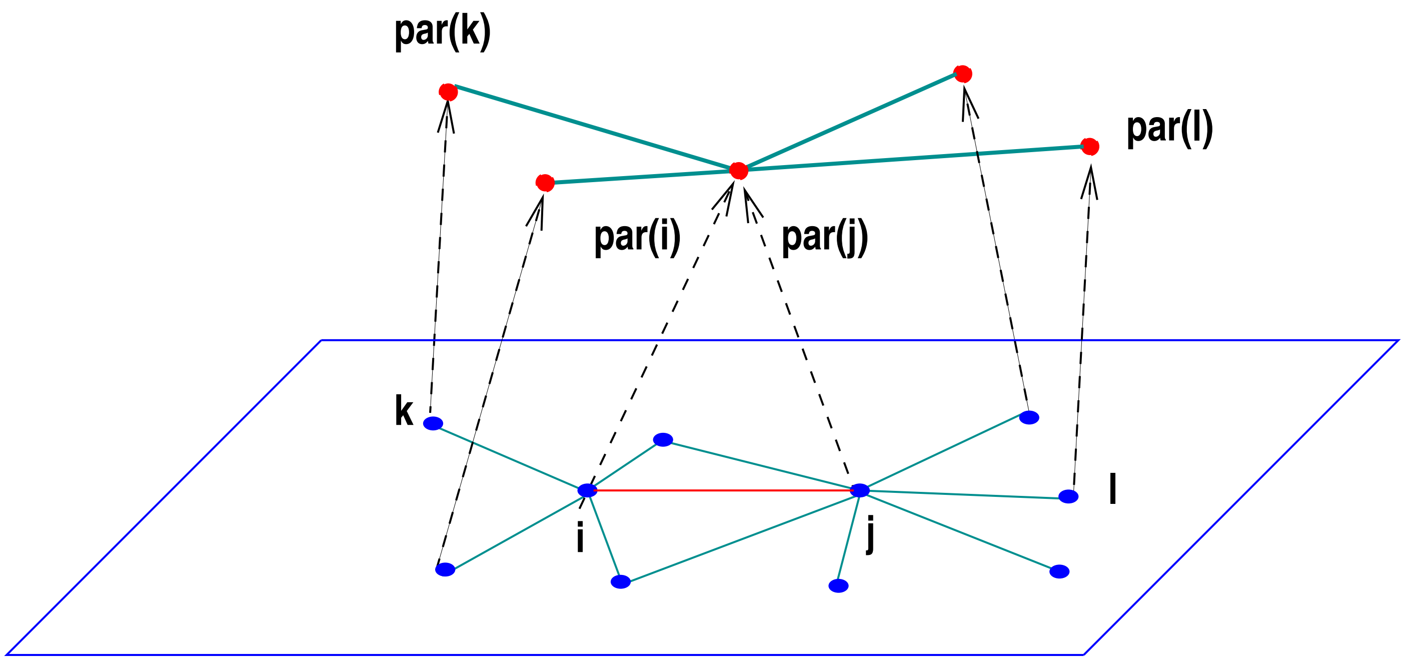


Second step: Get graph (+ weights) for the coarse nodes - Adj$[par(i,j)]$ is:

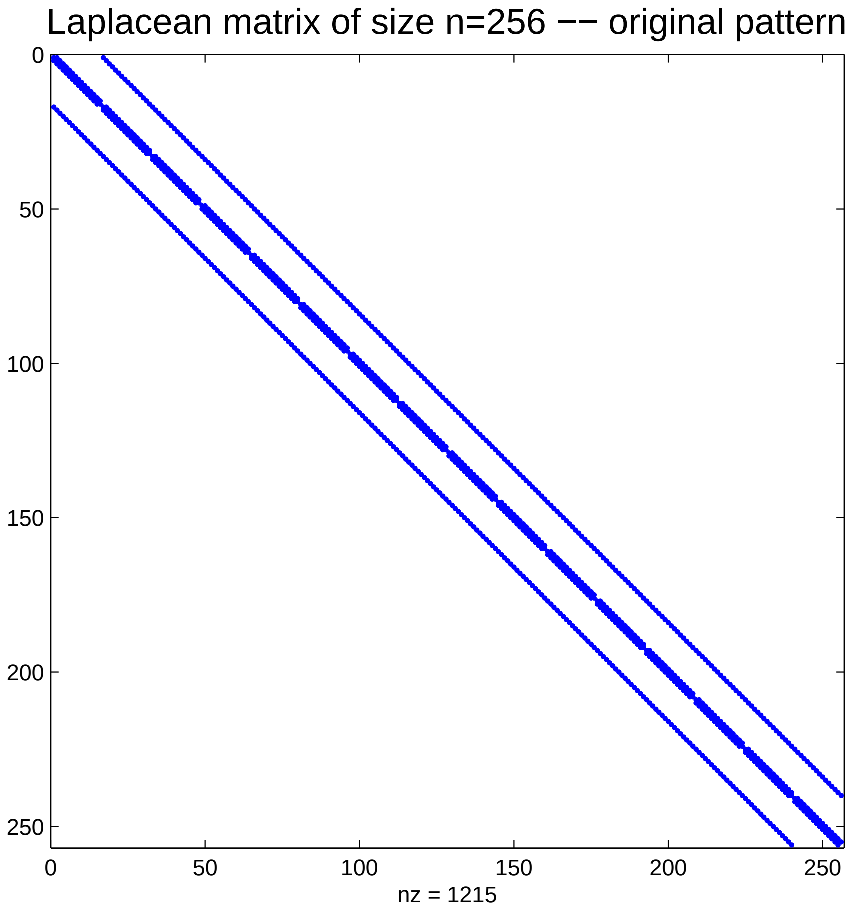
$$\{par(i,k)\ k \in Adj(i)\} \bigcup \{par(j,k)\ k \in Adj(j)\}$$

Third step: Repeat

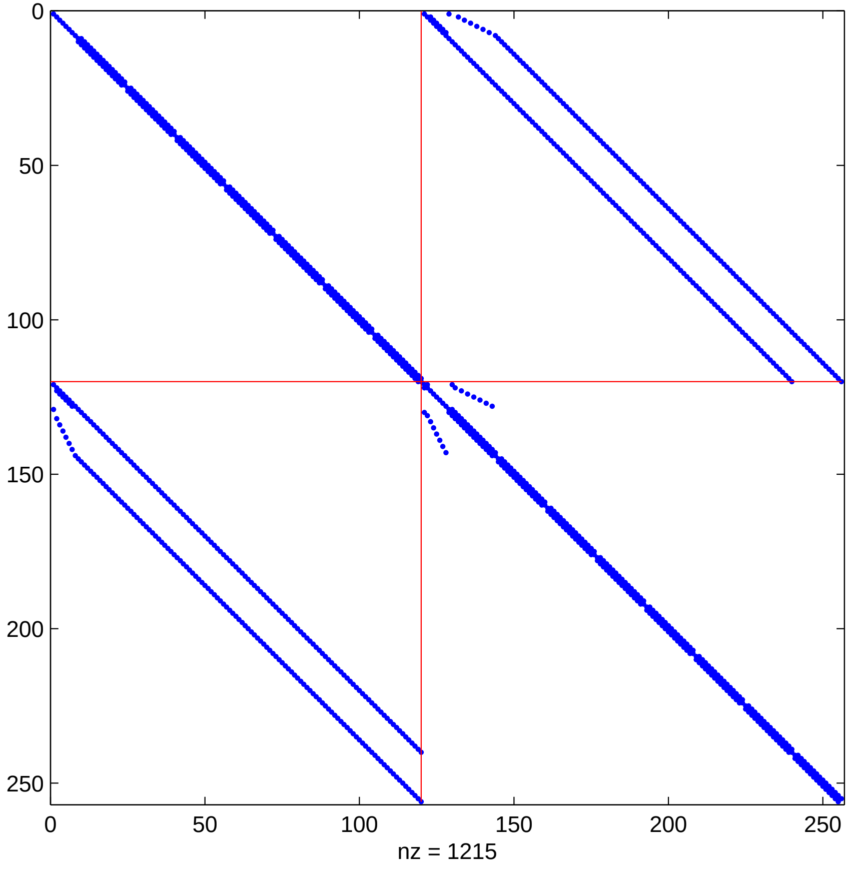
Illustration of the coarsening step



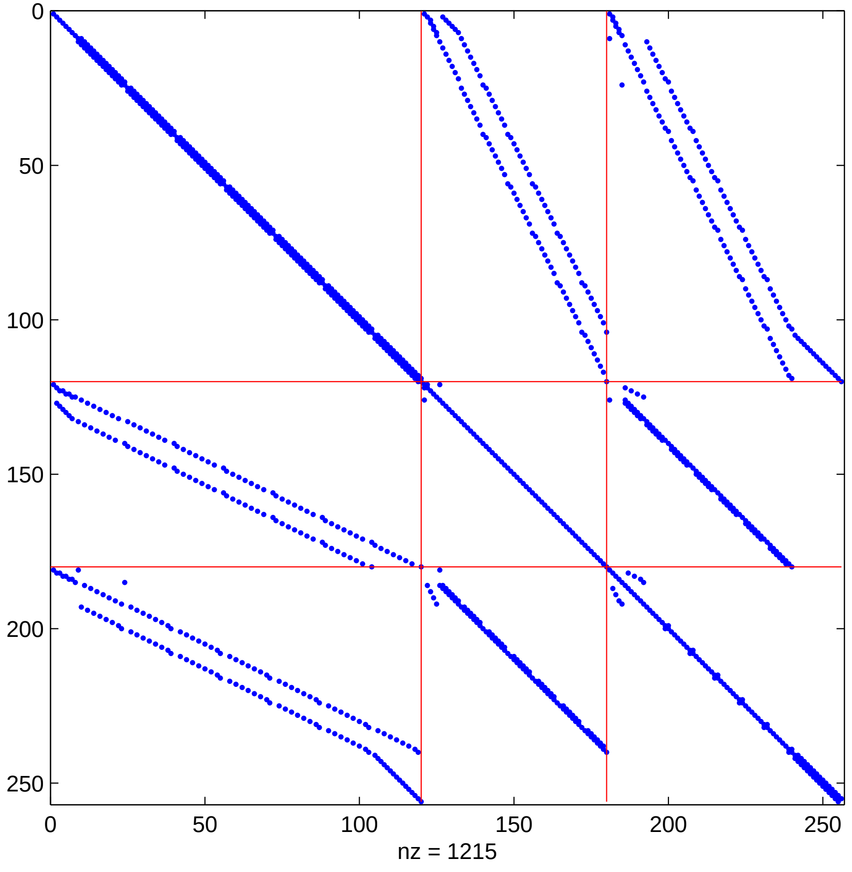
Example 1: A simple 16×16 mesh ($n = 256$).



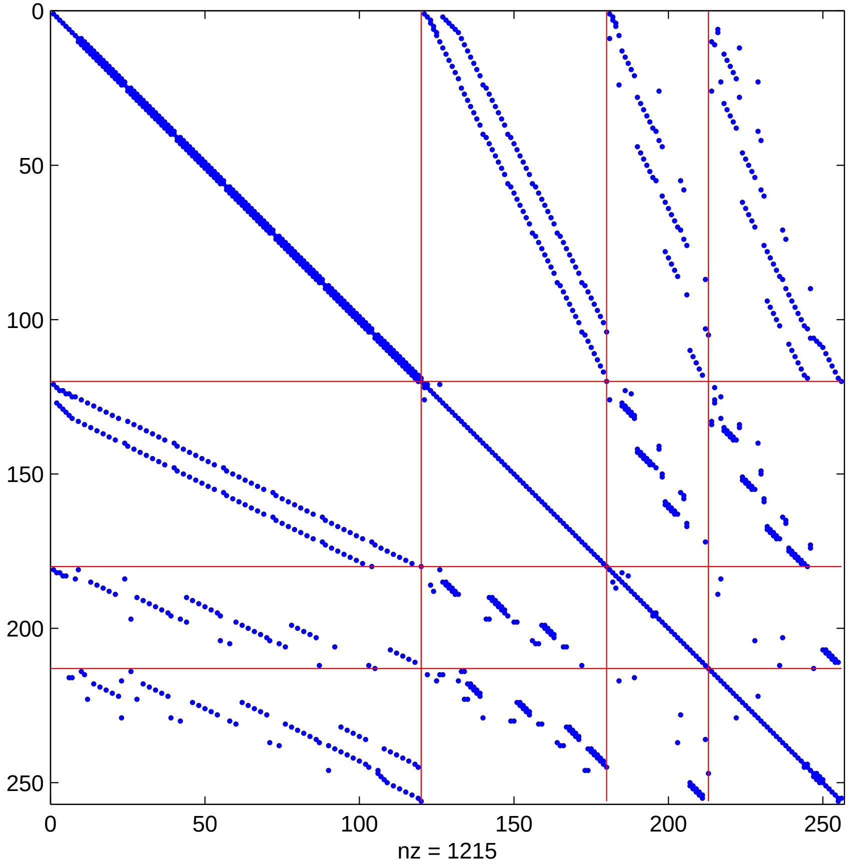
Matrix after 1 Levels of coarsening



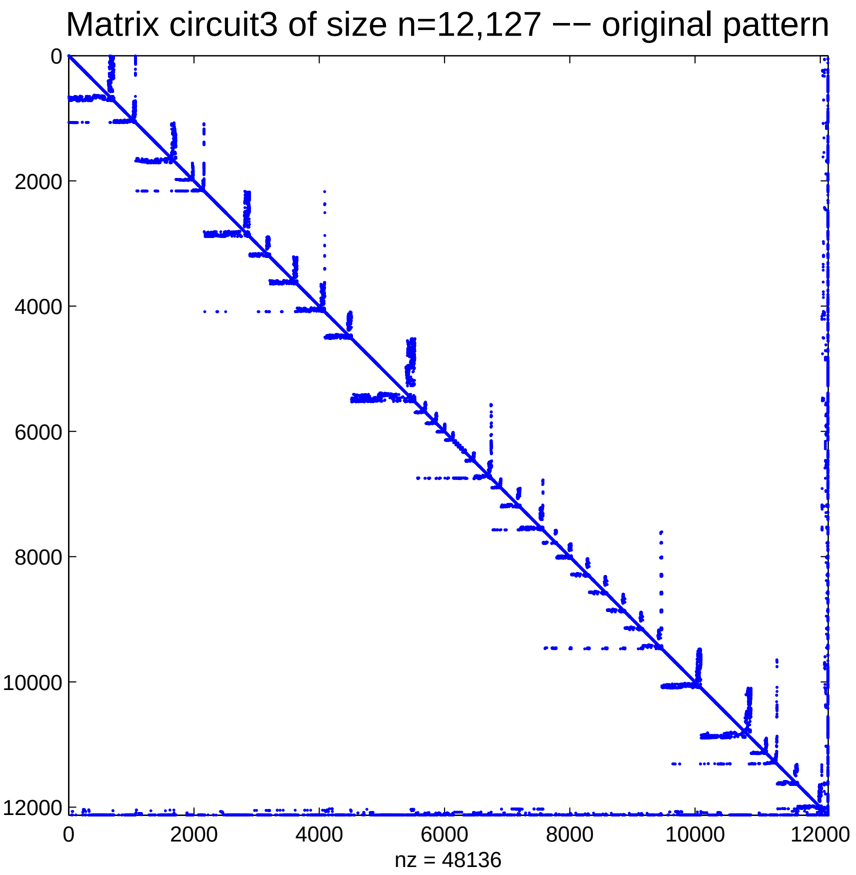
Matrix after 2 Levels of coarsening

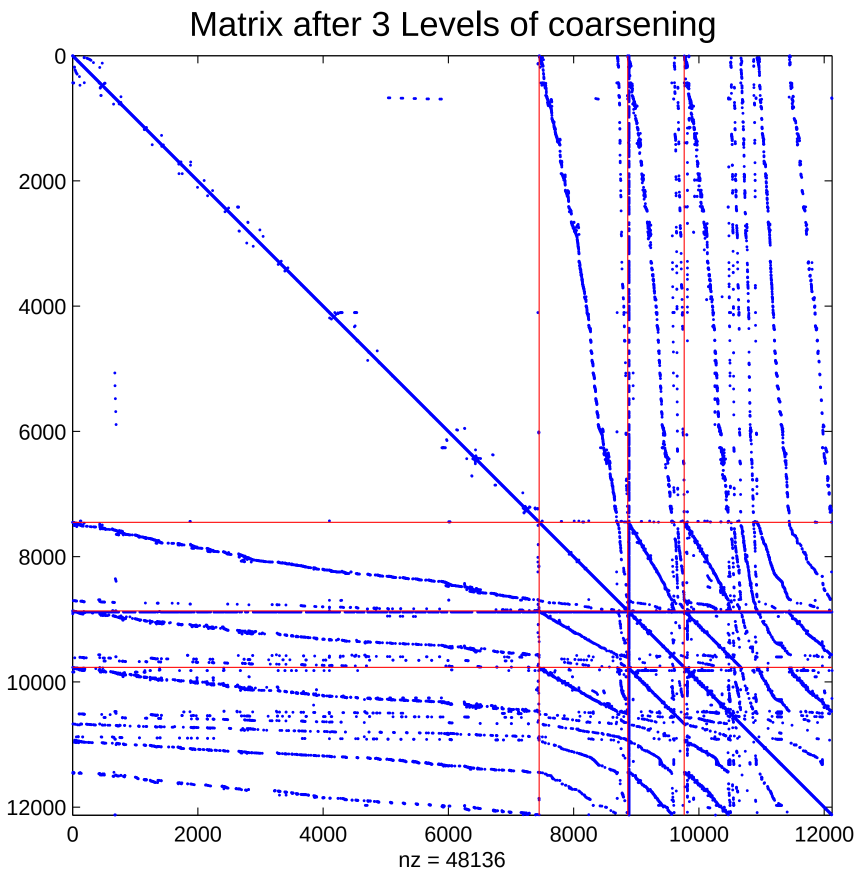


Matrix after 3 Levels of coarsening

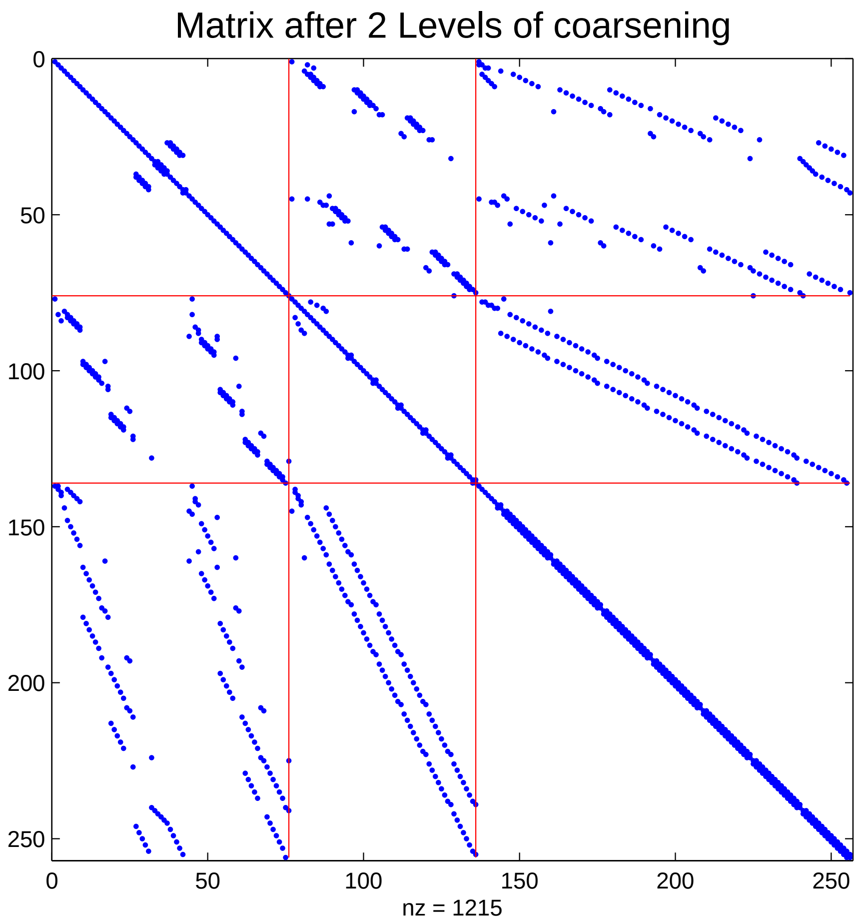


Example 2: circuit3 - An irregular matrix from circuit simulation





Note: Possible to order nodes the other way:



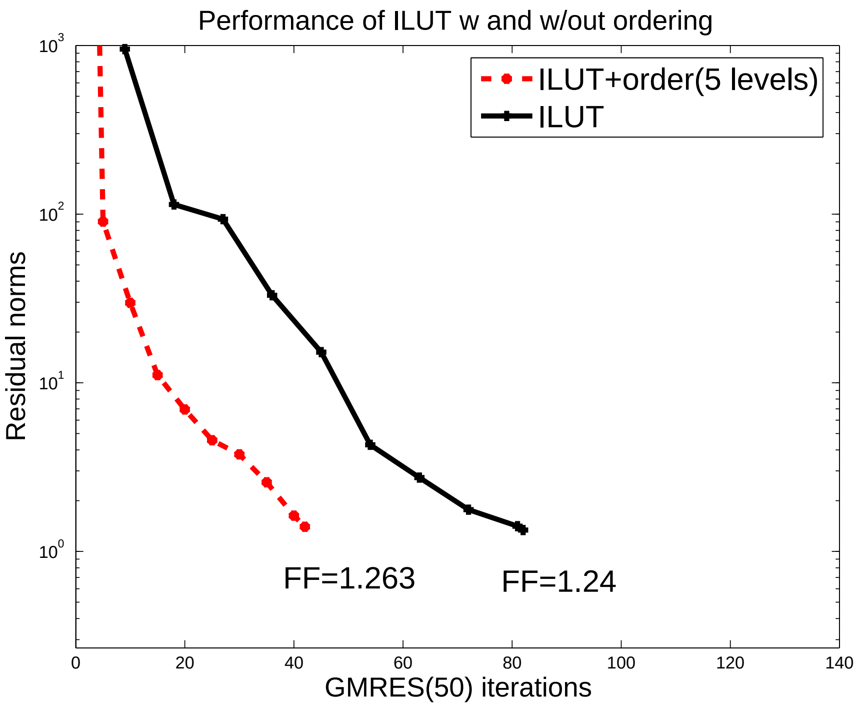
- Top left blocks always selected for good diagonal dominance properties
- Choice: Subdivide these blocks further – or subdivide remaining ones.
- Implemented both – advantages and disadvantages for each [main issue: cost]
- Wll illustrate only first ordering -

First idea: use ILU on the reordered matrix

- For example: use ILUT

Illustration: Matrix Raj1 from the Florida collection

- Size $n = 263,743$. $Nnz = 1,302,464$ nonzero entries
- Matrix is nearly singular – poorly conditioned. Iterate to reduce residual by 10^{10}.



- Reordering appears to be quite good for ILU.

Saving memory with Pruned ILU

- Let $A = \begin{pmatrix} B & F \\ E & C \end{pmatrix} = \begin{pmatrix} I & 0 \\ EB^{-1} & I \end{pmatrix} \begin{pmatrix} B & F \\ 0 & S \end{pmatrix}$;
- $S = C - EB^{-1}F$ = Schur complement

Solve:

$$\begin{pmatrix} I & 0 \\ EB^{-1} & I \end{pmatrix} \begin{pmatrix} B & F \\ 0 & S \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

1) $w_1 = B^{-1} b_1$
2) $w_2 = b_2 - E * w_1$
3) $x_2 = S^{-1} w_2$
4) $w_1 = b_1 - F * x_2$
5) $x_1 = B^{-1} w_1$

- Known result: LU factorization of S == trace of LU factorization of A.

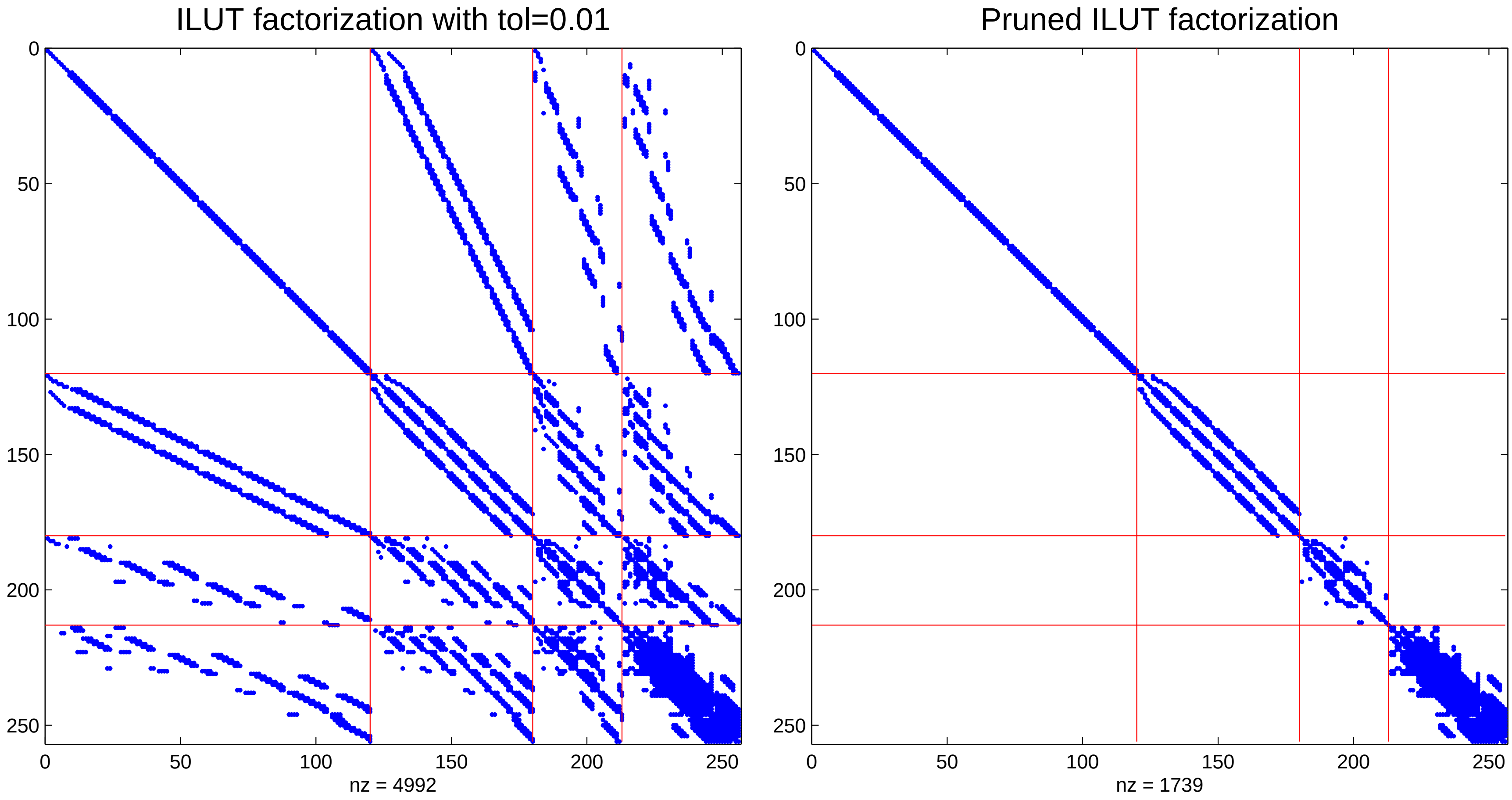
- Idea: exploit recursivity for B-solves - keep only the block-diagonals from ILU..

$$\text{From } LU = \left[\begin{array}{c|c|c} B_1 & B_1^{-1}F_1 & \\ \cline{1-2} E_1B_1^{-1} & S_1 & B_2^{-1}F_2 \\ \hline \multicolumn{2}{c|}{E_2B_2^{-1}} & S_2 \end{array}\right]$$

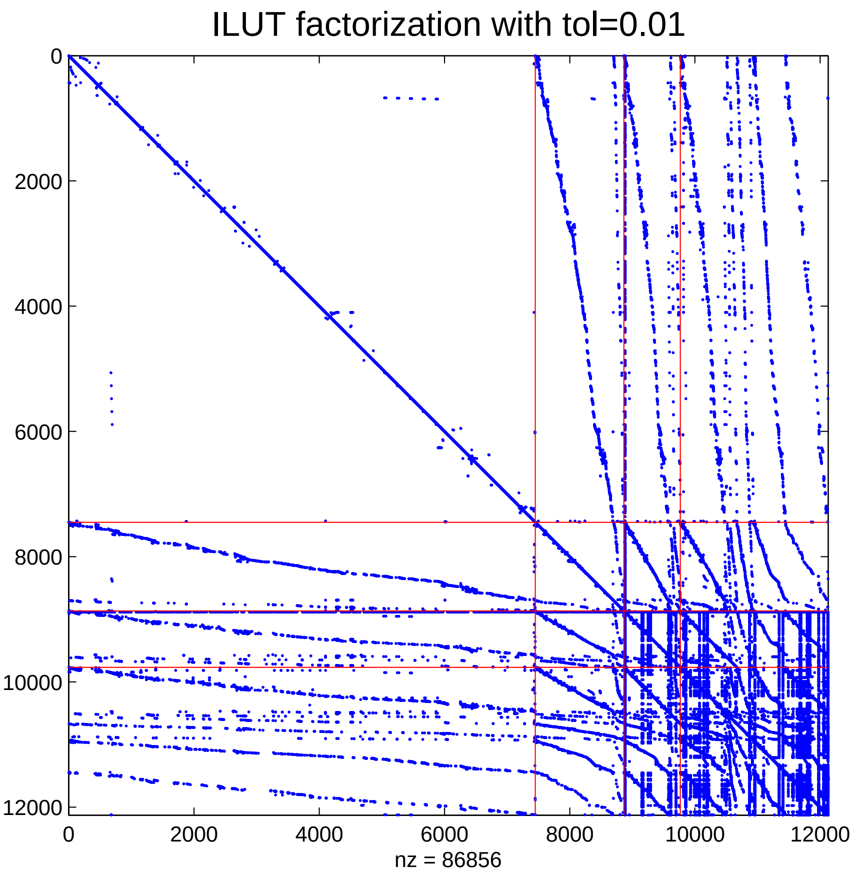
$$\text{Keep only} \left[\begin{array}{c|c|c} B_1 & & \\ \cline{1-2} & S_1 & \\ \hline \multicolumn{2}{c|}{} & S_2 \end{array}\right]$$

- Big savings in memory
- Additional computational cost
- Expensive for more than a few levels (2 or 3)..

Example 1: A simple 16×16 mesh ($n = 256$).



Example 2: circuit3 - An irregular matrix from circuit simulation



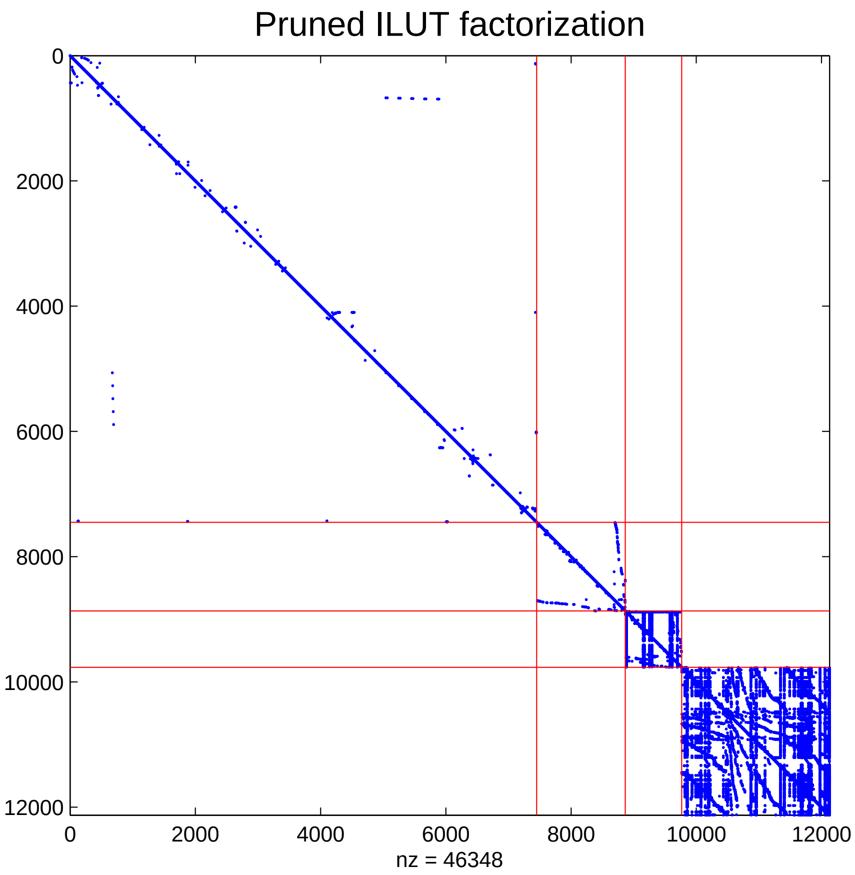
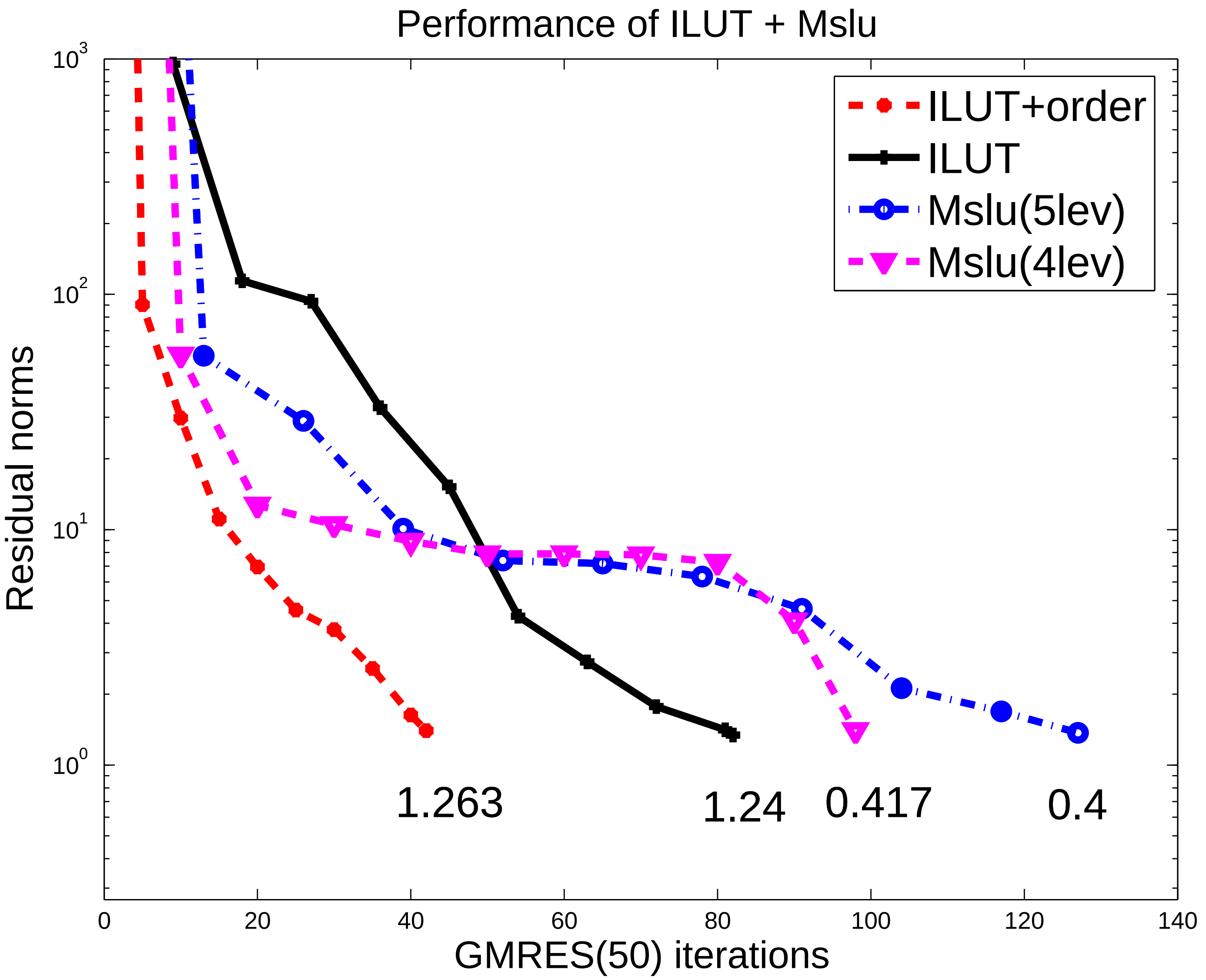


Illustration: Back to Raj1 matrix from the Florida collection

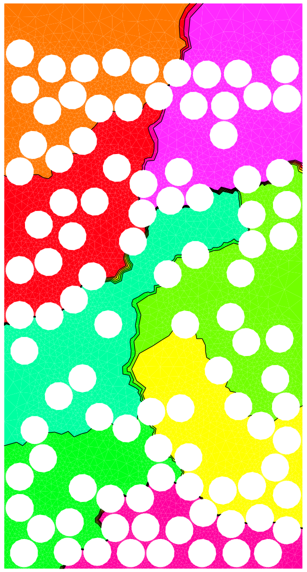


Another example – from solid-liquid flows

➤ Project we did about 8 years ago – with Dan Joseph, R. Glowinsky, ...

➤ Combines all complexities imaginable:

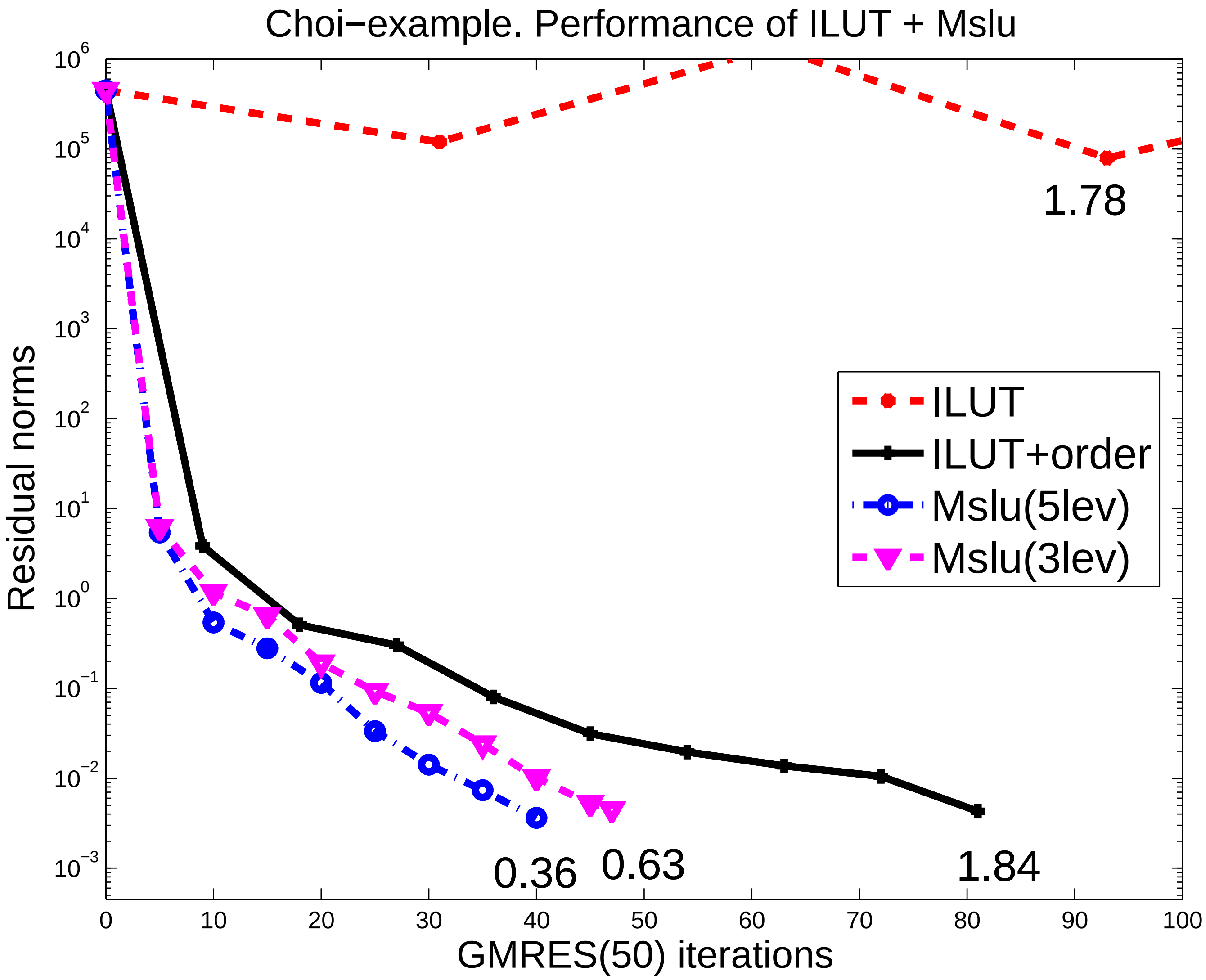
- Moving elements $\longrightarrow$ dynamic remeshing
- Difficult equations (nonlin) - ALE formulation used
- Large size problems - even in 2-D [for large number of particles]



➤ Matrix "Choi" is a small matrix from this application.

$\boldsymbol{n = 9,225}$, $\boldsymbol{nnz = 168,094}$, RHS is artificial.

Another example – from solid-liquid flows



Comparison

Meth.	Prec sec	Its sec.	fill-fact	Its
ILUT+C-ordering	0.520	1.240	1.843	81
Mslu(5lev)	0.450	2.270	0.362	40
Mslu(4lev)	0.530	1.120	0.626	47

Implementation issues

Preliminary implementation done in C [part of it must be re-done]

➤ Need a matrix (or a sequence of matrices) for the E-F part + an ILU factorization – i.e., store

$$\left[\begin{array}{c|c|c} & F_1 & \\ \cline{1-2} E_1 & & F_2 \\ \hline \multicolumn{2}{c|}{E_2} & \end{array}\right] \quad \text{and LU of} \quad \left[\begin{array}{c|c|c} B_1 & & \\ \cline{1-2} & S_1 & \\ \hline & & S_2 \end{array}\right]$$

➤ Main problem so far: issue of cost for large number of levels. [recursive calls]

Other options available with MSLU framework

- Can iterate at any level.
- Also possible: A form of block SSOR with the blocks of Schur complements
- Can (should) use a different drop tolerance for each level.
- Levels provide a middle ground between “levels of fill” and threshold dropping
- Can use approximate inverses in conjunction with ordering
- Forego ILUT factorization & perform relaxations instead. Would lead to a form of AMG

Issue 1: How to coarsen

- Basic criterion: If preconditioning matrix is ordered as

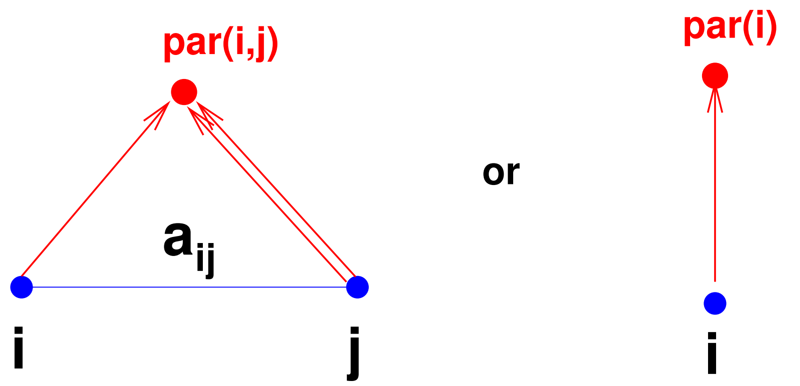
$$B = \begin{pmatrix} B_{FF} & B_{FC} \\ B_{CF} & B_{CC} \end{pmatrix}$$

Then, B_{FF} should be a good approximation to A_{FF}

- See Axelsson's book for bounds in SPD case.

- So far we used a heuristic based on diagonal dominance

- Scan the a_{ij}'s in a certain order
- If i "better" than j put i in F and j in C and vice-versa.



➤ Ideal procedure: [not implemented yet]

* Define one level using diagonal dominance,

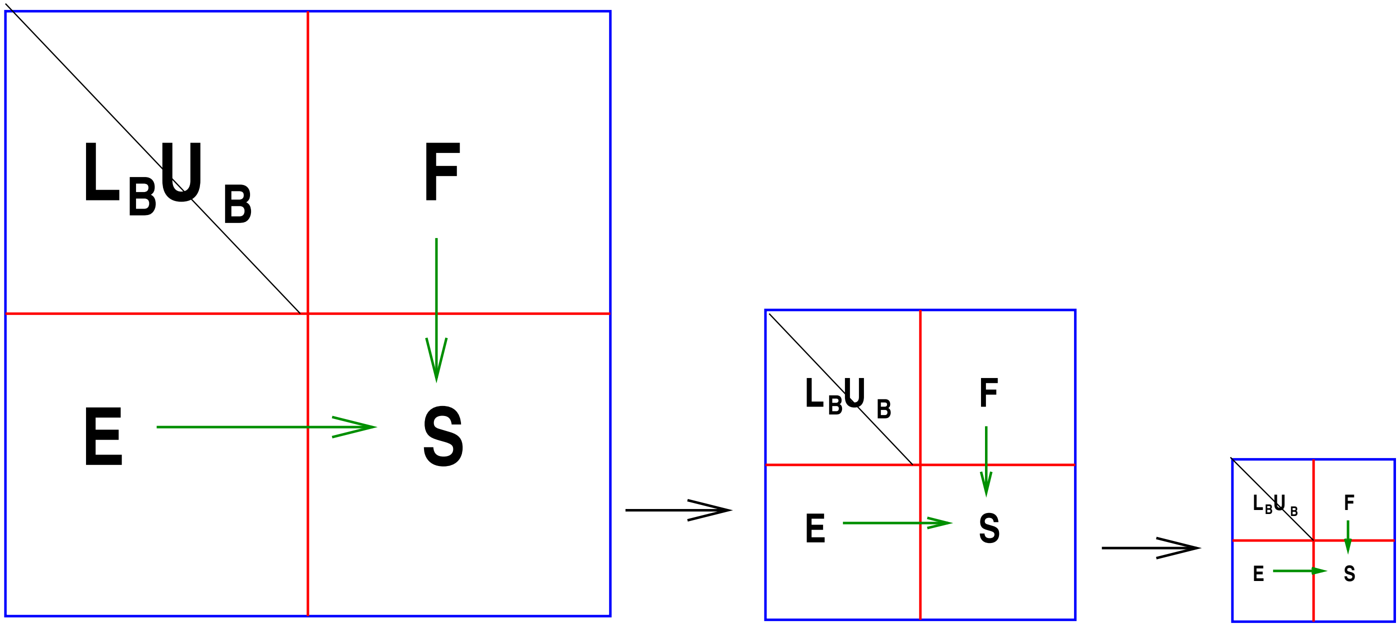
* Do elimination of fine nodes with ILU

* Get new diag. dominance factors

* Get new F and C sets and ...

* Repeat recursively..

➤ A picture is worth a thousand words



➤ Procedure quite similar to that of ARMS [Suchomel, YS 2002]

➤ See also: S. Mc Lachlan and YS [2007] on better ways to do coarsening in this context.

Issue 2: Use of nonsymmetric permutations

- Can use ARMS framework

- In this case, we only need to define: coarse-fine nodes after a selection of diagonal entries is made.

- In other words:

(a) first permute nonsymmetrically

(b) Then select fine / coarse sets [permute symmetrically]

Issue 3: Parallel Implementation

- Nice feature of MSLU (at least for easy problems):

can mingle coarsening and graph partitioning

THOUGHTS ON THE PRECONDITIONING MEETINGS

Previous preconditioning meetings we had :

1999 University of Minnesota, Minneapolis, June 10-12 1999.

2001 Granlibakken Conference Center, Tahoe City, April 29 - May 1, 2001.

2003 Napa Valley, Napa, October 27-29, 2003.

2005 Emory University, Atlanta, May 19-21, 2005.

2007 Météopole, Toulouse (France) July 9-19th, 2007

Focus: applications, ‘industrial’ problems, good mix of academia, Gov. Labs, and industry resesarchers.

MANY thanks to :

*** Michele Benzi for hosting the 2005 meeting

*** Luc Giraud for hosting the 2007 meeting

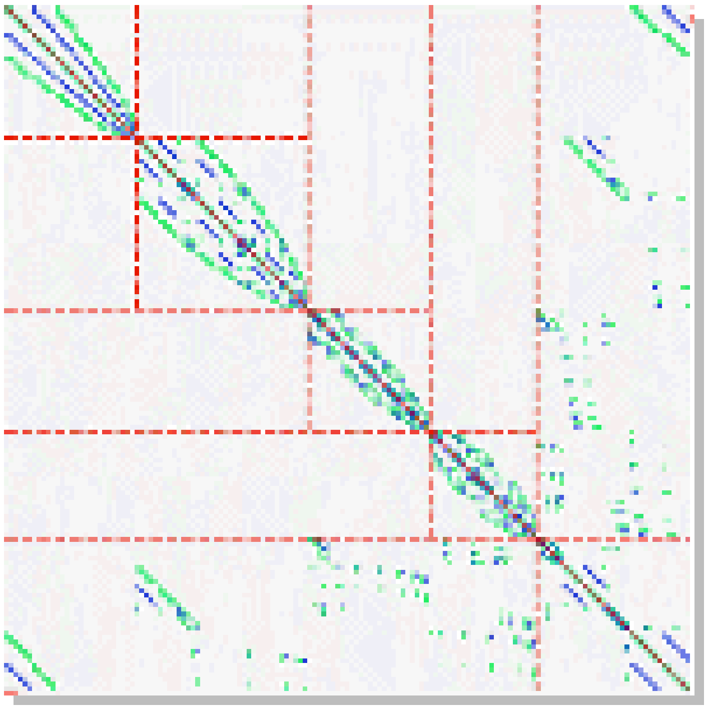
*** Michael Ng for hosting this meeting!

Hindsight...

What has changed since the first Preconditioning meeting?

Some thoughts on the Preconditiong meetings

What has changed since the first Preconditioning meeting?



This picture has not...

Ten years ago...

➤ Observations made from the intro to the special issue:

> *".... (1) the diminishing focus on parallel algorithms and implementations, (2) the continuing importance of sparse approximate inverse methods, (3) iterative solvers have been shown to be useful in areas (e.g. circuit simulation) where they were insucessful before"*

➤ Several talks on Approximate inverses. [A. Yeremin, E. Chow, Benzi-Tuma, ..], ..

➤ .. a few others on applications [W. Schilders, P. Forsythe, ..]

➤ A few talks on "saddle-point" problems [H. Elman, A. Wathen, ..]

Remarkably: topics are very similar today –

- Several talks on Approximate inverses.
- .. a few others on applications
- A few talks on “saddle-point” problems

One could easily copy the preface from the 1999 special issue

...

You mean we are a little repetitious??

"Well – Look at it this way, it's been proven that repetition is a good way to learn."

So - what is left to be done on preconditioners?

➤ From one viewpoint: we are spinning wheels – basically similar ideas recycled time and time again

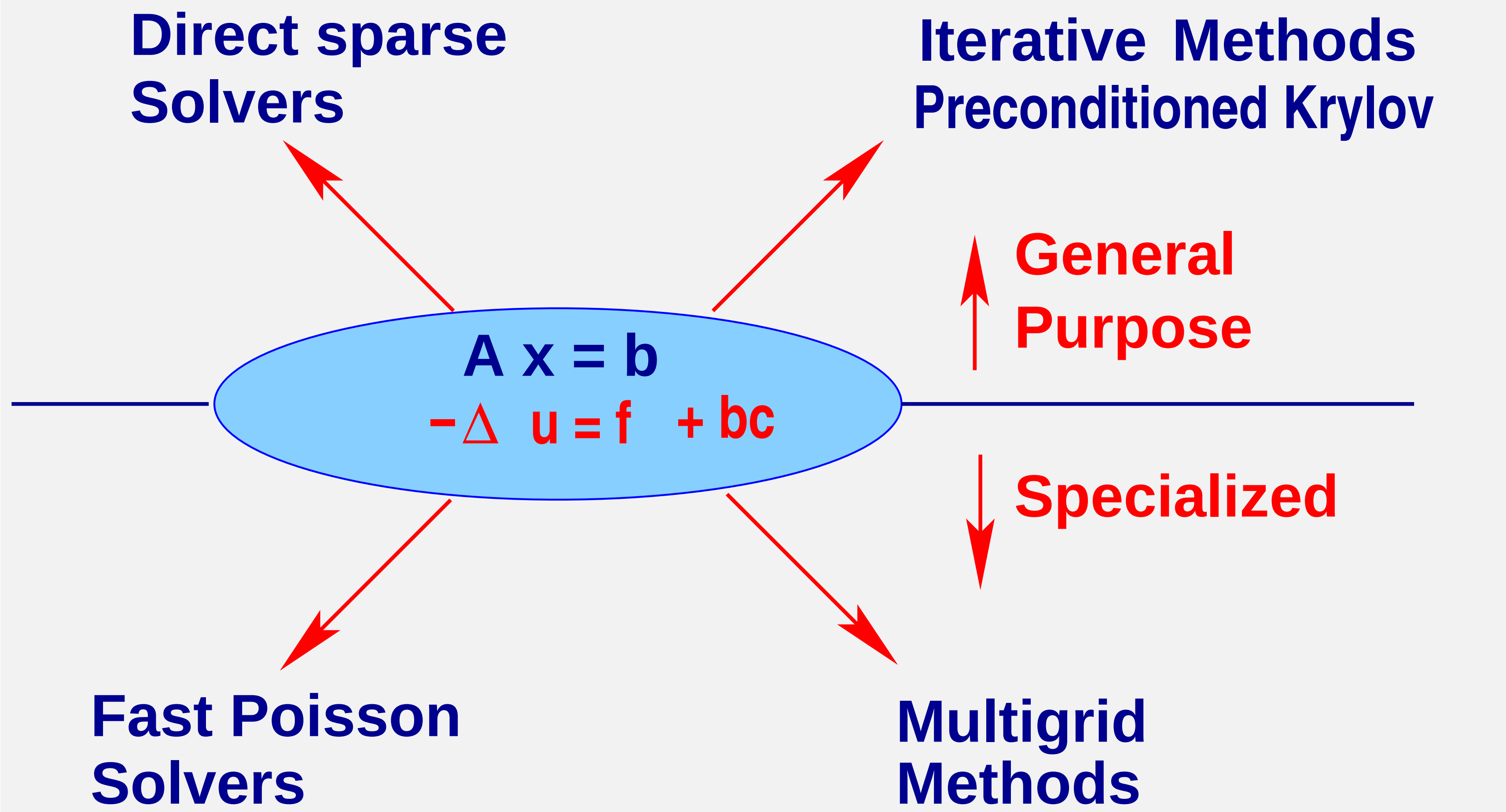
➤ Yet: few practitioners are satisfied with the state of the art.

Accomplishments made in past 10 years or so..

- Theory: some [e.g., saddle point problems..]
- Implementation/ software: very little [PETSc born in 1995!]
- Integration into applications: a lot [circuits, control, Helmholtz,..]
- Algorithmic innovations: Not too many. [e.g. use of nonsymmetric permutations (MC64, etc.)]

  [Note: this is purely for the area 'general-purpose' solvers]

So - what is left to be done on preconditioners?



Observations: what is still lacking

1. Parallel iterative packages have stagnated - not too satisfactory
2. Good understanding of relation partitioners+solvers [i.e. integrating partitionners into solvers]
3. Robust software based on a middle-ground approach [between general purpose and specialized]

What has been elusive

1. A truly robust black- box iterative solver –
2. A truly black-box AMG ['linear' scaling] solver
3. A really good parallel iterative solution software ..

One Difference with 1999: We are 10 years older and a bit ... wiser.

"Listen, you and I know there is very little left to be done in this area.. but let the young ones break their neck trying to find the miracle black-box solver. Wll just sit and watch."

What mini-trends are we seeing?

1. Rapprochement between iterative and direct solvers [the end of a long cold war?]
2. AMG seems to be gaining ground
3. Diminishing importance of “accelerators”
4. New interests in numerical linear algebra: data mining, problems in physics, bio, ... Wll CFD still rule NA?
5. Renewed interest in high-performance computing funding after years of slight neglect.
6. Parallelism is everywhere in applications world
7. Worry from impact of yet another architecture invasion

   Remember the title “invasion of the killer micros”?

More immediate down-to-earth questions

- Will it be useful to have other Preconditioning meetings?
- If so, should we change the main theme (s)?
- Do we feel that the current theme has “played itself out”?
- Or that there is a renewal of sorts under way [new architectures, new applications, ...] and that ...
- ... The conditions are similar to those of 1999?
- Is a small & focused meeting still needed given the other options available [SIAM-LAA, SIAM-CSE, Copper mountain meetings, ...] ?

Give us your thoughts

A couple of quotes from “Who moved my cheese”

- Highly recommended reading. Author: Dr. Spencer Johnson (1998)

“If you do not change, you can become extinct.”

“The Quicker You Let Go Of Old Cheese, The Sooner You Can Enjoy New Cheese.”