Unsymmetric Greedy Orderings for Stable Sparse Factorization

X. Sherry Li (LBNL, Berkeley).

with

Patrick R. Amestoy (ENSEEIHT-IRIT, Toulouse)

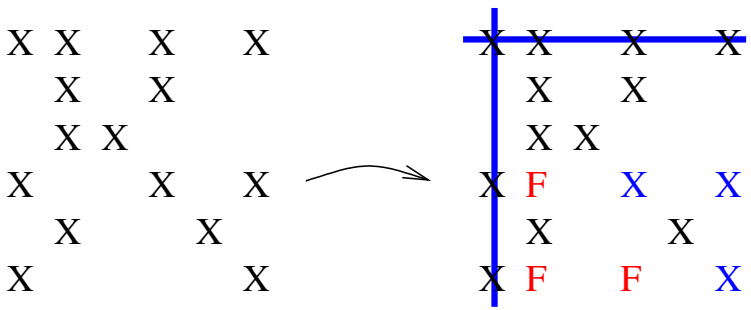
Stéphane Pralet (CNRS/ENSEEIHT-IRIT, Toulouse)

Outline

Three-phase approach

Solve $\mathbf{Ax} = \mathbf{b}$ with $\mathbf{A}$ sparse

- **Analysis**. Preprocessing.
- **Factorization**. compute $\mathbf{L}$ and $\mathbf{U}$. At step k, $a_{ij}^{(k+1)} = a_{ij}^{(k)} - \frac{a_{ik}^{(k)} a_{kj}^{(k)}}{a_{kk}^{(k)}}$.
  - rounding errors depend on growth factor
    $\rightarrow$ large pivots $\approx$ small errors
  - fill-in in the factors



- **Triangular solution**.

Motivations of this work

- Analysis often involves :
  - numerical step (scaling and maximum transversal)
  - structural reordering step (symmetric permutation to keep healthy diagonal)

  $\mathbf{A}' = \mathbf{P}(\mathbf{D}_r\mathbf{A}\mathbf{D}_c\mathbf{Q})\mathbf{P}^{\mathrm{T}}$
- Main objective of proposed ordering : *combine numerical preprocessing and reordering steps*
  - add flexibility to choose off-diagonal pivots $\longrightarrow$ reduce fill-in
  - control numerical quality of predicted pivots $\longrightarrow$ improve accuracy, reduce off-diagonal pivoting

Components of our unsymmetric ordering

Constraint matrix **C**, such that

- choose some $a_{ij} \neq 0$, set $c_{ij} = a_{ij}$; otherwise $c_{ij} = 0$
- **C** structurally nonsingular

Phase 1: Numerical preprocessing of **A** and construction of **C**.
Phase 2: Reorder **A**, with candidate pivots in **C**.

(Special case: $\mathbf{C} = I$, choose pivots on main diagonal.)

Phase 1: generic pre-processing ($NumThresh$, $StructThresh$)

Compute row and column scalings of $\mathbf{A}$, $\mathbf{A} \leftarrow \mathbf{D}_r\mathbf{A}\mathbf{D}_c$,

Build matrix $\mathbf{C}$:

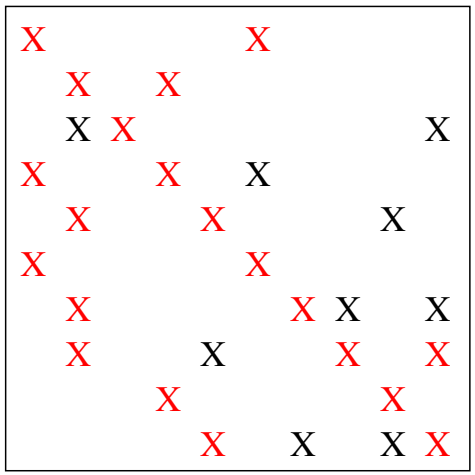
- $Struct(\mathbf{C}) = \{(i,j) \ : \ |a_{ij}| > NumThresh\}$,
  ($Struct(\mathbf{C}) \subseteq Struct(\mathbf{A})$)
- store numerical values in $\mathbf{C}$ if needed,
- add entries from $\mathbf{A}$ s.t. maximum matching $\mathcal{M} \subseteq \mathbf{C}$
- if needed, remove entries (not in $\mathcal{M}$) from $\mathbf{C}$
  until $|\mathbf{C}| < StructThresh$

Phase 2: one step of ordering

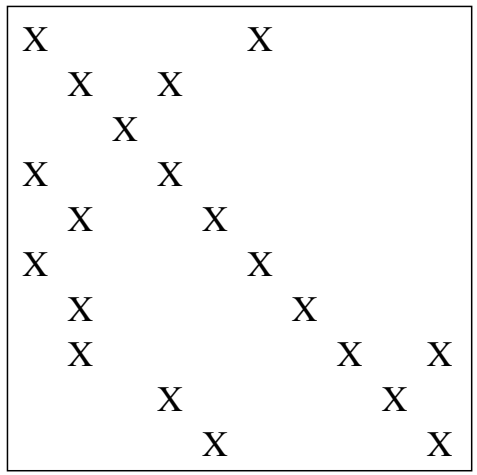
Let **A** *be a scaled matrix.*

C contains a subset of **A** entries

A =



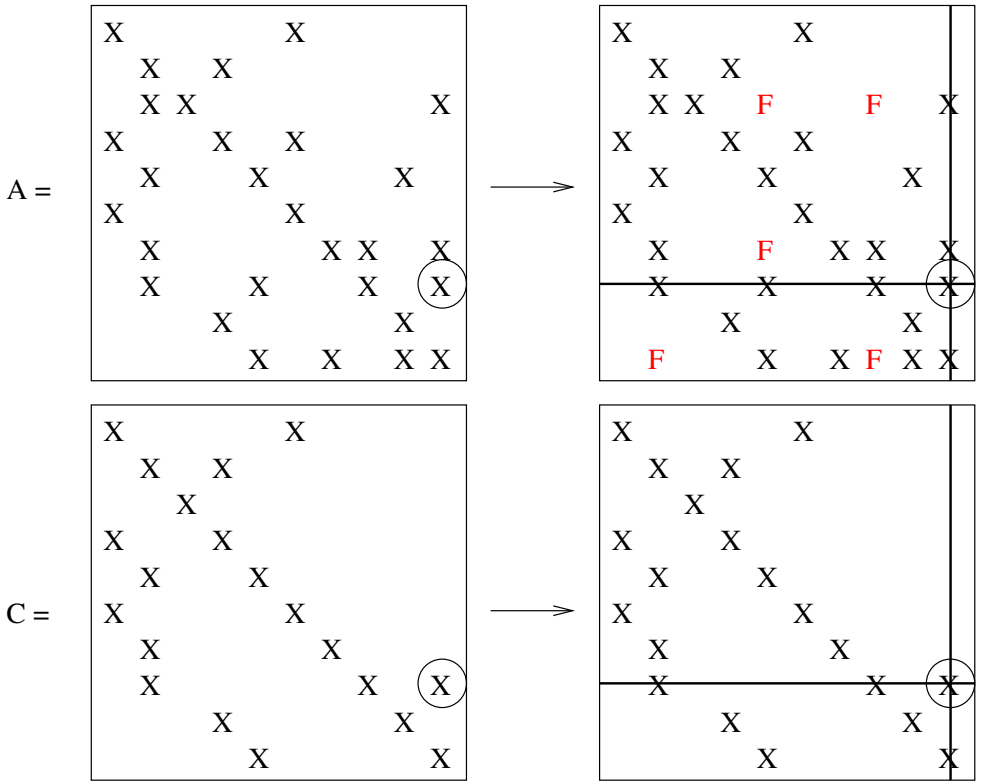
C =



Phase 2: one step of ordering

Let **A** *be a scaled matrix.*

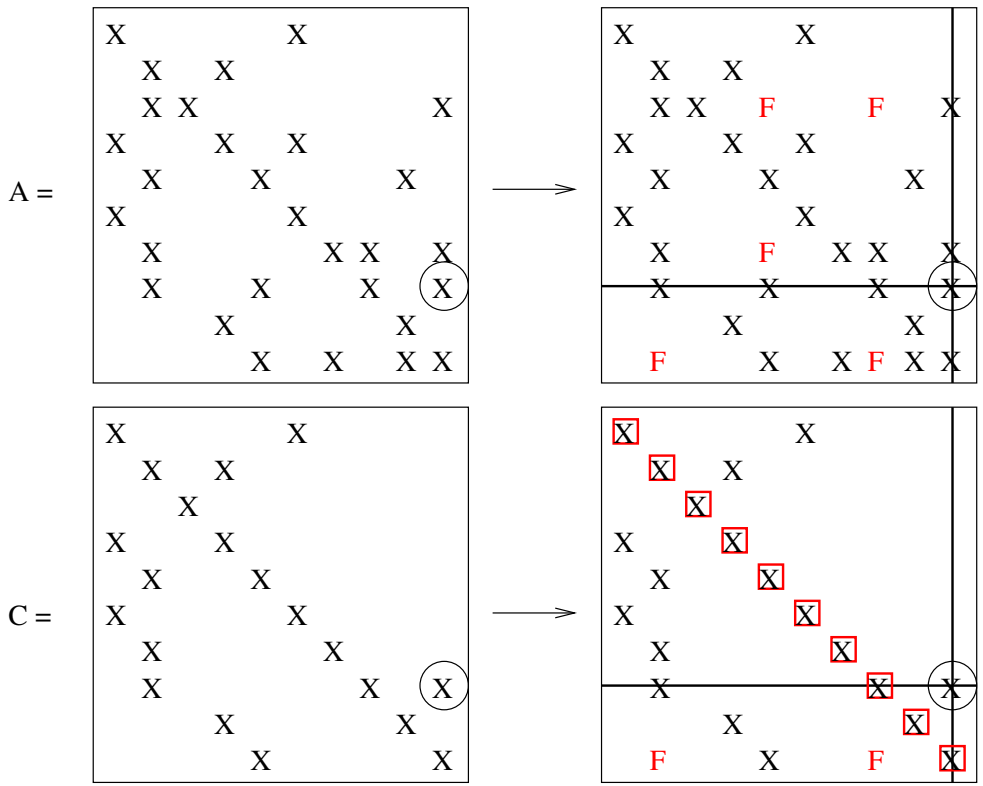
Pivot is selected and eliminated in **A**



Phase 2: one step of ordering

Let **A** *be a scaled matrix.*

Fill-in may occur in **C**

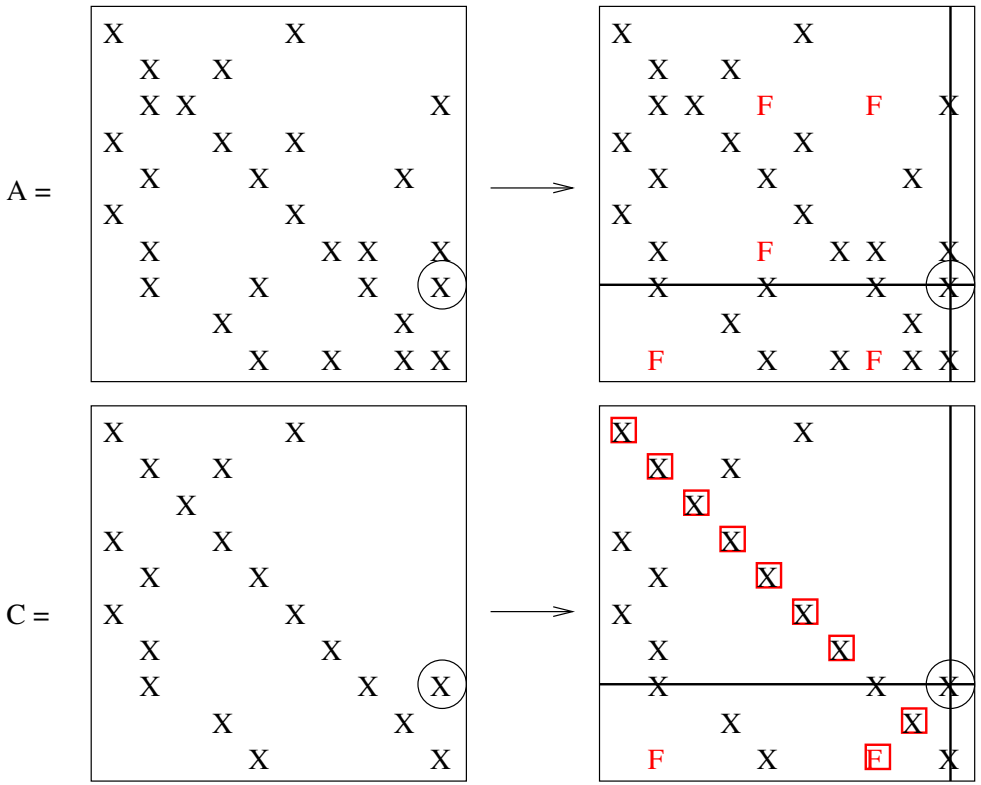


- May need to restrict fill-in in **C**.

Phase 2: one step of ordering

Let **A** *be a scaled matrix.*

Need update matching in **C** to maintain nonsingularity

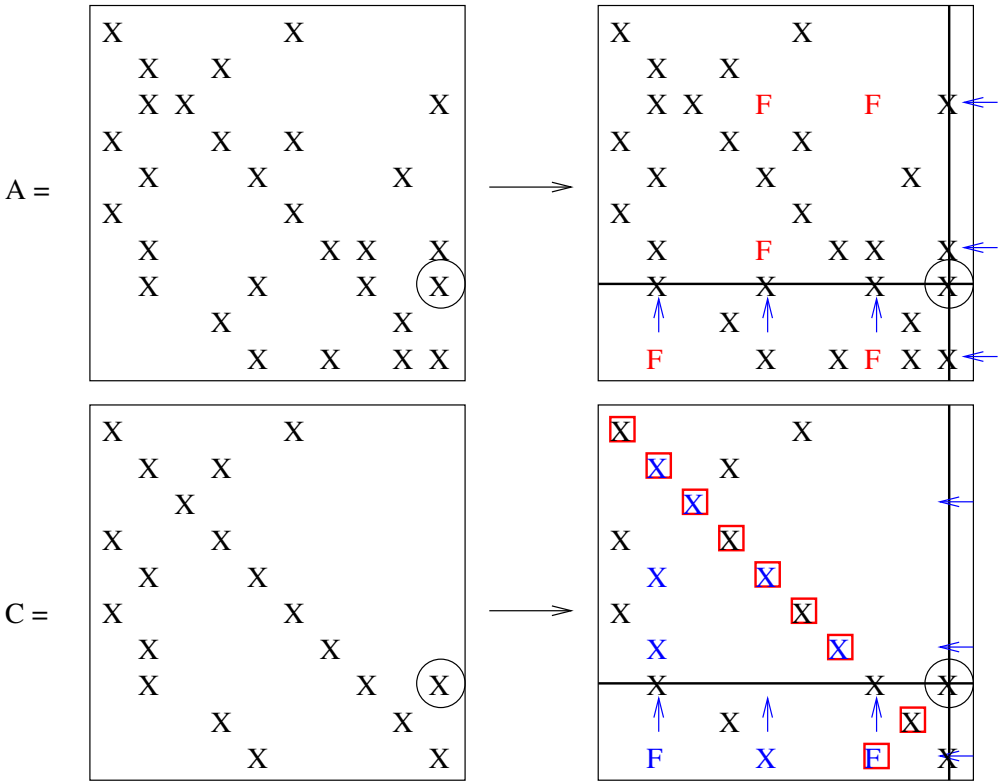


- This happens if pivot $(i, j) \notin \mathcal{M}$.

Phase 2: one step of ordering

Let **A** *be a scaled matrix.*

Which entries are affected by the metric update?



Phase 2: Constrained unsymmetric ordering

Let $\mathbf{A}^1 = \mathbf{A}$ (phase 1 : $\mathbf{A} \leftarrow \mathbf{D}_r\mathbf{A}\mathbf{D}_c$) and $\mathbf{C}^1 = \mathbf{C}$
while $k \leq n$ **do**
  Select best pivot in $\mathbf{C}^k$ w.r.t metric$(\mathbf{A}^k, \mathbf{C}^k)$
  $\mathbf{C}^{k+1} \leftarrow$ Update $(\mathbf{C}^k)$
  $\mathbf{A}^{k+1} \leftarrow$ Update $(\mathbf{A}^k)$
  Update metric values
end while

Phase 2: Constrained unsymmetric ordering

Let $\mathbf{A}^1 = \mathbf{A}$ (phase 1 : $\mathbf{A} \leftarrow \mathbf{D}_r\mathbf{A}\mathbf{D}_c$) and $\mathbf{C}^1 = \mathbf{C}$

while $k \leq n$ **do**

Select best pivot in $\mathbf{C}^k$ w.r.t metric$(\mathbf{A}^k, \mathbf{C}^k)$ $\rightarrow$ Which metric?

$\mathbf{C}^{k+1} \leftarrow$ Update $(\mathbf{C}^k)$

$\mathbf{A}^{k+1} \leftarrow$ Update $(\mathbf{A}^k)$

Update metric values

end while

Phase 2: Constrained unsymmetric ordering

Let $\mathbf{A}^1 = \mathbf{A}$ (phase 1 : $\mathbf{A} \leftarrow \mathbf{D}_r\mathbf{A}\mathbf{D}_c$) and $\mathbf{C}^1 = \mathbf{C}$

while $k \leq n$ **do**

Select best pivot in $\mathbf{C}^k$ w.r.t metric$(\mathbf{A}^k, \mathbf{C}^k)$ $\rightarrow$ Which metric?

$\mathbf{C}^{k+1} \leftarrow$ Update $(\mathbf{C}^k)$ $\rightarrow$ In-place ?

$\mathbf{A}^{k+1} \leftarrow$ Update $(\mathbf{A}^k)$

Update metric values

end while

Phase 2: Constrained unsymmetric ordering

Let $\mathbf{A}^1 = \mathbf{A}$ (phase 1 : $\mathbf{A} \leftarrow \mathbf{D}_r\mathbf{A}\mathbf{D}_c$) and $\mathbf{C}^1 = \mathbf{C}$

while $k \leq n$ **do**

Select best pivot in $\mathbf{C}^k$ w.r.t metric$(\mathbf{A}^k, \mathbf{C}^k)$ $\rightarrow$ Which metric?

$\mathbf{C}^{k+1} \leftarrow$ Update $(\mathbf{C}^k)$ $\rightarrow$ In-place ?

$\mathbf{A}^{k+1} \leftarrow$ Update $(\mathbf{A}^k)$ $\rightarrow$ In-place

Update metric values

end while

Phase 2: Constrained unsymmetric ordering

Let $\mathbf{A}^1 = \mathbf{A}$ (phase 1 : $\mathbf{A} \leftarrow \mathbf{D}_r\mathbf{A}\mathbf{D}_c$) and $\mathbf{C}^1 = \mathbf{C}$

while $k \leq n$ **do**

Select best pivot in $\mathbf{C}^k$ w.r.t metric$(\mathbf{A}^k, \mathbf{C}^k)$ $\rightarrow$ Which metric?

$\mathbf{C}^{k+1} \leftarrow$ Update $(\mathbf{C}^k)$ $\rightarrow$ In-place ?

$\mathbf{A}^{k+1} \leftarrow$ Update $(\mathbf{A}^k)$ $\rightarrow$ In-place

Update metric values $\rightarrow$ Cheap metric updates

end while

Constraints and algorithmic solutions

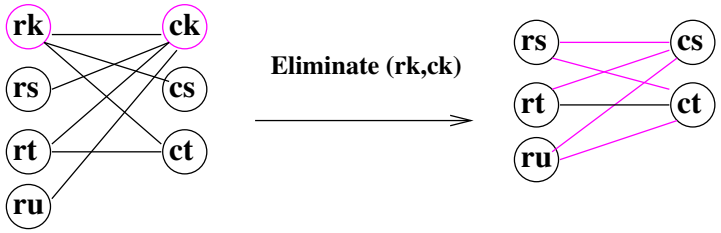
- To update $\mathbf{A}^k$: **in-place** algorithm $\longrightarrow$ bipartite quotient graph
- To update $\mathbf{C}^k$:
  - **knowledge of the metric** of each entry in $\mathbf{C}^k$ $\longrightarrow$ weighted bipartite graph and incomplete updates such that
    - $\mathbf{C}^k$ **structurally nonsingular**
    - **Struct**$(\mathbf{C^k}) \subseteq$ **Struct**$(\mathbf{A^k})$
  - MATCHUPDATE : minimum cost to only preserve $\mathbf{C}^k$ structurally nonsingular which can be done **in-place**
    (remove 3 entries (i, j), (mi, j), (i, mj), add 1 entry (mi, mj))

	j	mj
i	P	M
mi	M	F

  - TOTALUPDATE : perform all updates in $\mathbf{C}^k$

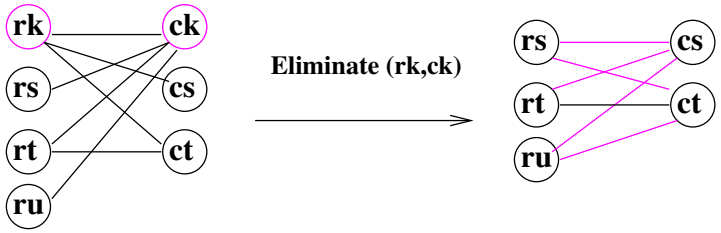
Bipartite quotient graph [Pagallo & Maulino, 1983]

- Bipartite graphs $G^k = \{V_r^k, V_c^k, E^k\}$ to model the elimination process of $\mathbf{A}$ (with $G^0 = G(\mathbf{A})$).
- After eliminating pivot (r_k, c_k), a bi-clique is formed.

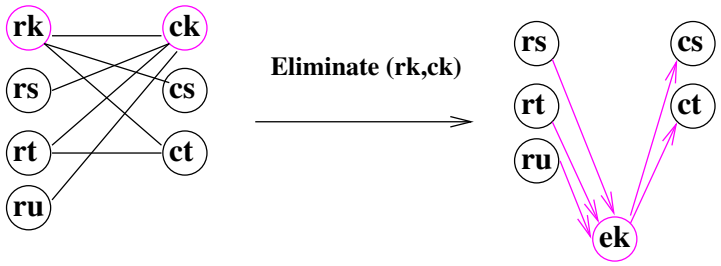


Bipartite quotient graph [Pagallo & Maulino, 1983]

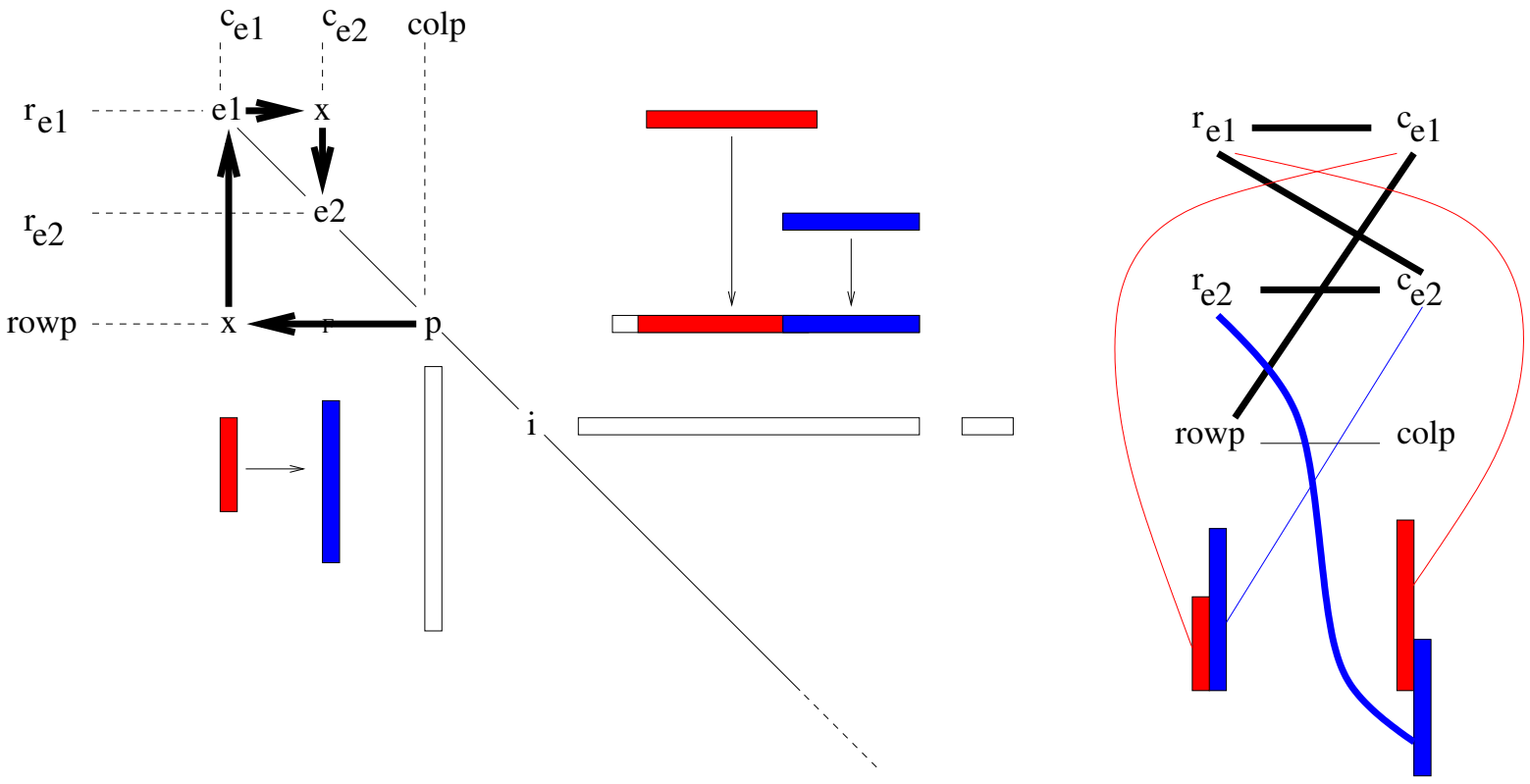
- Bipartite graphs $G^k = \{V_r^k, V_c^k, E^k\}$ to model the elimination process of $\mathbf{A}$ (with $G^0 = G(\mathbf{A})$).
- After eliminating pivot (r_k, c_k), a bi-clique is formed.



- *Memory complexity* : how to store the bi-cliques ?
  - Quotient Graph : implicit storage of the bi-clique $\longrightarrow$ in-place algorithm.
    - keep eliminated pivots (**elements**) and their adjacency lists.
    - keep edges incident with the elements that result in the bi-clique.

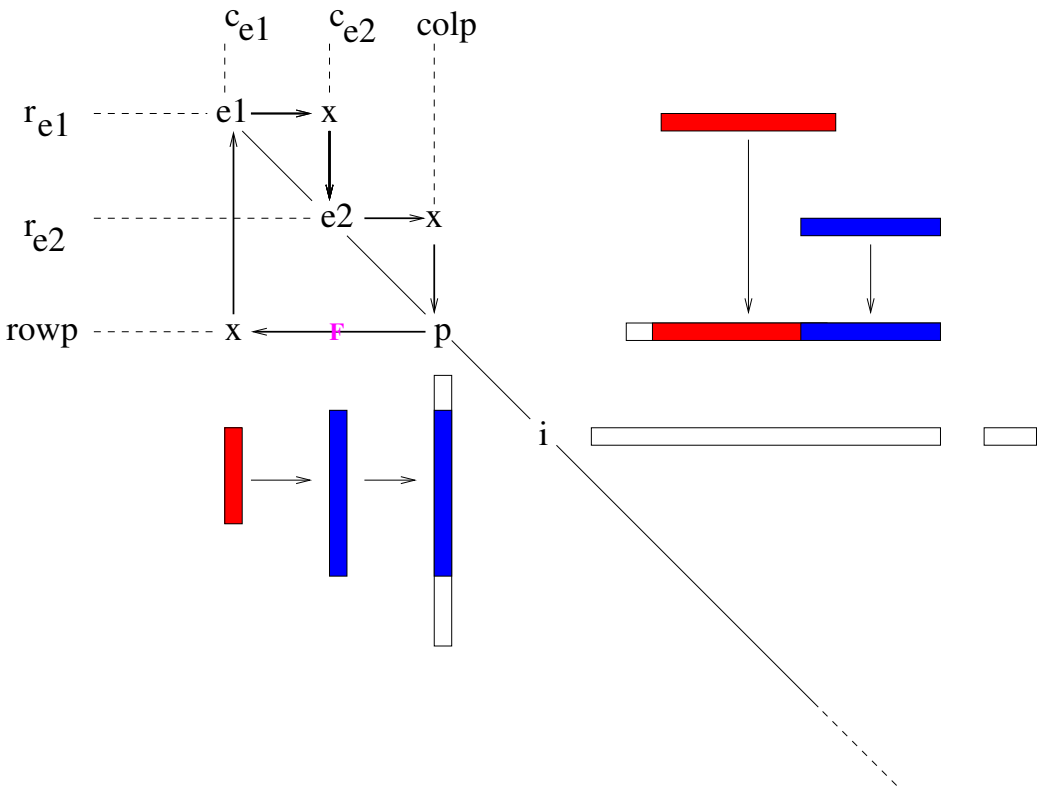


Bipartite quotient graph



- eliminates e_1: it becomes an element
- eliminates e_2: e_2 can absorb the column structure of e_1 but not the row structure
- In-place quotient graph: must search path $p \longrightarrow e_1 \longrightarrow e_2$ to determine that row p contains row structure of e_2

Strongly connected components



- after eliminating p, for the cycle (p, e_1, e_2), only 1 representative element p need be kept
  ... because if variable i is adjacent to e_1 or e_2, it must be adjacent to p

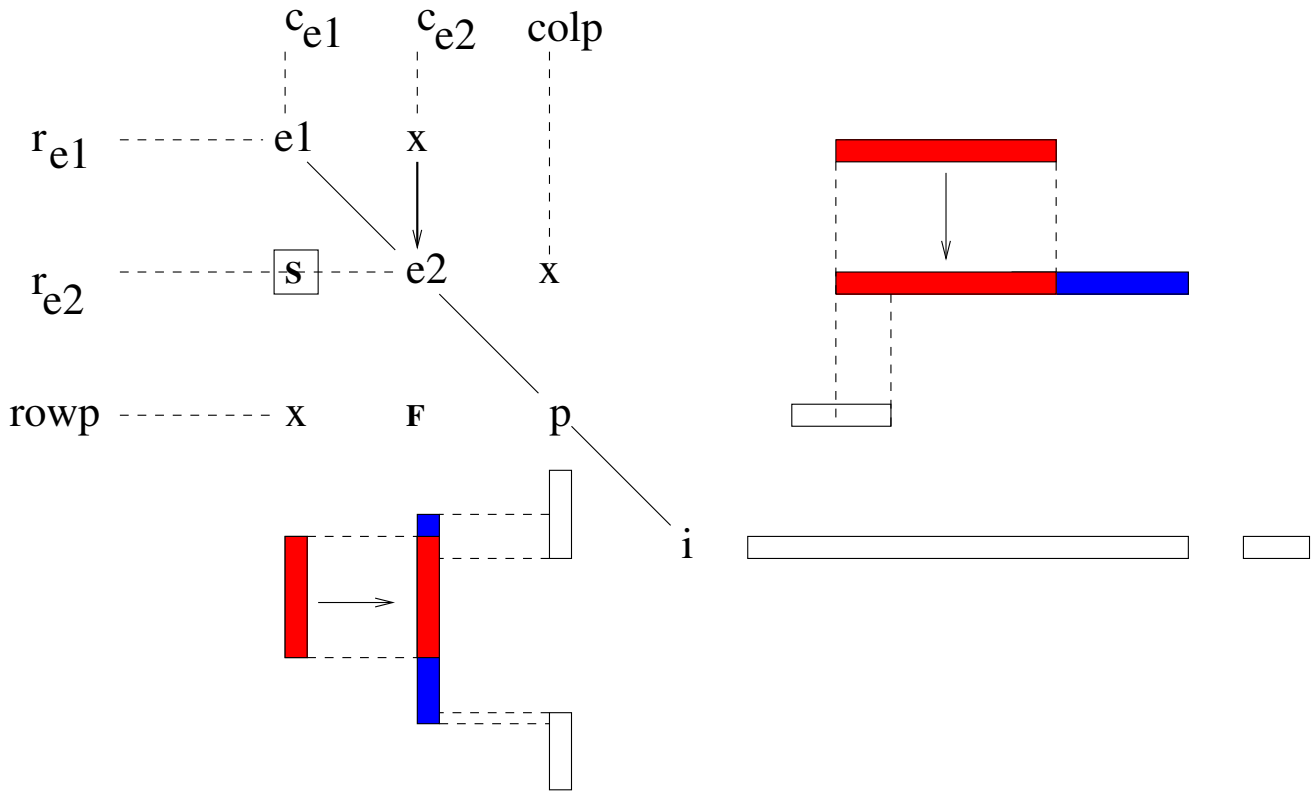
Improve runtime

Runtime cost : how to retrieve the adjacency list of a variable?

- **A** symmetric $\rightarrow$ path length ≤ 2
- **A** unsymmetric $\rightarrow$ long search path
- Local symmetrization *[Amestoy, Li, Ng, 2003]* $\rightarrow$ path length ≤ 2, but approximate structure

Quotient graph and local symmetrization

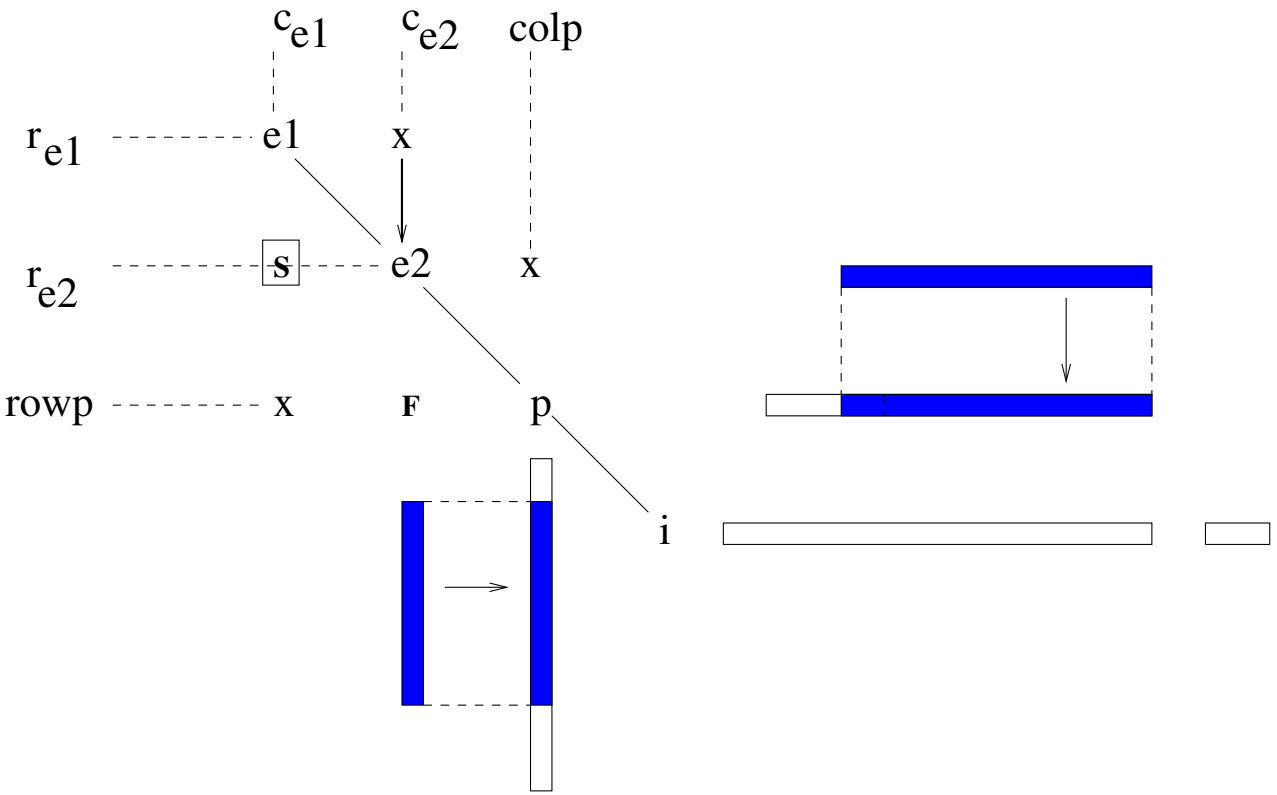
After eliminating e_2, a virtual entry **S** is added at (r_{e2}, c_{e1}), so that $e2$ can absorb $e1$.



- extra entries in the representation of the factors.
- *to select pivot p and in degree update, we must anticipate that symmetrization will be applied.*

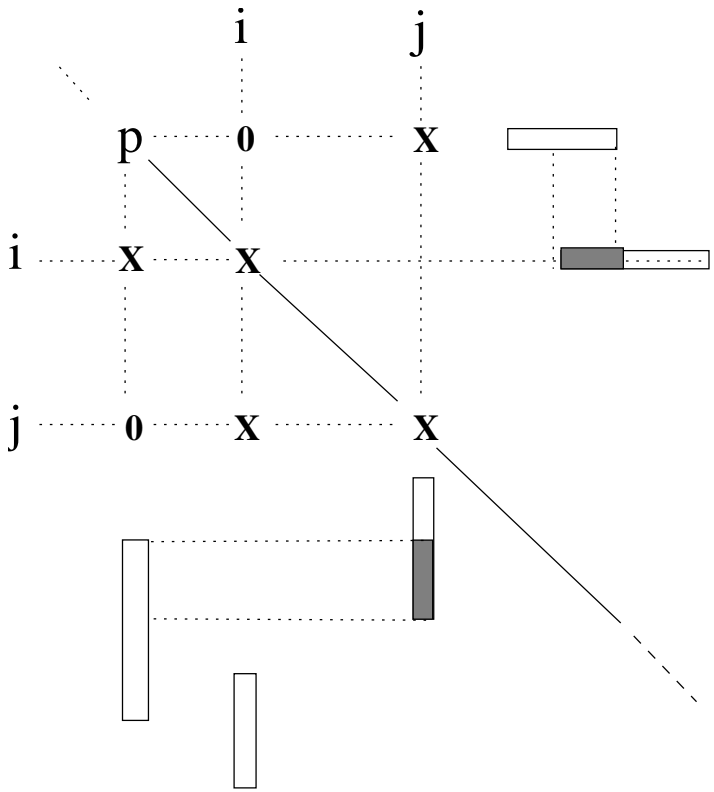
Quotient graph and local symmetrization

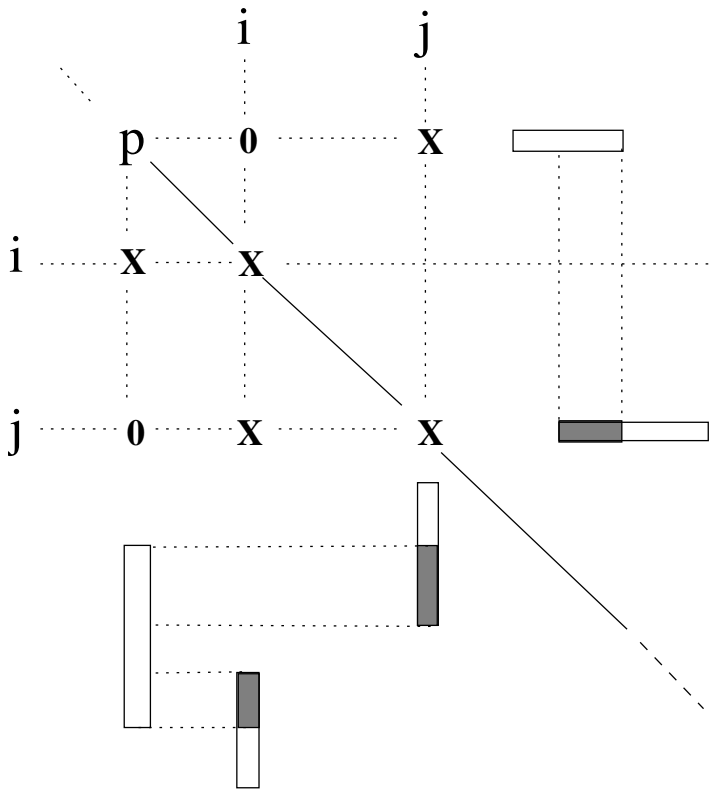
Only path $p \longrightarrow e2$ is needed to retrieve structure of p
After eliminating p, $e2$ is absorbed by p



- extra entries in the representation of the factors.
- *to select pivot p and in degree update, we must anticipate that symmetrization will be applied.*

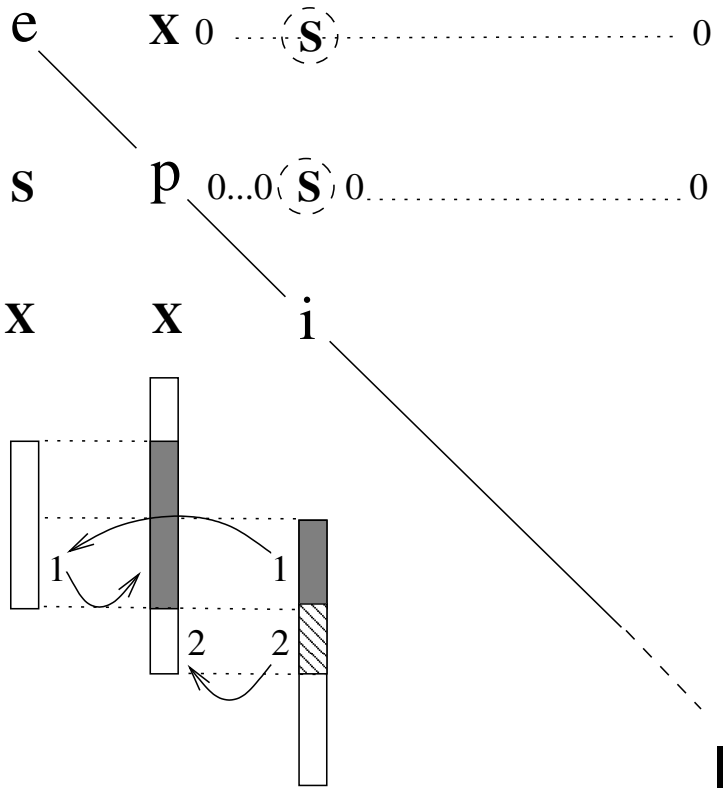
Variable elimination in one or both directions





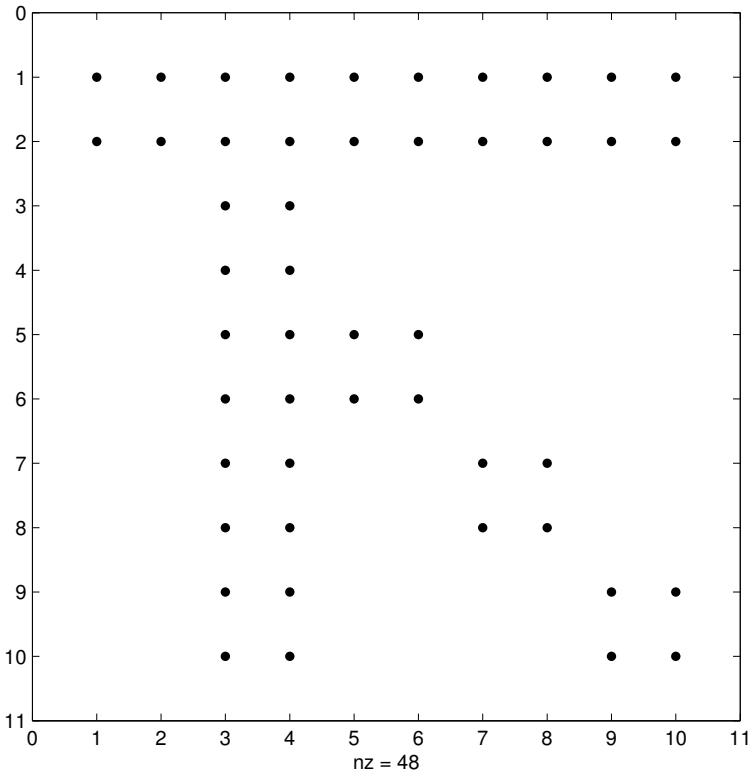
If the variable elimination is always done in one direction then the current pivot, say p, can be removed from the quotient graph when $\mathcal{U}_p = \emptyset$ or $\mathcal{L}_p = \emptyset$.

Irreducible components with eliminations in two directions

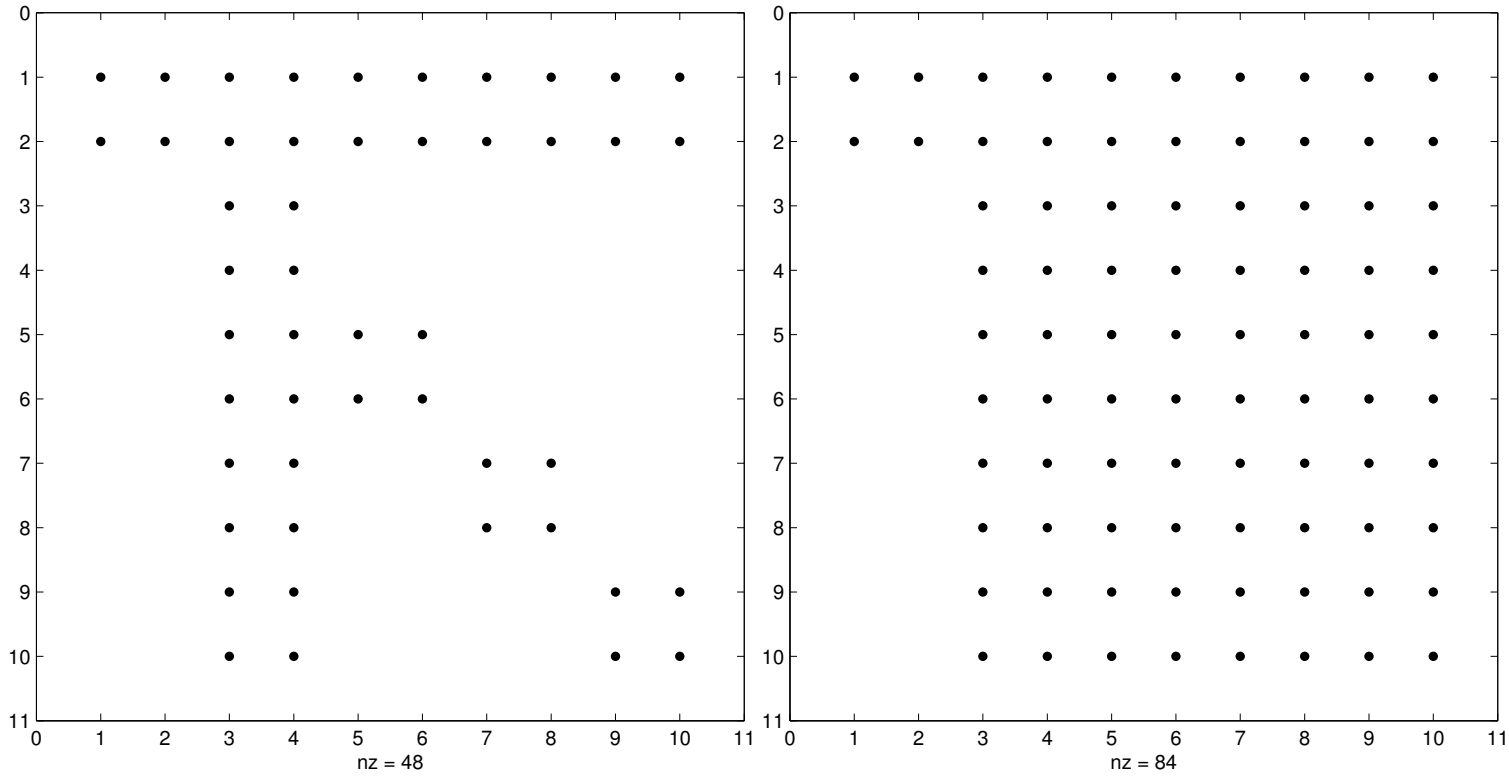


If two-way variable elimination has been done at least once, then the current pivot p can be safely removed from the quotient graph only if $\mathcal{U}_p = \emptyset$ and $\mathcal{L}_p = \emptyset$.

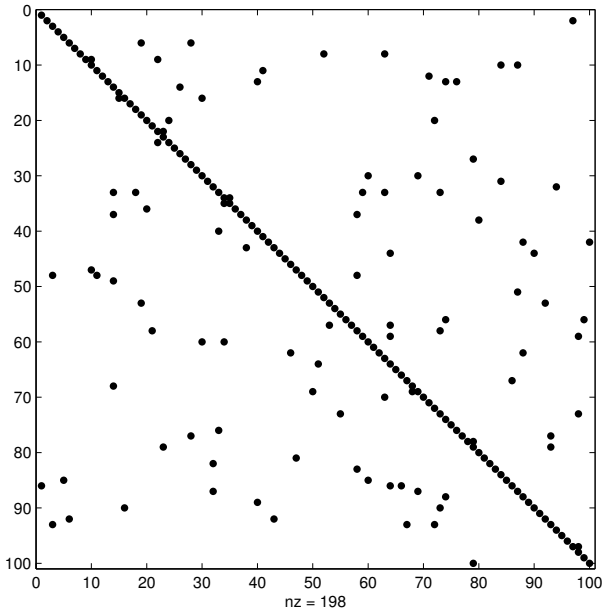
Toy example 1,



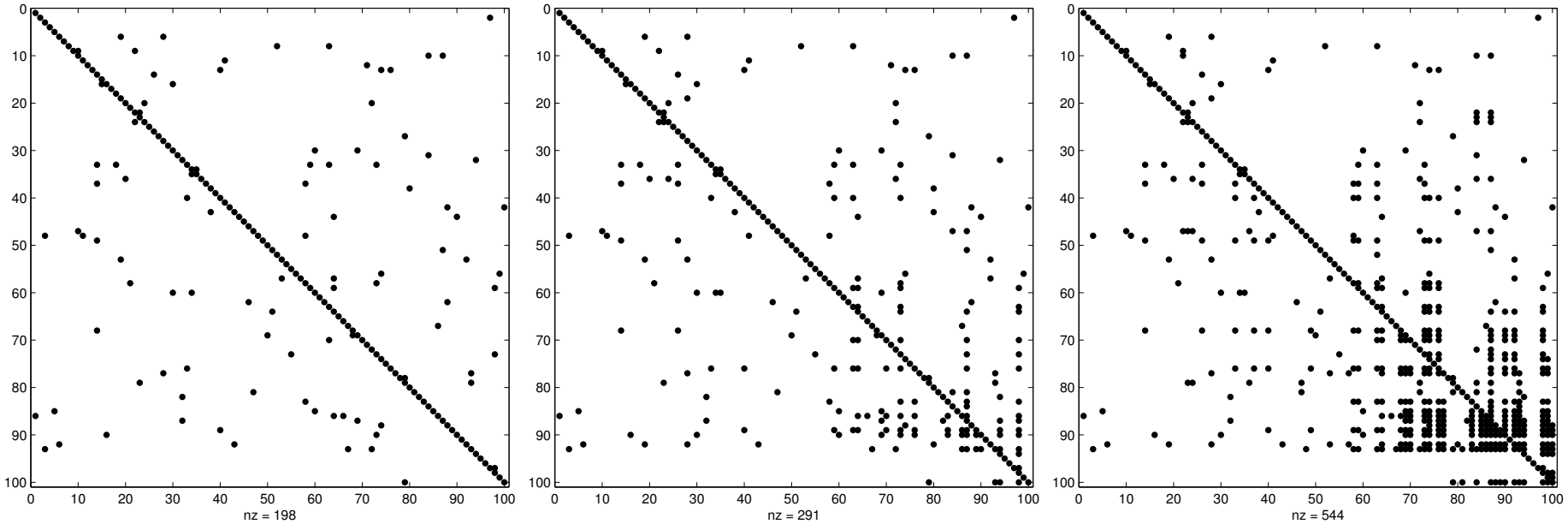
Toy example 1, after eliminations in both directions



Toy example 2,



Toy example 2, after eliminations in one (center) / both (right) directions



Real test case

mult_dcop_2, circuit simulation
n = 25187 nnz = 193276
ncomp = 7418
C = diag

- Ordering with one-way elimination:
  nnz = 446162
  Nblocks = 1303
- Ordering with two-way elimination:
  nnz = 985084
  Nblocks = 936

Which metric ?

- Structural information from $\mathbf{A}^k$: for each considered entry, CMLS can compute
  - approximate **Markowitz cost**
  - approximate **deficiency**
- Numerical information from $\mathbf{C}^k$: an entry is large enough
  - if $c_{ij} \neq 0$
  - if $|c_{ij}| \geq threshold$
  - if $|c_{ij}| \geq u \times \max c_{:j}$
  - ...

Testing environment

19 large unsymmetric matrices from Tim Davis' collection

- *symmetry* < 0.5; large dimension $11000 < n < 130000$

`MA41_UNS`: A tree-based unsymmetric multifrontal solver *[Amestoy and Puglisi, 2002]*

Phase 1 : preprocessing

`MC64` maximum weighted bipartite matching and scaling

Metric : approximate deficiency

Comparison with `DMLS` *[Amestoy, Li, Ng, 2003]*: special case of `CMLS` in which only the diagonal entries corresponding to the `MC64` matching can be selected

CMLS improves upon DMLS

If $\mathbf{C} = diag(\mathbf{AQ})$ CMLS predicts factors 7.5% sparser and with 15.7% fewer operations

CMLS improves upon DMLS

If $\mathbf{C} = diag(\mathbf{AQ})$ CMLS predicts factors 7.5% sparser and with 15.7% fewer operations

- Local symmetrization anticipation
- scaling of some AMF areas to reduce amount of tie-breaking
- distinguishing row or column supervariables $\longrightarrow$ improves accuracy of approximate deficiency

CMLS improves upon DMLS

If $\mathbf{C} = diag(\mathbf{AQ})$ CMLS predicts factors 7.5% sparser and with 15.7% fewer operations

- Local symmetrization anticipation
- scaling of some AMF areas to reduce amount of tie-breaking
- distinguishing row or column supervariables $\longrightarrow$ improves accuracy of approximate deficiency

CMLS algorithmic choices:
- not natural in a DMLS context
- could be applied to DMLS to improve it

Results: compare to DMLS, 19 matrices, mean value

- Size of factors: 13% less (0% - 43%)

Results: compare to DMLS, 19 matrices, mean value

- Size of factors: 13% less (0% - 43%)
- CMLS prediction: 9% less than actual (due to numerical pivoting)

Results: compare to DMLS, 19 matrices, mean value

- Size of factors: 13% less (0% - 43%)
- CMLS prediction: 9% less than actual (due to numerical pivoting)
- Total memory of `MA41_UNS`: 12% less (0% - 33%)

Results: compare to DMLS, 19 matrices, mean value

- Size of factors: 13% less (0% - 43%)
- CMLS prediction: 9% less than actual (due to numerical pivoting)
- Total memory of `MA41_UNS`: 12% less (0% - 33%)
- Runtime:
  - Ordering: 77% slower
  - Factorization: 16% faster
  - Solve: 10% faster

Concluding remarks

+ **sparser factors, faster factorizations**
+ more improvements with increasing size and asymmetry
+ in `MA41_UNS` framework: **reliable** pivots and memory estimations
+ in `SuperLU_DIST` framework: **improve accuracy** of the solution with fewer steps of iterative refinement
– **ordering phase slower** (future improvements)
- `CMLS` algorithmic choices can also be used to **improve** `DMLS` quality

Report: http://crd.lbl.gov/~xiaoye/LBNL-56861.pdf
also available as CERFACS report TR/PA/04/13, and
ENSEEIHT-IRIT report RT/APO/04/05
http://www.enseeiht.fr/apo/MUMPS/RT_APO_04_05.ps