

CSE 245: Computer Aided Circuit Simulation and Verification

Fall 2004, Oct 19

Lecture 7:

Matrix Solver I:

KLU: Sparse LU Factorization of Circuit

Outline

- ❑ circuit matrix characteristics
- ❑ ordering methods
 - AMD
- ❑ factorization methods
 - Gilbert & Peierls' left-looking sparse LU
- ❑ KLU: sparse LU for circuit matrices (K: "Clark Kent")
- ❑ experimental results
- ❑ Summary

- ❑ reference: [KLU: a "Clark Kent" sparse LU factorization algorithm for circuit matrices](#), a talk given by Ken Stanley at the 2004 SIAM Conference on Parallel Processing for Scientific Computing (PP04).

KLU OVERVIEW

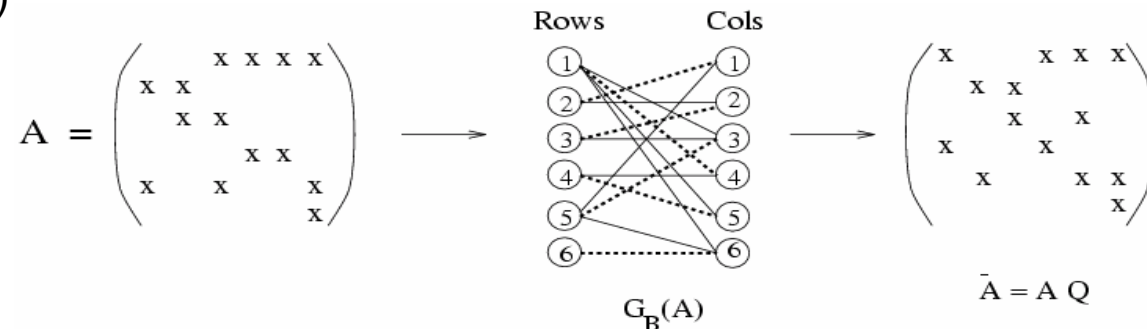
- ❑ KLU: An effective sparse matrix factorization algorithm for circuit matrices
- ❑ block triangular form (BTF)
- ❑ approximate minimum degree (AMD)
- ❑ Gilbert/Peierls' method of factorization
- ❑ very little fill-in for circuit matrices
- ❑ up to 1000 times faster than Kundert's Sparse 1.3 in SPICE

Sparse circuit matrix characteristics

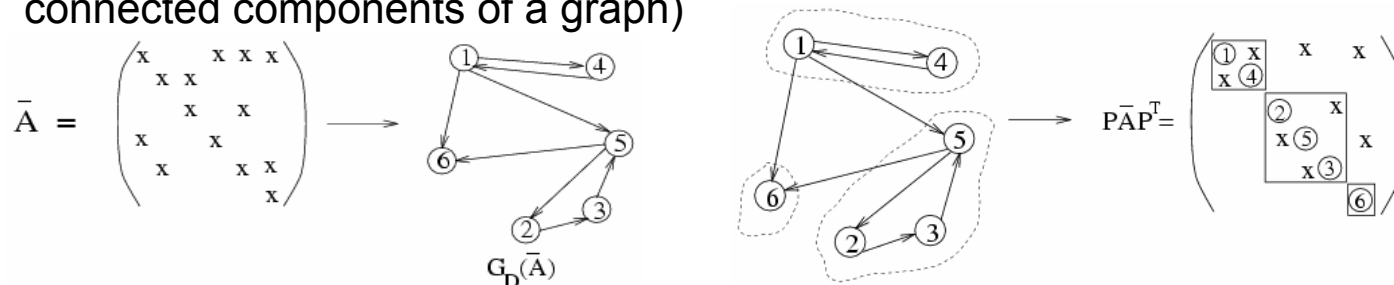
- ❑ zero-free or nearly zero-free diagonal
- ❑ unsymmetric values, pattern roughly symmetric
- ❑ permutable to block triangular form (BTF)
- ❑ very, very sparse
 - For example, for a 500x500 mesh, number of nonzero elements in the matrix is 1.2M. 99.998% sparse!!!
- ❑ if ordered properly: LU factors remain sparse (peculiar to circuit matrices)
- ❑ if ordered with typical strategies: can experience high fill-in

Ordering methods : BTF

- Unsymmetric permutation to upper block triangular form.
- MATLAB function: dmperm()
- step 1: unsymmetric permutation for a zero-free diagonal (a maximal matching graph problem)



- step 2: symmetric permutation to block triangular form (find the strongly-connected components of a graph)



- factor/solve block k, block back substitution, factor/solve block k-1..... No fill-in in off-diagonal blocks

Ordering methods : AMD

- ❑ AMD: Approximate Minimum Degree Ordering Algorithm. Finds Q to reduce fill-in for the Cholesky factorization of $Q^T A Q$ (or of $Q^T (A + A^T) Q$ if A unsymmetric). Which limits fill-in for LU Factorization of $Q^T A Q$ assuming no numerical pivoting.
- ❑ pre-ordering: find Q to control fill-in of Cholesky of $Q^T (A + A^T) Q$. This Q will limit the fill-in for LU factorization of $Q^T A Q$.
- ❑ numerical factorization: try to select $P = Q$ (constrained partial pivoting)
- ❑ Result: $PAQ = LU$
- ❑ AMD reduces an *optimistic* estimate of fill-in.

Factorization methods : Gilbert/Peierls

- left-looking. k th stage computes k th column of L and U

$L = I$

$U = I$

for $k = 1:n$

$s = L \setminus A(:,k)$

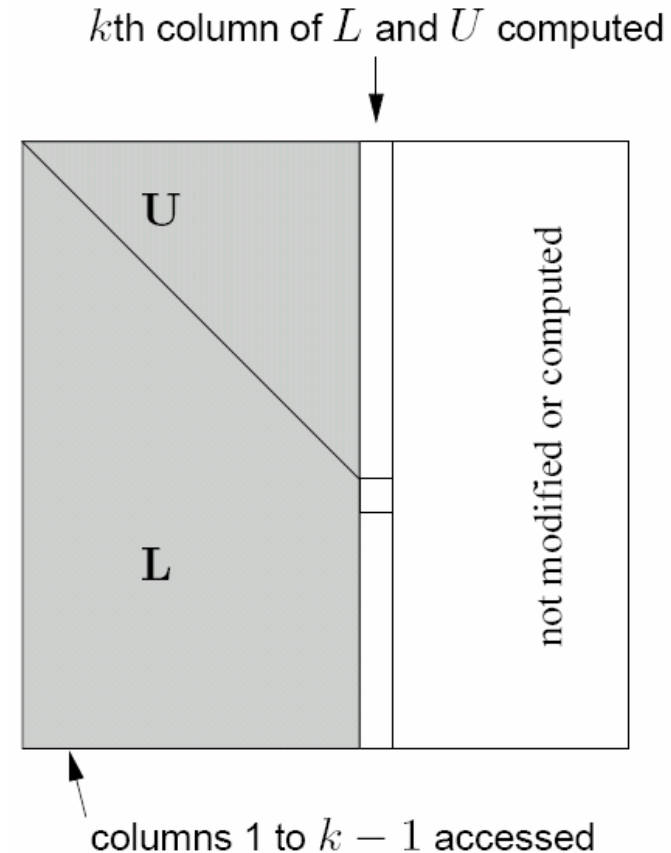
 (partial pivoting on s)

$U(1:k,k) = s(1:k)$

$L(k:n,k) = s(k:n) / U(k,k)$

end

- key step: solve $Lx = b$,
where L is sparse, and b is sparse



- Reference: J. R. Gilbert and T. Peierls, *Sparse partial pivoting in time proportional to arithmetic operations*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 862-874

Factorization methods : Gilbert/Peierls

□ Solving $Lx = b$, need nonzero pattern of x to compute x

□ $x = b$

for $j = 1:n$

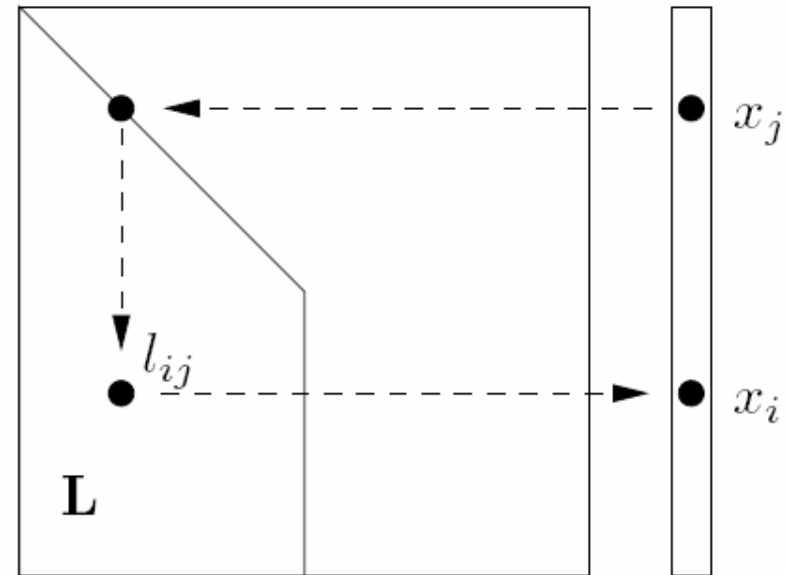
if $(x(j) \neq 0)$

$x(j+1:n) = x(j+1:n) - \dots$

$L(j+1:n,j) * x(j)$

end

end



● $b_i \neq 0 \rightarrow x_i \neq 0$

● $x_j \neq 0 \wedge l_{ij} \neq 0 \rightarrow x_i \neq 0$

● let $G(L)$ have an edge $j \rightarrow i$ if $l_{ij} \neq 0$

● let $\mathcal{B} = \{i \mid b_i \neq 0\}$ and $\mathcal{X} = \{i \mid x_i \neq 0\}$

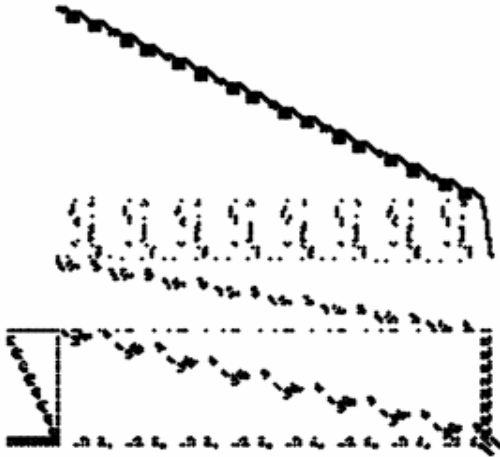
● then $\mathcal{X} = \text{Reach}_{G(L)}(\mathcal{B})$

KLU: a “Clark Kent” LU

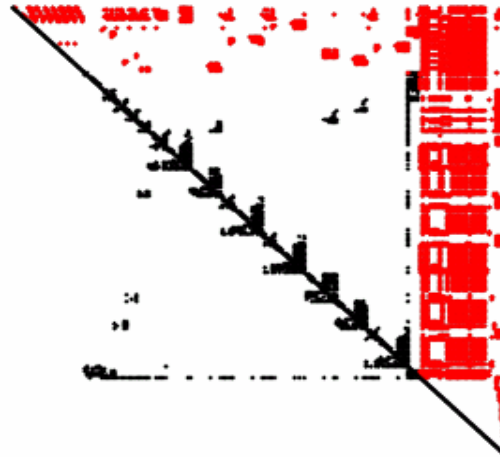
- ❑ Gilbert/Peierl's: used with unsymmetric preordering in MATLAB, so unsuitable for circuits
- ❑ KLU: a “Clark Kent” LU
- ❑ BTF ordering
- ❑ AMD ordering of each block
- ❑ Gilbert/Peierls' LU factorization of each block
- ❑ partial pivoting with diagonal preference
- ❑ refactorization can allow for partial pivoting

Results: Grund/meg1, n: 2902, non-zeros: 58,142

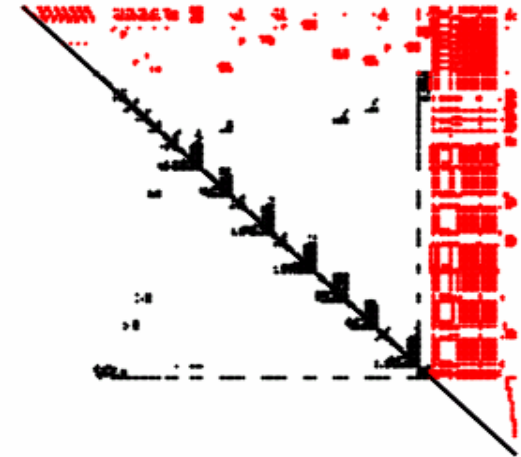
original matrix



preordering



factorization

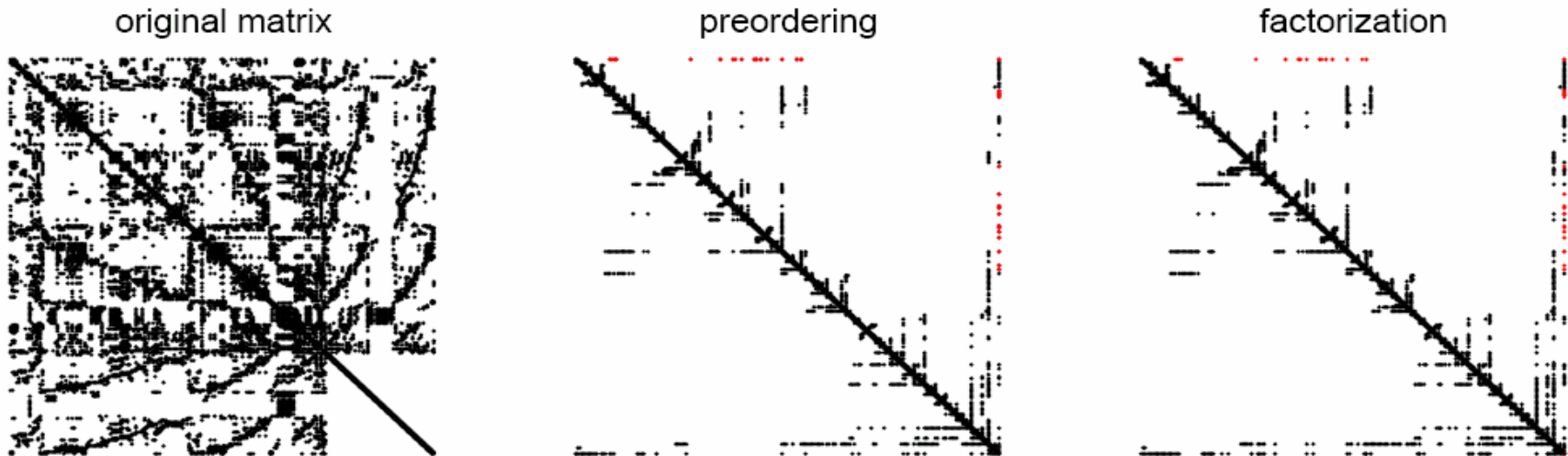


- largest block: 1906-by-1906
- total # nonzeros in diagonal blocks: 27,575
- one large block, the rest 1-by-1. BTF is very important

Results: Grund/meg1, n: 2902, non-zeros: 58,142

method	factorization time (sec)			nz L+U (10 ³)	# flop (10 ⁶)
	1st	2nd w/piv.	2nd no piv.		
KLU	.0167	.0072	0.0046	73.0	0.7
no BTF:					
MATLAB,GP+AMD	.4833	.4000	-	522.7	61.9
UMFPACK	.0355	.0252	-	80.4	1.0
Sparse 1.3	15.2600	-	1.0000	199.1	16.7
with BTF:					
MATLAB,GP+AMD	.0281	.0142	-	73.0	0.7
UMFPACK	.0473	.0259	-	76.6	0.6
Sparse 1.3	1.0500	-	.0500	90.5	1.9

Results: Hamm/scircuit, n: 170,988, nnz: 958,936



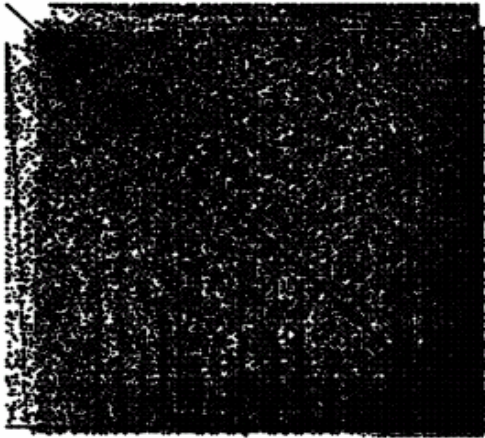
- largest block: 170,493-by-170,493
- total # nonzeros in diagonal blocks: 958,288
- essentially a single block, BTF has little effect

Results: Hamm/scircuit, n: 170,988, nz: 958,936

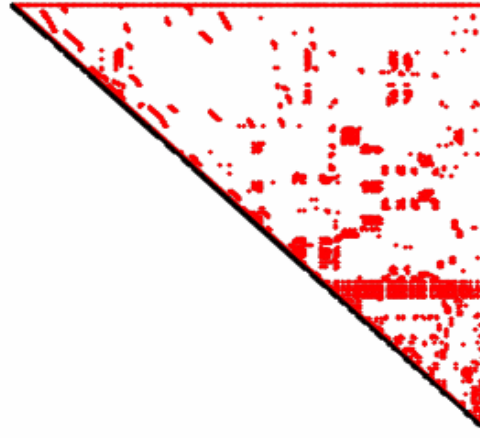
method	factorization time (sec)			nz L+U (10 ³)	# flop (10 ⁶)
	1st	2nd w/piv.	2nd no piv.		
KLU	1.03	.44	.28	2507	65.3
no BTF:					
MATLAB,GP+AMD	1.73	.94	-	2516	65.3
UMFPACK	2.98	1.93	-	3734	214.0
Sparse 1.3	126.92	-	3.23	2478	64.3
with BTF:					
MATLAB,GP+AMD	1.75	.94	-	2507	65.3
UMFPACK	3.19	2.14	-	4133	270.0
Sparse 1.3	116.58	-	3.10	2471	62.2

Results: Sandia/mult_dcop_03 , n: 25187, nz: 193,216

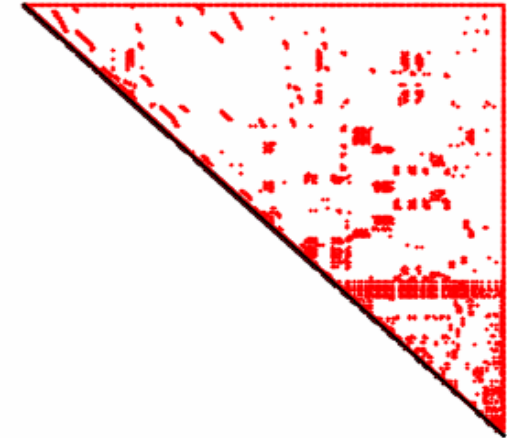
original matrix



preordering



factorization



- largest block: 70-by-70
- total # nonzeros in diagonal blocks: 80,038
- many small blocks, BTF is very important

Results: Sandia/mult_dcop_03 , n: 25187, nz: 193,216

method	factorization time (sec)			nz L+U (10 ³)	# flop (10 ⁶)
	1st	2nd w/piv.	2nd no piv.		
KLU	.545	.040	.020	201	.16
no BTF:					
MATLAB,GP+AMD	.487	.247	-	446	4.83
UMFPACK	.440	.257	-	372	2.56
Sparse 1.3	84.680	-	14.120	294	.90
with BTF:					
MATLAB,GP+AMD	.483	.113	-	201	.16
UMFPACK	.660	.255	-	200	.15
Sparse 1.3	.337	-	.017	202	.16

Summary

- ❑ KLU: An effective sparse matrix factorization algorithm for circuit matrices
- ❑ block triangular form
 - However, for circuit matrix, it is hard to permute to Block Triangular Form
- ❑ approximate minimum degree (AMD)
- ❑ Gilbert/Peierls' method of factorization
 - Much faster than the factorization method used in SPICE.