LUSOL and its uses in optimization

Sparse Days at CERFACS

Toulouse, France

June 15–16, 2006

Michael Saunders

Systems Optimization Laboratory (SOL)
Dept of Management Science & Engineering
Stanford University
saunders@stanford.edu

Abstract

LUSOL maintains LU factors of sparse matrices of any shape (square or rectangular). Threshold Rook Pivoting provides a reliable rank-revealing capability, shared by MA27 and MA57. (Threshold Complete Pivoting is also an option.) Updates are stabilized by the Bartels-Golub approach.

We review the open source Fortran and C implementations of LUSOL and their use within the optimization packages MINOS, SQOPT, SNOPT, PATH, ZIP, and lp_solve.

LU factors are often used to construct preconditioners.
We emphasize the need for one factor to be well-conditioned, and the need to know which one! (In this case it is L.)

LUSOL

Maintaining LU factors of a general sparse matrix A

Gill, Murray, Saunders, and Wright (1987)

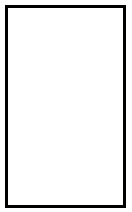
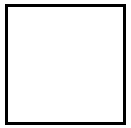
Code contributors

F77	Saunders (1986–present)
	following Duff, Reid, Zlatev, Suhl and Suhl
MATLAB Fmex	Michael O'Sullivan (1999–present)
C (for lp_solve)	Kjell Eikland (2004–present)
MATLAB Cmex	Yin Zhang (2005–present)

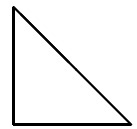
Features

Square or rectangular A	for basis selection, preconditioning
Rank-revealing LU	for "basis repair"
Stable updates	Bartels-Golub-Reid style

LUSOL

A =  or  or  = LU

FACTOR [L,U,p,q] = luSOL(A)

L(p,p) = 

L well-conditioned

UPDATE Add, replace, delete a column
Add, replace, delete a row
Add a rank-one matrix

$L \leftarrow LM_1M_2\ldots$

M_j well-conditioned

SOLVE $Lx = y,\ L^Tx = y,\ Ux = y,\ U^Tx = y,\ Ax = y,\ A^Tx = y$

MULTIPLY $x = Ly,\ x = L^Ty,\ x = Uy,\ x = U^Ty,\ x = Ay,\ x = A^Ty$

FACTOR

LU factors of a vector

$$\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} a & & & \\ b & 1 & & \\ c & & 1 & \\ d & & & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & & & \\ b/a & 1 & & \\ c/a & & 1 & \\ d/a & & & 1 \end{pmatrix} \begin{pmatrix} a \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Keep L well-conditioned $\Rightarrow$ $|a|$ not too small

LU factors of two vectors

$$\begin{pmatrix} a & e \\ b & f \\ c & g \\ d & h \end{pmatrix} = \begin{pmatrix} 1 & & \\ b/a & 1 & \\ c/a & & 1 \\ d/a & & & 1 \end{pmatrix} \begin{pmatrix} a & e \\ 0 & f-(b/a)e \\ 0 & g-(c/a)e \\ 0 & h-(d/a)e \end{pmatrix}$$

$$A = L \qquad U$$

Forward substitution: "$L\,x = b$"

Forward sub gives 2nd col of U: $L\,U_2 = A_2$

Permute rows and columns to preserve stability

Gaussian elimination

$$\begin{pmatrix} 1 & \\ -\mu & 1 \end{pmatrix} \begin{pmatrix} a & c & e \\ b & d & f \end{pmatrix} = \begin{pmatrix} a & c & e \\ & d-\mu c & f-\mu e \end{pmatrix}$$

LU factorization (forward substitution)

$$\begin{pmatrix} a & c & e \\ b & d & f \end{pmatrix} = \begin{pmatrix} 1 & \\ \mu & 1 \end{pmatrix} \begin{pmatrix} a & c & e \\ & d-\mu c & f-\mu e \end{pmatrix}$$

$$L = \begin{pmatrix} 1 & \\ 100 & 1 \end{pmatrix}$$

$$\mathrm{cond}(L) \approx 100\,??$$

$$L = \begin{pmatrix} 1 & \\ 100 & 1 \end{pmatrix}$$

$$\mathrm{cond}(L) \approx 10000$$

$$L = \begin{pmatrix} 1 & \\ 100 & 1 \end{pmatrix}$$

$$\mathrm{cond}(L) \approx 10000$$

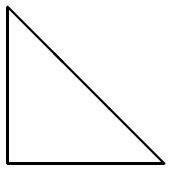
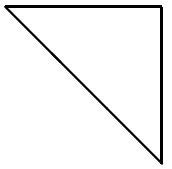
Fortunately, triangular solves

$$\begin{pmatrix} 1 & \\ \mu & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} b \\ e \end{pmatrix}$$

behave as if $\mathrm{cond}(L) \leq 100$ (Wilkinson, 1963)

RANK-REVEALING LU

FACTOR

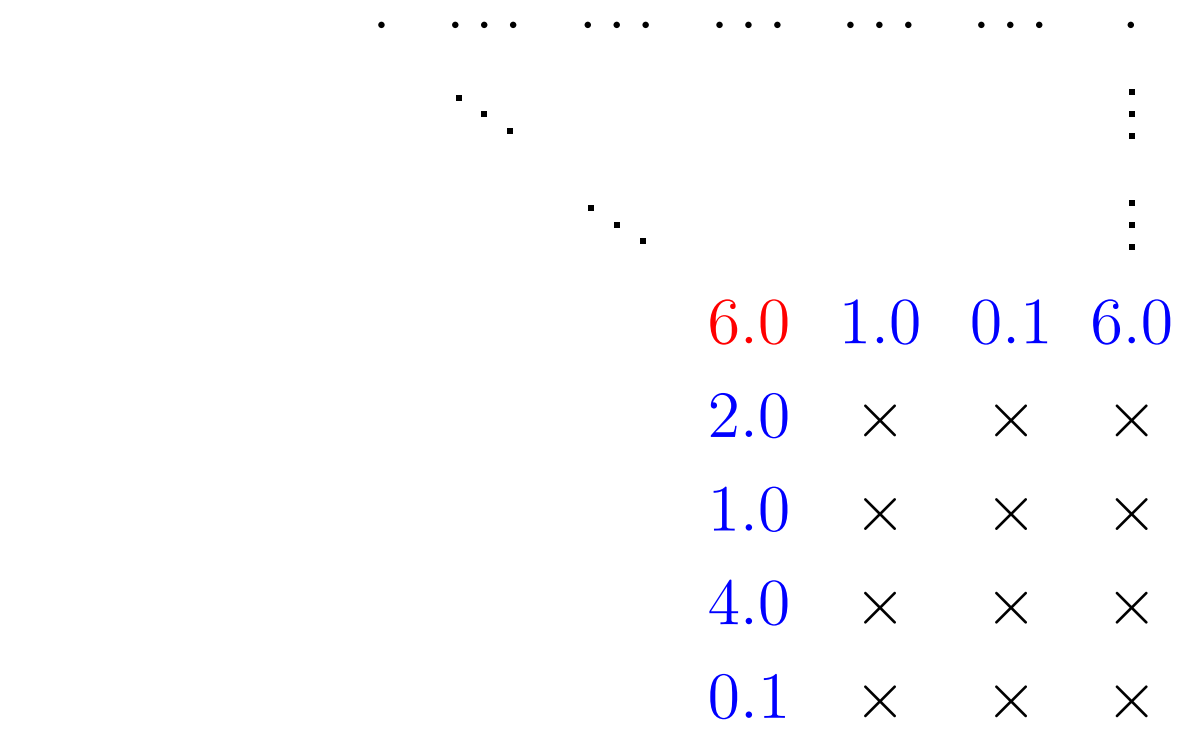
[L,U,p,q] = luSOL(A) L(p,p) =  U(p,q) = 

- Well defined for **any square or rectangular** A
- Permutations p, q balance **sparsity** and **stability**
- **Sparsity** achieved by Markowitz pivot strategy
  à la MA28, LA05, LA15
- **Stability** options:
  - TPP Threshold Partial Pivoting
  - TRP Threshold Rook Pivoting
  - TCP Threshold Complete Pivoting

Partial Pivoting

$$
\begin{array}{ccccccc}
\cdot & \cdots & \cdots & \cdots & \cdots & \cdots & \cdot \\
 & \ddots & & & & & \vdots \\
 & & \ddots & & & & \vdots \\
 & & & 4.0 & \times & \times & \times \\
 & & & 2.0 & \times & \times & \times \\
 & & & 1.0 & \times & \times & \times \\
 & & & 4.0 & \times & \times & \times \\
 & & & 0.1 & \times & \times & \times
\end{array}
$$

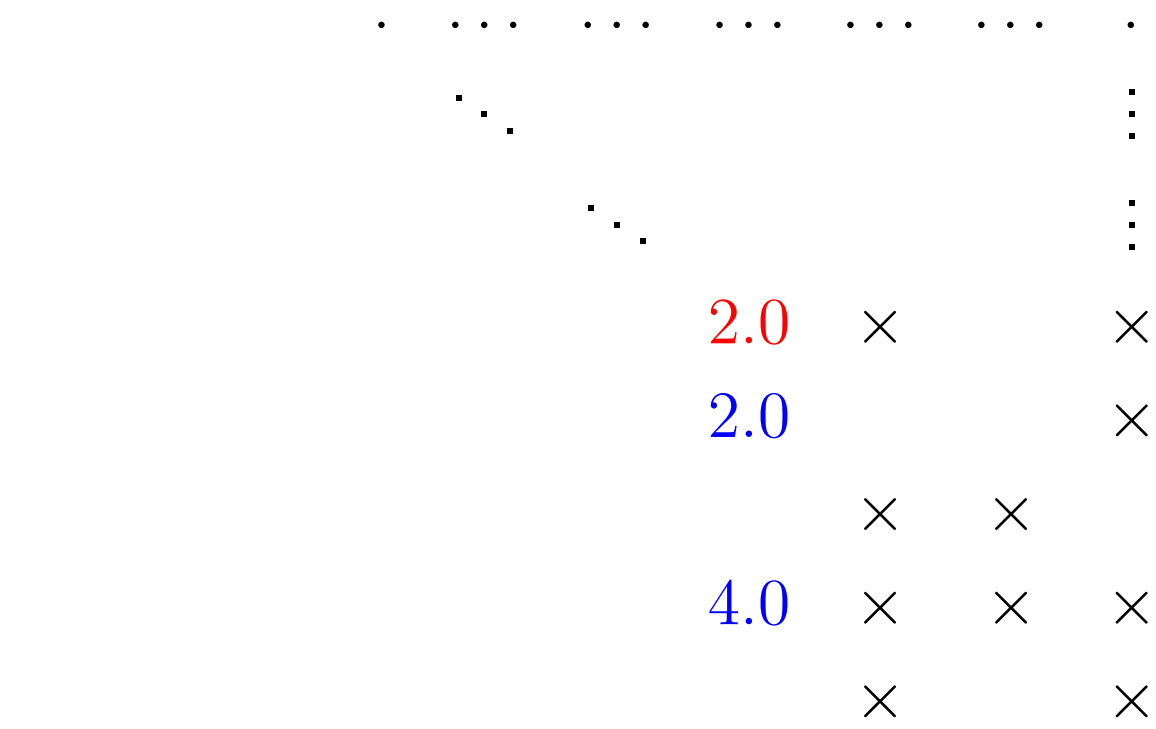
Rook Pivoting



Complete Pivoting

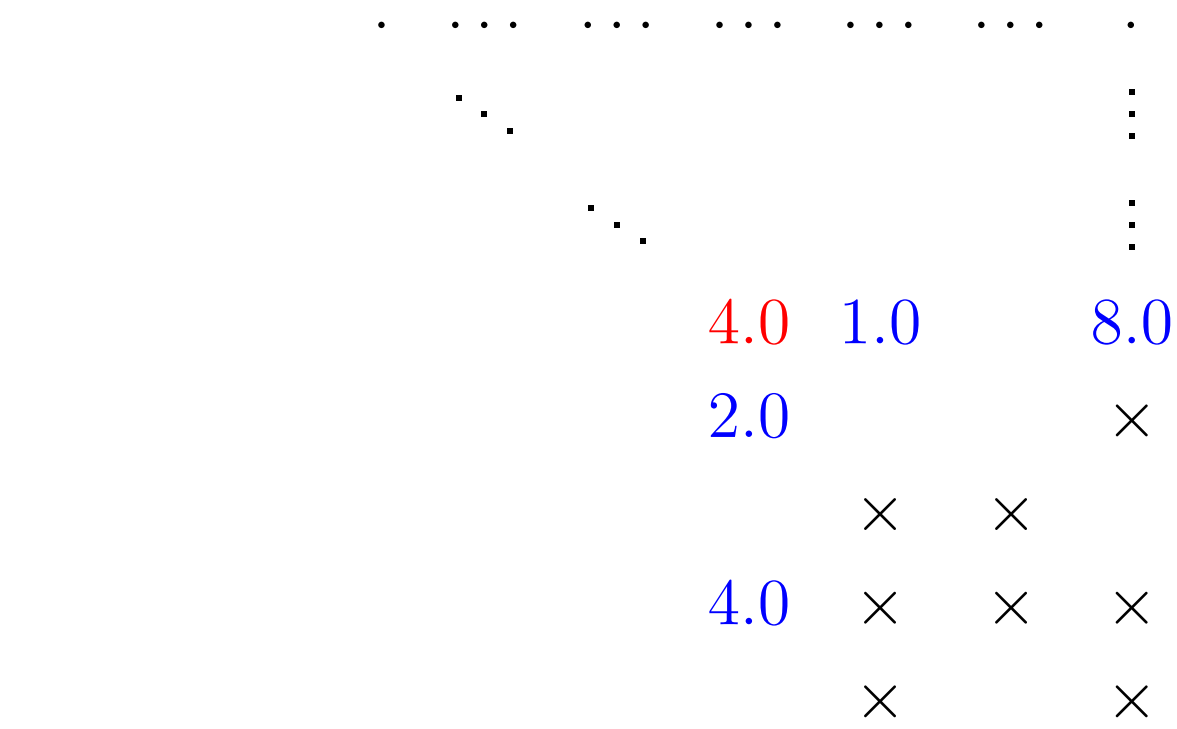
$$
\begin{array}{ccccccc}
\cdot & \cdots & \cdots & \cdots & \cdots & \cdots & \cdot \\
 & \ddots & & & & & \vdots \\
 & & \ddots & & & & \vdots \\
 & & & 9.0 & 1.0 & 0.1 & 6.0 \\
 & & & 2.0 & \times & \times & \times \\
 & & & 1.0 & \times & \times & \times \\
 & & & 4.0 & \times & \times & 9.0 \\
 & & & 0.1 & \times & 0.1 & \times
\end{array}
$$

TPP: Threshold Partial Pivoting

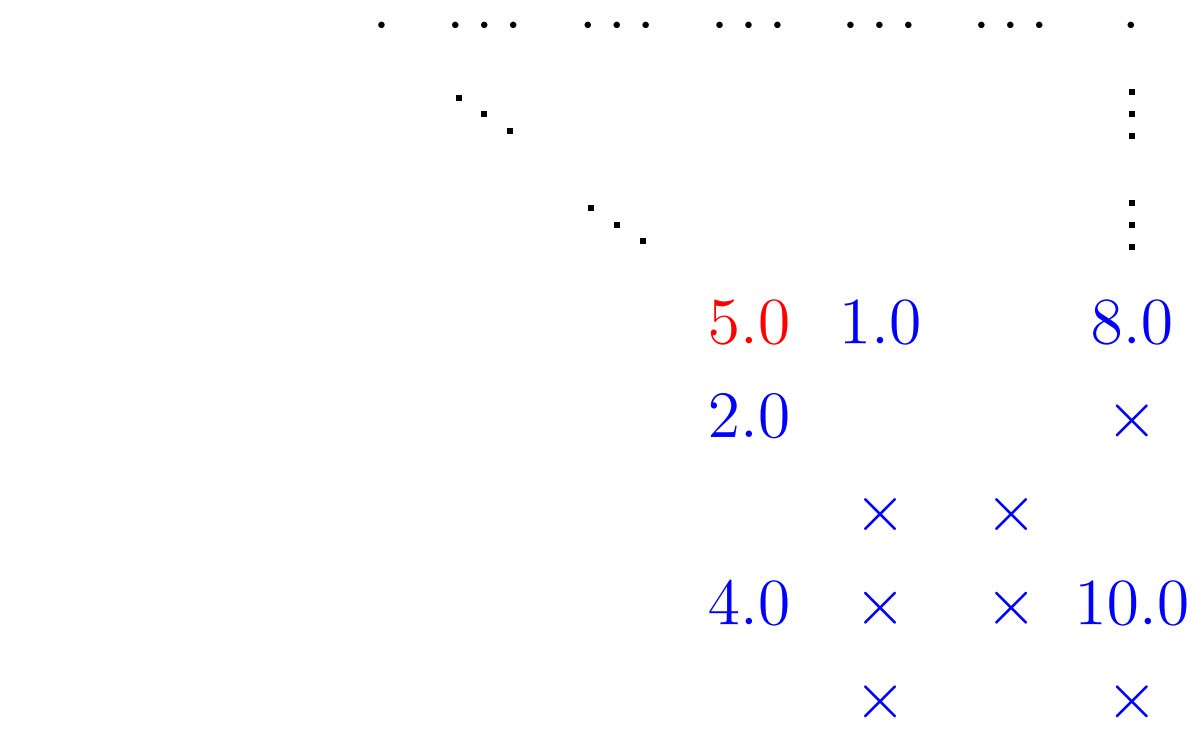


Require $|L_{ij}| \leq 2.0$ say (not 1.0)

TRP: Threshold Rook Pivoting

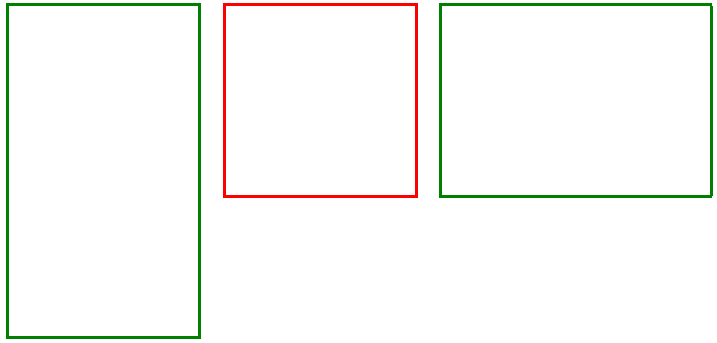


TCP: Threshold Complete Pivoting



Rank-Revealing Factors

$$A = XDY^T =$$



X, Y well-conditioned

$\mathrm{cond}(A) \approx \mathrm{cond}(D)$

- SVD UDV^T
- QR with column interchanges QDR
- LU with Rook Pivoting LDU
- LU with Complete Pivoting LDU
- QLP $A\Pi = QR = QLP^T$

Stability tolerance τ

$$PAQ = LDU$$

Threshold pivoting bounds elements of L and/or U:

$$\left.\begin{array}{l}\text{TPP}\end{array}\right\} \quad |L_{ij}| \leq \tau \approx 100 \text{ or } 10 \text{ or } 5$$

$$\left.\begin{array}{l}\text{TRP}\\ \text{TCP}\end{array}\right\} \quad |L_{ij}|, |U_{ij}| \leq \tau \approx 3 \text{ or } 2 \text{ or } 1.1$$

TRP, TCP are more **Rank-Revealing** with low τ:

$$\begin{aligned} \text{cond}(L),\ \text{cond}(U) &< (1+\tau)^n \\ \text{cond}(D) &\approx \text{cond}(A) \end{aligned}$$

The need for rank-revealing LU

$$A = \begin{pmatrix} \delta & 1 & 1 & 1 \\ & \delta & 1 & 1 \\ & & \delta & 1 \\ & & & \delta \end{pmatrix} = LU \qquad \delta \text{ small}$$

TPP would give $L = I, \quad U = A, \quad \operatorname{rank}(A) = 4$ or 0 (!)

TRP or TCP would give

$$\begin{pmatrix} 1 & 1 & 1 & \delta \\ \delta & 1 & 1 & \\ & \delta & 1 & \\ & & \delta & \end{pmatrix} \approx L \begin{pmatrix} 1 & 1 & 1 & \delta \\ & 1 & 1 & -\delta^2 \\ & & 1 & \delta^3 \\ & & & -\delta^4 \end{pmatrix} \qquad \operatorname{rank}(A) \approx 3$$

Implementing TRP

At each stage of Gaussian elimination: $A \leftarrow A - lu^T$

We need

α_j = biggest element in col j

$$\begin{bmatrix} \alpha_j \\ \text{x} \\ \text{x} \end{bmatrix}$$

β_i = biggest element in row i

$$\begin{bmatrix} \beta_i & \text{x} & \text{x} \end{bmatrix}$$

Implementing TCP

At each stage of Gaussian elimination: $A \leftarrow A - lu^T$

We need

α_j = biggest element in col j

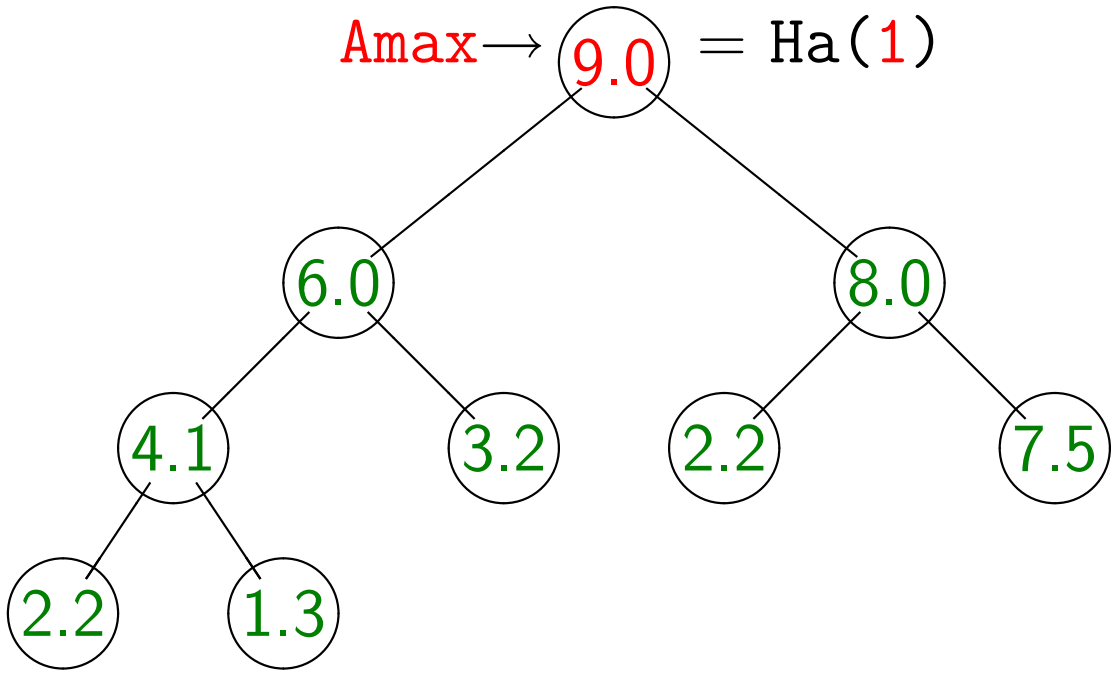
$$\begin{bmatrix} \alpha_j \\ \text{x} \\ \text{x} \end{bmatrix}$$

Amax = biggest element in A

$= \max \alpha_j$

TCP: Store α_j in a Heap

Thanks to John Gilbert



α_j	=	Ha(k)	9.0	6.0	8.0	...
j	=	Hj(k)	2	17	1	...
k	=	Hk(j)	3	1	15	... (location of j in heap)

FACTOR RESULTS

Problem memplus from Harwell-Boeing collection

$A = 18000 \times 18000$, 126000 nonzeros, Scaled

		τ	nnz(L+U)	Time
TCP		100.0	140000	2
		10.0	579000	30
	RR	3.99	2460000	475
	RR	2.5	2890000	6610
	RR	1.5	7080000	27875
TRP		100.0	142000	5
		10.0	141000	5
	RR	3.99	142000	5
	RR	2.50	146000	6
	RR	1.99	166000	7
	RR	1.58	172000	7
	RR	1.26	174000	8
PP (SuperLU, colamd)		1.0	4470000	≈ 250

TCP Profile

Harwell-Boeing Problem memplus

$A = 18000 \times 18000$, 126000 nonzeros

TCP, $\tau = 10.0$, LU = 578000 nonzeros

Markowitz	Find stable pivot	65.0%
Dense CP	600×600	18.5%
Elimination	The algebra	7.4%
Update α_j	for modified cols	4.7%
Update heap		0.1% (!)

Tracking $\max |A_{ij}|$ is easy!

TRP Profile

CUTE Problem BRATU2D

$A = 4900 \times 4900$, 24000 nonzeros

TRP, $\tau = 1.26$, LU = 206000 nonzeros

Update β_i	for modified rows	57.7%
Markowitz	Find stable pivot	31.4%
Elimination	The algebra	4.0%
Dense CP	228×228	2.0%
Update α_j	for modified cols	1.6%

Symmetric indefinite systems

$$A = LDL^T, \qquad D = \begin{pmatrix} \square & & & & & \\ & \square & & & & \\ & & \square & & & \\ & & & \square & & \\ & & & & \square & \\ & & & & & \ddots \end{pmatrix} \qquad (1 \times 1, \quad 2 \times 2)$$

- Bunch-Parlett, Bunch-Kaufman strategies don't bound $|L_{ij}|$
- Duff and Reid, Harwell Subroutine Library:

  MA27, MA57 **do bound** $|L_{ij}|$

  Equivalent to Threshold Rook Pivoting

  Hence: MA27, MA57 are rank-revealing with $|L_{ij}| \leq 2$, say

SOLVE

SOLVE

Dense rhs

- Currently, $Lx = y$, $L^T x = y$, ... assume rhs y is dense

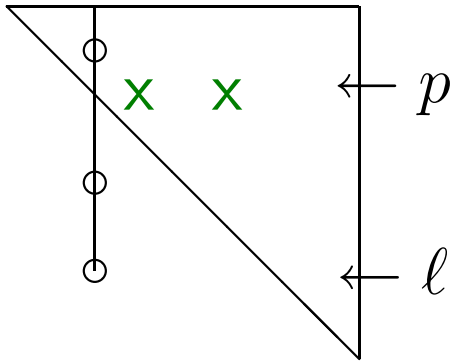
Sparse rhs (future)

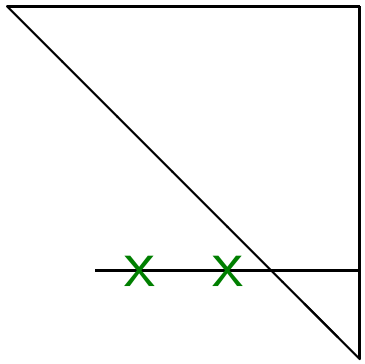
- Gilbert and Peierls (1988), CPLEX 7.1 (2001), MA48
- $Lx = y$ requires L **column-wise**
- $L^T x = y$ needs second copy of L (**row-wise**)
- Similarly for U, U^T (not good when updates modify U)

UPDATE

Bartels-Golub updates

à la Reid 1976, 1982, 2004 LA05, LA15

$U' =$ 

$U'' \equiv P^T U' P =$ 

- Avoid Hessenberg matrix
- Use **cyclic permutation** P
- Eliminate x using $M_j = \begin{pmatrix} 1 & \\ \mu & 1 \end{pmatrix}$ or $\begin{pmatrix} 1 & \\ \mu & 1 \end{pmatrix} \begin{pmatrix} & 1 \\ 1 & \end{pmatrix}$

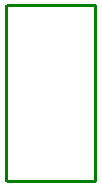
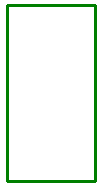
Block-LU updates

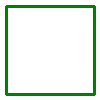
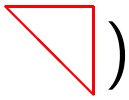
Part of Hanh Huynh's thesis

Replacing cols or rows is equiv to solving with a bordered system:

$$\begin{pmatrix} B_0 & V \\ W^T & D \end{pmatrix} = \begin{pmatrix} L_0 & \\ Z^T & I \end{pmatrix} \begin{pmatrix} U_0 & Y \\ & C \end{pmatrix}$$

$L_0 Y = V, \qquad {U_0}^T Z = W$

$Y =$  and $Z =$  are likely to be **sparse**

LUMOD maintains dense $LC = U$ $\quad (L =$  $\quad U =$  $)$

Similarly for KKT systems, cf. GALAHAD's QPA

APPLICATIONS

- **Rank detection**
- **Preconditioning**
- **Null-space basis**

LUSOL in MINOS and SQOPT

Rank detection for square B

$B = \square = LU = LD\bar{U}$ — L and $\bar{U}$ well-conditioned

TRP or TCP — Small D_{ii} point to singularities

Basis detection for rectangular $A = (B\ S)$

$A^T = \square = LU,$ — L well-conditioned

TPP, TRP or TCP — New $(\bar{B}\ \bar{S}) = A(:, p)$

Just need row permutation p

LU preconditioning

$A = pt$  $= LU,$ L well-conditioned

Least squares: $\min \|Ax - b\|$

$\min \|Ly - b\|$ Apply LSQR

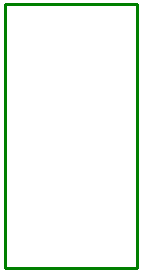
Solve $Ux = y$

Min-length solution: $\min \|x\|$ st $A^T x = b$

Solve $U^T c = b$

$\min \|x\|$ st $L^T x = c$ Apply CRAIG or LSQR

Null-space basis (full rank)

$A =$  $= LU, \qquad [L, U, P, Q] = \texttt{lu}(A) \qquad L$ well-conditioned

$$L^{-1}A = U \quad \Rightarrow \quad PL^{-1}AQ \;=\; PUQ \;=\; \begin{pmatrix} U_1 \\ 0 \end{pmatrix}$$

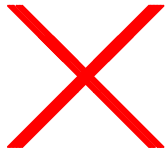
$$\Rightarrow (PL^{-1})A \;=\; \begin{pmatrix} U_1 \\ 0 \end{pmatrix} Q^T$$

Hence:

$$A^T Z \;=\; 0, \qquad Z = L^{-T} P^T \begin{pmatrix} 0 \\ I \end{pmatrix}, \qquad \mathrm{cond}(Z) \leq \mathrm{cond}(L)$$

Caution!

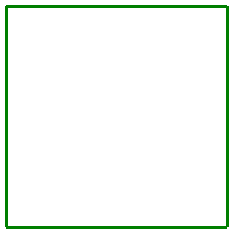
Is `L` well-conditioned?

`[L,U,p,q] = lusol(A,2.0)`	LUSOL	
`[L,U,P] = lu(A,0.5)`	GP	
`[L,U,P,Q] = lu(A,0.5)`	UMFPACK	

UMFPACK's "`LDU`" spreads `D` into both `L` and `U`

(Easily fixed later)

Null-space basis (low rank)

$A =$  $= LDU$ $\qquad L, U$ well-conditioned

Threshold Rook Pivoting

$$AU^{-1} = LD, \qquad \text{some } D_{jj} = 0$$

Hence:

$$AZ = 0, \qquad Z = U^{-1}E, \qquad \text{cond}(Z) \leq \text{cond}(U)$$

lp_solve

An open-source Linear and Mixed Integer Programming solver

http://groups.yahoo.com/group/lp_solve/

http://groups.yahoo.com/group/lp_solve/files/LUSOL/

- Implemented in C, runs on most platforms
- Repository for a C implementation of LUSOL created by Kjell Eikland (F77 $\longrightarrow$ Pascal $\longrightarrow$ C)
  Factor includes dynamic reallocation of storage
- Choice of basis factorization packages (BFPs)
  LUSOL is now the default

Yin Zhang (Rice U)

Bug report to lp_solve@yahoogroups.com, 29 Nov 2005

I'm using LUSOL for LU decomposition. I need a rank-revealing LU decomposition method that can handle matrices of size 1,000,000 by 50,000 (with ≈5M nonzeros). The lu() in matlab/umfpack is not rank-revealing. So I wrote a mex wrapper for LUSOL. But then I found the above bug.

Saunders: On such large systems, you should be using TRP – it's much more efficient than TCP and essentially as reliable for rank-detection (as long as the factor tolerance is pretty close to 1).

Yin Zhang: For some reason, even TCP seems to be pretty fast on my matrices. TRP with 1.1 factor also works very well. This may be due to the structure of my matrices (am working on large-scale network inference, in particular, inferring link delay from end-to-end path delay measurements, so my matrices are "routing matrices", which are highly sparse and most nonzero entries are 1.)

SUMMARY

LUSOL features

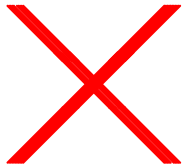
Square or rectangular A

Normal sparse LU	Threshold Partial Pivoting
Rank-revealing LU	for "basis repair" Threshold Rook Pivoting Threshold Complete Pivoting
Stable updates	add, replace, delete, rank-one Bartels-Golub-Reid style

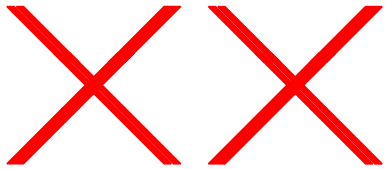
Open source

Terminology

LU with row interchanges

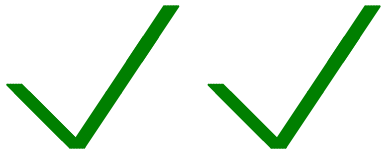


LU with column pivoting



LU with L well-conditioned

LDU with L well-conditioned



LDU with L and U well-conditioned

Anyone for TENNIS?!