LUSOL: A basis package for constrained optimization

IFORS Triennial Conference on OR/MS
Honolulu, HI, July 11–15, 2005

Michael O'Sullivan and Michael Saunders

Dept of Engineering Science
University of Auckland
Auckland, New Zealand
michael.osullivan@auckland.ac.nz

Dept of Management Sci & Eng
Stanford University
Stanford, CA 94305-4026
saunders@stanford.edu

Abstract

LUSOL is currently the BFP (basis factorization package) for several optimization packages, including MINOS, SQOPT, SNOPT, ZIP, PATH, and lp_solve. Threshold Rook Pivoting is an important feature for basis repair (recovery from unexpected singularity). We review the open source Fortran and C implementations of LUSOL.

LUSOL

Maintains 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 Cmex	Michael O'Sullivan (1999–present)
C (for lp_solve)	Kjell Eikland (2004–present)

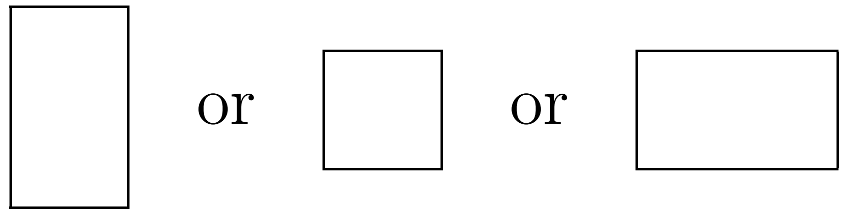
Features

Square or rectangular A

Rank-revealing LU	for "basis repair"
Stable updates	Bartels-Golub style

LUSOL

A =



= LU

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

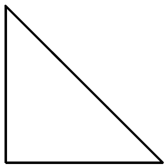
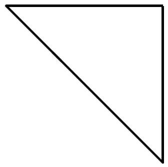
SOLVE $Lx = y,\ L^Tx = y,\ Ux = y,\ U^Tx = y,\ Ax = y,\ A^Tx = y$

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

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

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 **stability** and **sparsity**
- Markowitz strategy for suggesting sparse pivots
- Stability options:

  TPP Threshold Partial Pivoting
  TRP Threshold Rook Pivoting
  TCP Threshold Complete Pivoting

Partial Pivoting

$$
\begin{matrix}
\cdot & \cdots & \cdots & \cdots & \cdots & \cdots & \cdot \\
 & \ddots & & & & & \vdots \\
 & & \ddots & & & & \vdots \\
 & & & \color{red}{4.0} & \times & \times & \times \\
 & & & \color{blue}{2.0} & \times & \times & \times \\
 & & & \color{blue}{1.0} & \times & \times & \times \\
 & & & \color{blue}{4.0} & \times & \times & \times \\
 & & & \color{blue}{0.1} & \times & \times & \times
\end{matrix}
$$

Rook Pivoting

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

Complete Pivoting

$$\begin{matrix} \cdot & \cdots & \cdots & \cdots & \cdots & \cdots & \cdot \\ & \ddots & & & & & \vdots \\ & & \ddots & & & & \vdots \\ & & & {\color{red}9.0} & 1.0 & 0.1 & 6.0 \\ & & & 2.0 & \times & \times & \times \\ & & & 1.0 & \times & \times & \times \\ & & & 4.0 & \times & \times & 9.0 \\ & & & 0.1 & \times & 0.1 & \times \end{matrix}$$

TPP: Threshold Partial Pivoting

$$
\begin{array}{ccccccc}
\cdot & \cdots & \cdots & \cdots & \cdots & \cdots & \cdot \\
 & \ddots & & & & & \vdots \\
 & & \ddots & & & & \vdots \\
 & & & \color{red}{2.0} & \times & & \times \\
 & & & \color{blue}{2.0} & & & \times \\
 & & & & \times & \times & \\
 & & & \color{blue}{4.0} & \times & \times & \times \\
 & & & & \times & & \times
\end{array}
$$

TRP: Threshold Rook Pivoting

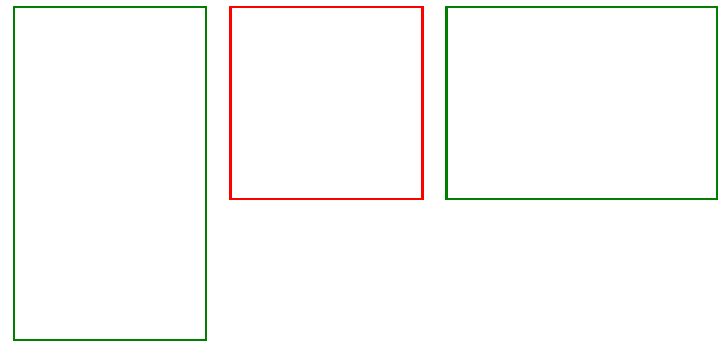
$$
\begin{array}{ccccccc}
\cdot & \cdots & \cdots & \cdots & \cdots & \cdots & \cdot \\
 & \ddots & & & & & \vdots \\
 & & \ddots & & & & \vdots \\
 & & & 4.0 & 1.0 & & 6.0 \\
 & & & 2.0 & & & \times \\
 & & & & \times & \times & \\
 & & & 4.0 & \times & \times & \times \\
 & & & & \times & & \times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

TCP: Threshold Complete Pivoting

$$
\begin{array}{ccccccc}
\cdot & \cdots & \cdots & \cdots & \cdots & \cdots & \cdot \\
 & \ddots & & & & & \vdots \\
 & & \ddots & & & & \vdots \\
 & & & \color{red}{5.0} & \color{blue}{1.0} & & \color{blue}{6.0} \\
 & & & \color{blue}{2.0} & & & \color{blue}{\times} \\
 & & & & \color{blue}{\times} & \color{blue}{\times} & \\
 & & & \color{blue}{4.0} & \color{blue}{\times} & \color{blue}{\times} & \color{blue}{9.0} \\
 & & & & \color{blue}{\times} & & \color{blue}{\times}
\end{array}
$$

Rank-Revealing Factors

$$A = XDY^T =$$



X, Y well conditioned, D diagonal

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

- SVD UDV^T
- QR with column interchanges QDR
- LU with Rook Pivoting LDU
- LU with Complete Pivoting LDU

$$
\begin{aligned}
L &= \begin{pmatrix} 1 & \\ 100 & 1 \end{pmatrix} \\[1em]
\operatorname{cond}(L) &= 100?
\end{aligned}
$$

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

$$\operatorname{cond}(L) = 10000$$

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

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

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

TRP or TCP would give

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

Implementing TRP

At each stage of Gaussian elimination:

$$A \leftarrow A - lu^T$$

$$\alpha_j = \text{biggest element in col } j$$
$$\beta_i = \text{biggest element in row } i$$

Implementing TCP

At each stage of Gaussian elimination:

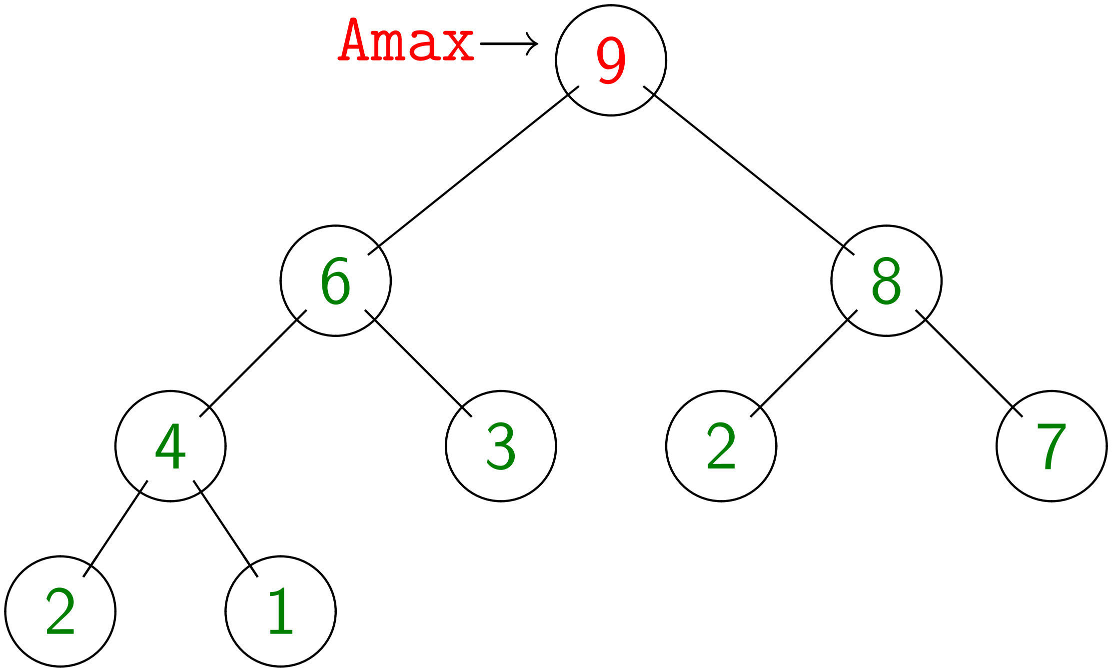
$$A \quad \leftarrow \quad A - lu^T$$

$$\alpha_j \quad = \quad \text{biggest element in col } j$$

$$\texttt{Amax} \quad = \quad \text{biggest element in } A \qquad (= \max\ \alpha_j = \max\ \beta_i)$$

TCP: Store α_j in a Heap

Thanks to John Gilbert



Ha(k)	9.0	6.0	8.0	...	α_j
Hj(k)	2	7	1	...	j
Hk(j)	3	1	6	...	k (location of j in heap)

RESULTS

Problem memplus from Harwell-Boeing collection

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

		τ	nnz(L+U)	Time
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
TCP		100.0	140000	2
		10.0	579000	30
	RR	3.99	2460000	475
	RR	2.5	2890000	6610
	RR	1.5	7080000	27875
PP (SuperLU, colamd)		1.0	4470000	≈ 250

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%

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% (!)

SOLVE

SOLVE

Dense rhs

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

Sparse rhs (future)

- Gilbert and Peierls (1988), CPLEX 7.1 (2001)
- $Lx = y$ 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)
- Product-form update would be ok: $B_k = L_0 U_0 E_1 E_2 \ldots E_k$
- Points to Schur-complement updates

APPLICATIONS

LUSOL in MINOS and SQOPT

BR factorization   rank detection for square B

$$B = \square = LU, \quad PLP^T = \begin{pmatrix} L_1 & \\ L_2 & L_3 \end{pmatrix}, \quad PUQ = \begin{pmatrix} U_1 & U_2 \\ & \ddots \end{pmatrix}$$

TRP or TCP, $\tau \le 2.5$, discard factors

BS factorization   basis detection for rectangular $W = (B \;\; S)$

$$W^T = \square = LU, \quad PLP^T = \begin{pmatrix} L_1 & \\ L_2 & I \end{pmatrix}, \quad PUQ = \begin{pmatrix} U_1 \\ 0 \end{pmatrix}$$

TPP, TRP, or TCP $\tau \le 2.5$, discard factors

New B = first m columns of WP^T

Deficient-basis simplex methods

Ping-Qi Pan:

- A revised dual projective pivot algorithm for linear programming, SIOPT, to appear 2005
- A revised primal deficient-basis simplex algorithm for linear programming, SIOPT, submitted June 2005

Take advantage of degeneracy:

$$A = \begin{bmatrix} B & N \end{bmatrix}, \qquad Bx_B = b$$

Thm: $\mathrm{cond}(B) \le \mathrm{cond}(B \;\; a)$ Apply LUSOL to rectangular B

lp_solve

An open source Mixed Integer Programming solver
http://groups.yahoo.com/group/lp_solve/

- GNU LGPL, implemented in C, runs on most platforms

- Repository for a C implementation of LUSOL created by Kjell Eikland (F77 $\rightarrow$ Pascal $\rightarrow$ C)
  Main LU includes dynamic reallocation of storage
  http://groups.yahoo.com/group/lp_solve/files/LUSOL/

- Choice of BFPs
  LUSOL is now the default

SUMMARY

LUSOL

Features

Square or rectangular A

Rank-revealing LU	for "basis repair"
Stable updates	Bartels-Golub style

Future Tasks

FACTOR	Improve β_i = max element in each row
	Special handling of dense columns
SOLVE	Sparse rhs's
UPDATE	Schur-complement
	(F90, Hanh Huynh's thesis)
Language	F77 $\rightarrow$ C always possible via f2c
	F77 $\rightarrow$ Pascal $\rightarrow$ C done for lp_solve
	F90 $\rightarrow$ C? (NAG F95 compiler?)

COIN-OR project

Future Tasks

FACTOR	Improve β_i = max element in each row
	Special handling of dense columns
SOLVE	Sparse rhs's
UPDATE	Schur-complement
	(F90, Hanh Huynh's thesis)
Language	F77 $\rightarrow$ C always possible via f2c
	F77 $\rightarrow$ Pascal $\rightarrow$ C done for lp_solve
	F90 $\rightarrow$ C? (NAG F95 compiler?)

COIN-OR project

Documentation