A review of sparsity vs stability in LU updates

Michael Saunders

Dept of Management Science and Engineering (MS&E)
Systems Optimization Laboratory (SOL)
Institute for Computational Mathematics and Engineering (ICME)

Stanford University, California

Session: 40 Years of Forrest and Tomlin

INFORMS 2012 Annual Meeting
Phoenix AZ, Oct 14–18, 2012

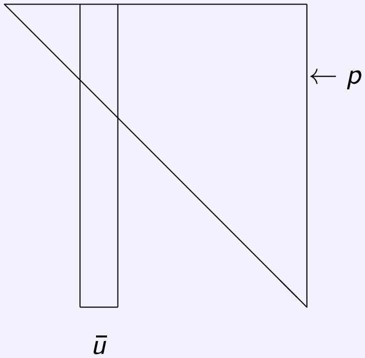
Abstract

The Forrest-Tomlin update has stood the test of time within many generations of commercial mathematical programming systems. Its ease of implementation leads to high efficiency and evidently acceptable reliability. We review its relation to Reid's version of the Bartels-Golub update as implemented in LA05, LA15, and LUSOL. In particular, we examine the extent to which FT implementations must "live dangerously" in order to achieve the desired efficiency.

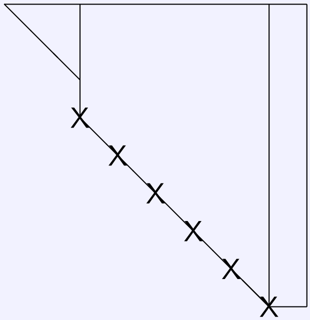
Outline

The Bartels-Golub update

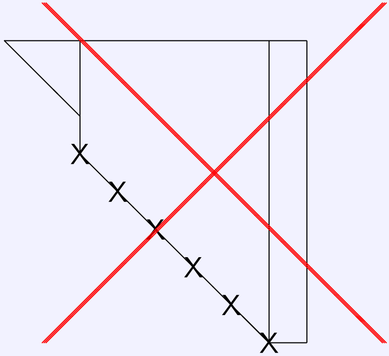
Dense Bartels-Golub



Dense Bartels-Golub



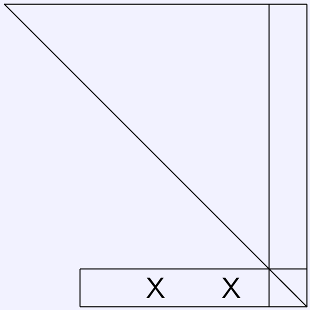
Dense Bartels-Golub



Forget the Hessenberg

The Forrest-Tomlin update

Forrest-Tomlin (conceptually)

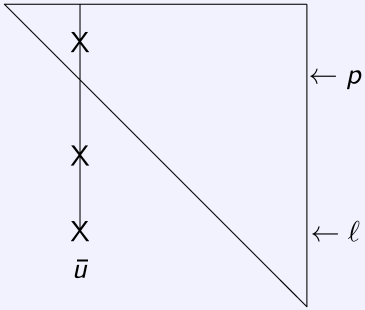


Few nonzeros to eliminate

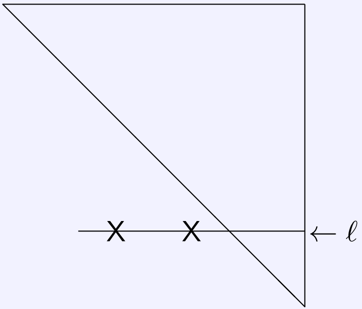
U stored by columns

Sparse Bartels-Golub

Sparse Bartels-Golub (Reid, Saunders)



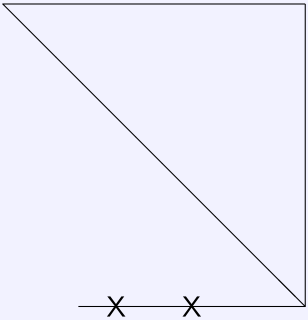
Sparse Bartels-Golub (Reid, Saunders)



LA05, LA15, LUSOL

U stored by rows

The connection



FT: Eliminate X's with diags as pivots

BG: Allow row interchanges

Use a product of $\begin{pmatrix} 1 & \\ \mu & 1 \end{pmatrix}$ and permutations

How big can μ safely be??

A Tale of Two Tolerances

LU factor tol α
LU update tol β

Control and/or monitor stability

LU factor tol α

Threshold partial pivoting controls cond(L)

$$B = LU \qquad L_{ii} = 1 \qquad |L_{ij}| \le \alpha$$

$\alpha = 10.0$ or $100.0 \Rightarrow$ usually sparse and stable

LU factor tol α

Threshold partial pivoting controls cond(L)

$$B = LU \qquad L_{ii} = 1 \qquad |L_{ij}| \le \alpha$$

$\alpha = 10.0$ or $100.0 \Rightarrow$ usually sparse and stable

Threshold rook pivoting controls cond(L), cond(U)

$$B = LDU \qquad L_{ii} = 1 \qquad |L_{ij}| \le \alpha$$

$$\textit{and}\ U_{ii} = 1 \qquad |U_{ij}| \le \alpha$$

$1 < \alpha \le 2.0 \Rightarrow$ rank-revealing (for basis repair)

Factor and Update involve triangular matrices

$$M = \begin{pmatrix} 1 & \\ 100 & 1 \end{pmatrix} \qquad \text{cond}(M) = ??$$

Factor and Update involve triangular matrices

$$M = \begin{pmatrix} 1 & \\ 100 & 1 \end{pmatrix} \qquad \text{cond}(M) \approx 10^4$$

Factor and Update involve triangular matrices

$$M = \begin{pmatrix} 1 & \\ \mu & 1 \end{pmatrix} \qquad \text{cond}(M) \approx \mu^2$$

$$|\mu| > 1$$

FT update involves triangular matrices

$$R = \begin{pmatrix} 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & r_{p+1} & \dots & r_m & 1 \end{pmatrix} \qquad \text{cond}(R) \approx r_{\text{max}}^2$$

Product-form update involves triangular matrices

$$T = \begin{pmatrix} 1 & & v_1 & & \\ & 1 & \vdots & & \\ & & v_p & & \\ & & \vdots & \ddots & \\ & & v_m & & I \end{pmatrix} \qquad \text{cond}(T) \approx v_{\max}^2/|v_p|$$

LU update tol β

$$\bar{B} = B(I + (v - e_p)e_p^T)$$

Use β to control or monitor basis updates

Product-form	$\bar{B} = BT$	monitor cond(T)
Bartels-Golub	$\bar{L} = LM_1M_2\ldots$	control cond(M_j)
Forrest-Tomlin	$\bar{L} = LR$	monitor cond(R)

LU update tol β

Product-form update

$$\bar{B} = BT \qquad T = I + (v - e_p)e_p^T \quad \text{monitor cond}(T)$$

Update if $v_{\max}^2/|v_p| \le \beta$ else refactor

How big should β be?? 10^6?

LU update tol β

Product-form update

$\bar{B} = BT \qquad T = I + (v - e_p)e_p^T \qquad$ monitor cond(T)

Update if $v_{\max}^2/|v_p| \le \beta \qquad$ else refactor

How big should β be?? 10^6?

LU updates

BG	$\bar{L} = LM_1M_2\ldots$	control $\lvert M_{ij}\rvert \le \beta$
FT	$\bar{L} = LR$	monitor $\lvert R_{ij}\rvert \le \beta$?

For BG we can enforce $\beta = 10$ say

For FT how big should β be?? 10^6?

A numerical experiment

Simulate FT via BG

- Run MINOS on a numerically challenging LP (e.g. pilot87)
- Set LUSOL's LU update tol $\beta = 10$, 10^2, 10^3, 10^4, 10^5, 10^6, ...
- Count number of times something broke

Simulate FT via BG

- Run MINOS on a numerically challenging LP (e.g. pilot87)
- Set LUSOL's LU update tol $\beta = 10, 10^2, 10^3, 10^4, 10^5, 10^6, \ldots$
- Count number of times something broke

Normally

Factorization freq $= 100$
LU factor tol α $= 100.0$
LU update tol β $= 10.0$

Simulate FT via BG

- Run MINOS on a numerically challenging LP (e.g. pilot87)
- Set LUSOL's LU update tol $\beta = 10,\ 10^2,\ 10^3,\ 10^4,\ 10^5,\ 10^6,\ \ldots$
- Count number of times something broke

Normally

Factorization freq $= 100$
LU factor tol α $= 100.0$
LU update tol β $= 10.0$

- How would MINOS measure broke?

Simulate FT via BG

- Run MINOS on a numerically challenging LP (e.g. pilot87)
- Set LUSOL's LU update tol $\beta = 10, 10^2, 10^3, 10^4, 10^5, 10^6, \dots$
- Count number of times something broke

Normally

Factorization freq $= 100$
LU factor tol α $= 100.0$
LU update tol β $= 10.0$

- How would MINOS measure broke?
- Row check failed? Every 60 itns, see if $\|Ax - b\|$ is too big
  Never happened

Simulate FT via BG

- Run MINOS on a numerically challenging LP (e.g. pilot87)
- Set LUSOL's LU update tol $\beta = 10, 10^2, 10^3, 10^4, 10^5, 10^6, \ldots$
- Count number of times something broke

Normally

Factorization freq $= 100$
LU factor tol α $= 100.0$
LU update tol β $= 10.0$

- How would MINOS measure broke?
- Row check failed? Every 60 itns, see if $\|Ax - b\|$ is too big
  Never happened
- LU became singular? Every update, check if U has small diag
  Increasingly often

Tests for accepting LU update

$$\bar{L} = L \begin{pmatrix} I & \\ r^T & 1 \end{pmatrix} \qquad \bar{U} = \begin{pmatrix} U_1 & u \\ & \delta \end{pmatrix}$$

Tomlin 1975

$$\frac{r_{\max} u_{\max}}{|\delta|} \le 16^7 \qquad (3 \cdot 10^8)$$

Our experiment

$$r_{\max} \le \beta \quad \text{and} \quad \frac{\|u\|_1}{|\delta|} \le \epsilon^{-2/3} \qquad (3 \cdot 10^{10})$$

Tests for accepting LU update

$$\bar{L} = L \begin{pmatrix} I & \\ r^T & 1 \end{pmatrix} \qquad \bar{U} = \begin{pmatrix} U_1 & u \\ & \delta \end{pmatrix}$$

Tomlin 1975

$$\frac{r_{\max} u_{\max}}{|\delta|} \le 3 \cdot 10^8$$

Our experiment

$$r_{\max} \le \beta \quad \text{and} \quad \frac{\|u\|_1}{|\delta|} \le \epsilon^{-2/3} \qquad (3 \cdot 10^{10})$$

Roughly equivalent to

$$\frac{r_{\max} u_{\max}}{|\delta|} \le \beta 10^{10} \qquad = 10^{11}, 10^{12}, \ldots$$

Numerical results

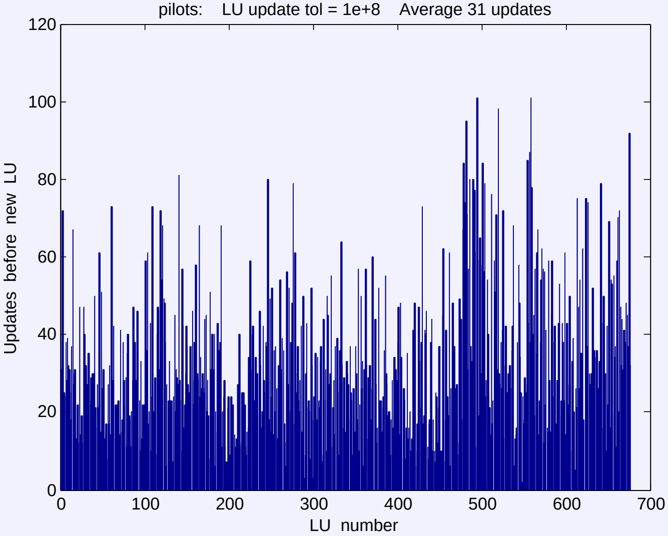
MINOS simulation of FT updates

LPnetlib problem `pilot` $\epsilon = 10^{-16}$ Featol = Opttol = 10^{-6}

LU update tol $\beta = 10,\ 10^2,\ 10^3,\ \ldots$

beta	itns	time	LUfacs	singular warnings	average updates
1e+1	16865	5.06	167	0	100
1e+2	16865	5.02	170	19	99
1e+3	16865	5.24	220	176	77
1e+4	16865	5.73	305	287	55
1e+5	18837	7.05	445	442	42
1e+6	20079	8.28	576	573	35
1e+7	16704	7.21	514	510	32
1e+8	21115	9.31	678	674	31

MINOS simulation of FT updates



MINOS simulation of FT updates

LPnetlib problem `pilot87` $\epsilon = 10^{-16}$ Featol = Opttol $= 10^{-6}$

LU update tol $\beta = 10,\ 10^2,\ 10^3,\ \ldots$

beta	itns	time	LUfacs	singular warnings	average updates
1e+1	20055	14.64	197	0	101
1e+2	19159	13.99	194	30	99
1e+3	20001	15.34	259	202	77
1e+4	20001	17.66	380	360	53
1e+5	21635	22.19	526	511	41
1e+6	21837	24.41	624	617	35
1e+7	23158	27.72	772	770	30
1e+8	36862	48.91	1385	1379	27

MINOS simulation of FT updates

LPnetlib problem mod2pre $\epsilon = 10^{-16}$ Featol = Opttol = 10^{-6}

LU update tol $\beta = 10,\ 10^2,\ 10^3,\ \ldots$

beta	itns	time	LUfacs	singular warnings	
1e+1	500000	705.0	4844	1	
1e+2	500000	704.9	4859	73	
1e+3	500000	698.5	5312	1860	
1e+4	11057*				
1e+5	17256*				
1e+6	500000	793.4	8955	7941	859 infeas
1e+7	99774*				
1e+8	500000	785.6	7528	5661	3000 infeas

*Refactorization gave singular LU
Rook pivoting invoked, β reduced to 10.0

Quad SQOPT simulation of FT updates

LPnetlib problem pilot $\epsilon = 10^{-34}$ Featol = Opttol = 10^{-15}

LU update tol $\beta = 10^4,\ 10^6,\ 10^8,\ \ldots$

beta	itns	time	LUfacs	singular warnings	average updates	dual infeas
1e+4	17251	216.0	173	2	100	E-25
1e+6	17251	215.4	187	70	92	E-27
1e+8	17251	215.1	213	144	81	E-21
1e+10	17251	216.0	252	205	68	E-24
1e+12	17251	220.3	264	219	65	E-27
1e+14	17251	219.8	264	206	65	E-29
1e+16	17251	217.6	229	150	75	E-12
1e+18	17251	216.9	212	111	81	E-19
1e+20	11013*					
1e+22	7864*					
1e+24	4098*					

*Refac singular, beta reduced to 50.0

Conclusion

Conclusion

Test for accepting an FT update

WHIZARD	$\frac{r_{\max} u_{\max}}{\lvert\delta\rvert} \le 10^8$
CLP, CPLEX, Gurobi, Xpress, Mozek, . . .	$\frac{r_{\max} u_{\max}}{\lvert\delta\rvert} \le$? or ?
Probably safer	$r_{\max} \le \beta$ and $\frac{u_{\max}}{\lvert\delta\rvert} \le 10^{10}$

Average updates per LU

$\beta = 10^4$	50
$\beta = 10^5$	40
$\beta = 10^6$	35
$\beta = 10^7$	30

References (in chronological order)

- R. H. Bartels (1971). A stabilization of the simplex method, *Numerische Mathematik* 16, 414–434.
  Eliminated subdiag of dense Hessenberg $H = \bar{U}P$

References (in chronological order)

- R. H. Bartels (1971). A stabilization of the simplex method, *Numerische Mathematik* 16, 414–434.

  Eliminated subdiag of dense Hessenberg $H = \bar{U}P$

- J. J. H. Forrest and J. A. Tomlin (1972). Updating triangular factors of the basis to maintain sparsity in the product form simplex method, *Mathematical Programming* 2, 263–278.

  Eliminated pth row of sparse H (not subdiag)

References (in chronological order)

- R. H. Bartels (1971). A stabilization of the simplex method, *Numerische Mathematik* 16, 414–434.

  Eliminated subdiag of dense Hessenberg $H = \bar{U}P$

- J. J. H. Forrest and J. A. Tomlin (1972). Updating triangular factors of the basis to maintain sparsity in the product form simplex method, *Mathematical Programming* 2, 263–278.

  Eliminated pth row of sparse H (not subdiag)

- J. A. Tomlin (1975). An accuracy test for updating triangular factors, *Math. Prog. Study* 4, 142–145.

  Test for an unstable update

References (in chronological order)

- R. H. Bartels (1971). A stabilization of the simplex method, *Numerische Mathematik* 16, 414–434.

  Eliminated subdiag of dense Hessenberg $H = \bar{U}P$

- J. J. H. Forrest and J. A. Tomlin (1972). Updating triangular factors of the basis to maintain sparsity in the product form simplex method, *Mathematical Programming* 2, 263–278.

  Eliminated pth row of sparse H (not subdiag)

- J. A. Tomlin (1975). An accuracy test for updating triangular factors, *Math. Prog. Study* 4, 142–145.

  Test for an unstable update

- M. A. Saunders (1976). The complexity of LU updating, in Anderssen and Brent (eds.), *The Complexity of Computational Problem Solving*, 214–230.

  Picture of symmetrically permuted $P^T\bar{U}P$ with row spike to be eliminated

References (contd)

- J. K. Reid (1982). A sparsity-exploiting variant of the Bartels-Golub decomposition for linear programming bases, *Mathematical Programming* 24, 55–69.

  U stored by rows to facilitate BG update

References (contd)

- J. K. Reid (1982). A sparsity-exploiting variant of the Bartels-Golub decomposition for linear programming bases, *Mathematical Programming* 24, 55–69.

  U stored by rows to facilitate BG update

- P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright (1987). Maintaining LU factors of a general sparse matrix, *Linear Algebra and its Applics.* 88/89, 239–270.

  Rectangular LU, more general updates, cond(L) controlled throughout

References (contd)

- J. K. Reid (1982). A sparsity-exploiting variant of the Bartels-Golub decomposition for linear programming bases, *Mathematical Programming* 24, 55–69.

  U stored by rows to facilitate BG update

- P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright (1987). Maintaining LU factors of a general sparse matrix, *Linear Algebra and its Applics.* 88/89, 239–270.

  Rectangular LU, more general updates, cond(L) controlled throughout

- P. E. Gill, W. Murray, and M. A. Saunders (2005). SNOPT: An SQP algorithm for large-scale constrained optimization, *SIAM Review* 47:1, 99–131.

  LU with Threshold Rook Pivoting, same general updates

Forrest-Tomlin