2005 International Symposium on Nonlinear
Theory and its Applications (NOLTA2005)
Bruges, Belgium, October 18-21, 2005

Additive-form Iterative Refinement of LU Factorization of an Ill-Conditioned Matrix

Kunio Tanabe†

†School of Science and Engineering, Waseda University
3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555 Japan
Email:tanabe.kunio@waseda.jp

Abstract—Rump[1] showed that an iterated preconditioning of an ill-conditioned coefficient matrix could produce a more accurate solution of a linear equation. His method can be interpreted as a *product-form* iterative refinement of *inverse* of a matrix. In this paper we introduce an *additive-form* iterative refinement of *LU factorization* of an ill-conditioned matrix, in which the triangular factor matrices are approximated by sums of matrices with row precision entries.

1. Introduction

It has been a common practice in numerical solution of a system of linear equations to factorize the coefficient matrix once and then proceed directly to a solution process, as is the case with Gaussian elimination method in which the backward substitution follows directly the forward elimination. When encountered with an ill-conditioned matrix, we have only to accept a resulting numerical solution accompanied by such a warning statement as 'Matrix is close to singular or badly scaled. Results may be inaccurate.'

Rump[1] advocated, however, that even a decomposition of an ill-conditioned matrix could be utilized for pre-conditioning to obtain a more accurate solution by decomposing again the precondtioned coefficient matrix. Employing the existing algorithms for high precision accumulated-inner-product[3, 5], Ohta, Ogita, Rump and Oishi[6] found that a repeated application of Rump's algorithm could bring about an eventual inversion of an arbitrarily ill-conditioned matrix, enabling us to obtain a verified a posteriori estimate of component-wise errors of a numerical solution of the associated linear equation.

With the same purpose as the cited papers, we introduce an additive-form iterative refinement of LU factorization of an ill-conditioned matrix, in which the upper- and lower-triangular factor matrices are approximated by the sums of the same type of matrices with low precision entries.

In Section 2 and 3, we give a prototype of the iterative refinement algorithm and its mathematical analysis. In Section 4 and 5, we give a modified version of it and its convergence analysis. In Section 6, we describe a practical implementation of one of the modified version.

For the similar iterative refinements of Cholesky- and QR-factorizations, see Tanabe[7, 8].

2. Iterative Refinement of LU Factorization

We assume that A is an n by n matrix which allows the LU factorization $A = L^*U^*$.

We introduce a prototype of the iterative refinement of LU factorization.

Additive-form Iterative Refinement of LU Factorization (AIR-LUF): Starting with a pair (L_0, U_0) of lower-triangular matrix with unit diagonal elements and an invertible upper-triangular matrix, generate a sequence of pairs $\{(L_k, U_k)\}_{k=1,2,\cdots}$ by the following iteration:

Step 1: Find a pair $(\delta L_k, \delta U_k)$ of a lower-triangular matrix with zero diagonal elements and an upper-triangular matrix which satisfy the matrix linear equation,

$$L_k \delta U_k + \delta L_k U_k = R_k, \tag{1}$$

where

$$R_k \equiv A - L_k U_k \tag{2}$$

Step 2: Form

$$\begin{aligned} L_{k+1} &= L_k + \delta L_k & (3)\\ U_{k+1} &= U_k + \delta U_k. & (4) \end{aligned}$$

The linear equation (1) has a special structure,

$$L\delta U + \delta L U = R, \tag{5}$$

where we drop the subscript k. If we put

$$L \equiv \begin{bmatrix} 1 & 0 \\ * & L_- \end{bmatrix}, \quad U \equiv \begin{bmatrix} u_1^1 & * \\ 0 & U_- \end{bmatrix} \tag{6}$$

$$\delta L \equiv \begin{bmatrix} 0 & 0 \\ * & \delta L_- \end{bmatrix}, \quad \delta U \equiv \begin{bmatrix} \delta u_1^1 & * \\ 0 & \delta U_- \end{bmatrix} \tag{7}$$

then, the equation (5) reduces to the system of equations

$$L_- \delta U_- + \delta L_- U_- = R_- \tag{8}$$

and

$$R - l_1 \delta u^1 - \delta l_1 u^1 = \begin{bmatrix} 0 & 0 \\ 0 & R_- \end{bmatrix}, \tag{9}$$

where the j-th column- and the i-th row-vectors of a matrix X are denoted respectively by x_j and x^i, the (i, j) element of X is denoted by x^i_j and the subscript symbol '-' indicates that the size of the attached matrix is reduced by one.

Since Eq.(9) is easily solved with respect to δu^1 and δl_1, we can solve Eq.(5) by the following recursive algorithm.

Recursive Algorithm : Compute the row vectors of δU and column vectors of δL alternately by the recursion:
Step 1: Put the first row vector δu^1 of δU by

$$\delta u^1 = r^1 \tag{10}$$

where r^1 is the first row vector of R.
Step 2: Compute the first column vector δl_1 of δL by

$$\begin{aligned} \delta l_1 &= \frac{1}{u^1_1}(r_1 - \delta u^1_1 l_1) && (11)\\ &= \frac{1}{u^1_1}(r_1 - r^1_1 l_1), && (12) \end{aligned}$$

where u^1_1 and r^1_1 is the (i,j) elements of U and R respectively. The first element of δl_1 is zero and needs not be actually computed.
Step 3: Compute the matrix R_- by

$$\begin{aligned} \begin{bmatrix} 0 & 0 \\ 0 & R_- \end{bmatrix} &= R - l_1\delta u^1 - \delta l_1 u^1 && (13)\\ &= R - l_1 r^1 - \delta l_1 u^1 && (14) \end{aligned}$$

where R_- is computed by the formula (14) and the first column and the first row vectors of the right hand side are zero vectors, hence need not be actually computed.

Taking advantage of the structural similarity between Eq.(8) and Eq.(5), we can solve Eq.(1) by recursively applying the same steps to the reduced matrices , L_-, U_- and R_-. In fact, the vectors $\delta u^1, \delta l_1, \delta u^2, \delta l_2, \cdots, \delta u^n, \delta l_n \equiv 0$ are generated in this order by the recursion.

Recursive Algorithm requires about $4n^3/3$ arithmetic operations for solving Eq.(5) with dense coefficient matrices L and U.

The solution of Eq.(1) is obtained by putting $R \equiv R_k, L \equiv L_k, U \equiv U_k, \delta L \equiv \delta L_k$ and $\delta U \equiv \delta U_k$ in this algorithm.

3. Convergence of AIR-LUF

Given a square matrix X, let $\Delta(X)$ denote the lower triangular matrix with zero diagonal elements, whose nonzero elements coincides with the corresponding elements of X and let $\nabla(X)$ denote the upper triangular matrix whose nonzero elements coincide with the corresponding elements of X. Note that the equality, $\Delta(X)+\nabla(X) \equiv X$, holds and hence, $\Delta(I) = O$ and $\nabla(I) = I$ for identity matrix I, where O is zero matrix.

Lemma 1: The solution $(\delta L_k, \delta U_k)$ of the linear equation (1) is represented by the formulae

$$\delta L_k = L_k \Delta(L_k^{-1} R_k U_k^{-1}) \tag{15}$$

$$\delta U_k = \nabla(L_k^{-1} R_k U_k^{-1}) U_k. \tag{16}$$

We can compute the solution by Eqs.(15) and (16) instead of by Recursive Algorithm, but it requires about $7n^3/3$ arithmetic operations and is more susceptible to numerical errors.

Corollary 2: The pair (L_{k+1}, U_{k+1}) of the matrices generated in Step 2 of AIR-LUF can be alternatively given by the formulae

$$\begin{aligned} L_{k+1} &= L_k + L_k\Delta(L_k^{-1}R_kU_k^{-1}) && (17)\\ &= L_k(I + \Delta(L_k^{-1}R_kU_k^{-1})) && (18)\\ &= L_k + L_k\Delta(L_k^{-1}AU_k^{-1}) && (19)\\ &= L_k(I + \Delta(L_k^{-1}AU_k^{-1})) && (20) \end{aligned}$$

$$\begin{aligned} U_{k+1} &= U_k + \nabla(L_k^{-1}R_kU_k^{-1})U_k && (21)\\ &= (I + \nabla(L_k^{-1}R_kU_k^{-1}))U_k && (22)\\ &= \nabla(L_k^{-1}AU_k^{-1})U_k. && (23) \end{aligned}$$

We can compute L_k and U_k either by the product-form updating formulae (18) and (22) or by (20) and (23). If this is the case, the method is called **Product-form Iterative Refinement of LU Factorization (PIR-LUF)**. It can be implemented conveniently in an computational environment such as MATLAB, but is more susceptible to numerical error than AIR-LUF.

Let the matrix E_k be defined by

$$E_k \equiv L_k^{-1}R_kU_k^{-1} \equiv L_k^{-1}AU_k^{-1} - I, \tag{24}$$

then we have the following lemma.

Lemma 3: If the initial pair (L_0, U_0) satisfies the inequality,

$$\|E_0\|_\infty = \left\|L_0^{-1}R_0U_0^{-1}\right\|_\infty < 1, \tag{25}$$

the process of AIR-LUF is well-defined and the inequality,

$$\|E_{k+1}\|_\infty < \frac{\|E_k\|_\infty^2}{4(1 - \|E_k\|_\infty)} \tag{26}$$

holds for $k = 1, 2, \cdots, \infty$, where $\|X\|_\infty$ is the maximum norm of a matrix X.

Corollary 4: If the initial pair (L_0, U_0) satisfies the inequality, $\|E_0\|_\infty < 1 - \frac{1}{5-4\rho}$, for a small positive number ρ, then the inequality, $\|E_{k+1}\|_\infty < (1-\rho)\|E_k\|_\infty$ holds for $k = 1, 2, \cdots, \infty$. The monotonically decreasing sequence $\{\|E_k\|_\infty\}_{k=1,2,\cdots}$ converges to zero.

Theorem 5: Under the condition, $\|E_0\|_\infty < \frac{1}{2}$, the sequence of pairs $\{(L_k, U_k)\}_{k=1,2,\cdots}$ generated by AIR-LUF converges quadratically to (L^*, U^*) and satisfies the inequalities,

$$\|L_k - L^*\|_\infty \le 2^{k+2}\left(\frac{\|E_0\|_\infty}{2}\right)^{2^k}\|L_0\|_\infty \tag{27}$$

$$\|U_k - U^*\|_\infty \le 2^{k+2}\left(\frac{\|E_0\|_\infty}{2}\right)^{2^k}\|U_0\|_\infty . \tag{28}$$

4. Modified Iterative Refinement of LU Factorization

In the implementation of AIR-RUF, it is necessary to form the matrices L_k and U_k, which may require high precision storages throughout the iterations. We can, however, modifiy AIR-LUF so that the pair (L_k, U_k) of multiple-precision matrices are represented in terms of the sum of of (L_0, U_0) and $\{(\delta L_j, \delta U_j)\}_{j=1,2,\cdots k}$, each of which is saved in single(or double) precision storage and the coefficient matrices of Eq.(1) is approximated by matrices *with single-precision* in a controlled manner. Before introducing the detail of this implementation, which is described in Section 6, we give the modicication of AIR-LUF.

Modified Additive-form Iterative Refinement of LU Factorization (MAIR-LUF): Starting with a pair (L_0, U_0) of lower-triangular matrix with unit diagonal elements and an invertible upper-triangular matrix, generate a sequence $\{(L_k, U_k)\}_{k=1,2,\cdots}$ by the following iteration:

Step 1: Find a pair $(\delta L_k, \delta U_k)$ of a lower-triangular matrix with zero diagonal elements and an upper-triangular matrix which satisfy the matrix linear equation,

$$\tilde{L}_k \delta U_k + \delta L_k \tilde{U}_k = R_k, \qquad (29)$$

where

$$R_k \equiv A - L_k U_k \qquad (30)$$

and $\tilde{L}_k$ is a lower-triangular matrix with unit diagonal elements and $\tilde{U}_k$ is an invertible upper-triangular matrix which are chosen to approximate L_k and U_k respectively so that

$$\left\| \tilde{L}_k^{-1} L_k - I \right\|_\infty < \tau_k \qquad (31)$$

$$\left\| U_k \tilde{U}_k^{-1} - I \right\|_\infty < \tau_k, \qquad (32)$$

where $\{\tau_k\}$ is asequence of positive numbers such that $\tau_0 \equiv 0$ and $0 \le \tau_k < 1$. $\tilde{L}_k$ and $\tilde{U}_k$ will be called *'design matrices'*.

Step 2: Form

$$\begin{aligned} L_{k+1} &= L_k + \delta L_k & (33) \\ U_{k+1} &= U_k + \delta U_k. & (34) \end{aligned}$$

The solution of Eq.(29) is obtained by putting $R \equiv R_k, L \equiv \tilde{L}_k, U \equiv \tilde{U}_k, \delta L \equiv \delta L_k$ and $\delta U \equiv \delta U_k$ in Recursive Algorithm given in Section 2. Note that $L_0 \equiv \tilde{L}_0, U_0 \equiv \tilde{U}_0$ by the definition of τ_0.

5. Convergence of MAIR-LUF

Lemma 6: The solution $(\delta L_k, \delta U_k)$ of the linear equation (27) is represented by the formulae,

$$\delta L_k = \tilde{L}_k \Delta(\tilde{L}_k^{-1} R_k \tilde{U}_k^{-1}) \qquad (35)$$

$$\delta U_k = \nabla(\tilde{L}_k^{-1} R_k \tilde{U}_k^{-1}) \tilde{U}_k. \qquad (36)$$

Note that an analogous statement to Corollary 2 is not possible with MAIR-LUF. Hence, there could be no 'MPIR-LUF'.

Lemma 7: The sequence generated by MAIR-LUF satisfies the inequality,

$$\|R_{k+1}\|_\infty \le cond(\tilde{L}_k) cond(\tilde{U}_k) \left(\frac{\left\|\tilde{L}_k^{-1}\right\|_\infty \left\|\tilde{U}_k^{-1}\right\|_\infty \|R_k\|_\infty}{4} + \tau_k \right) \|R_k\|_\infty, \qquad (37)$$

where $cond(X) \equiv \|X\|_\infty \left\|X^{-1}\right\|_\infty$.

This lemma implies that if we choose the *design matrices* $\tilde{L}_k$ and $\tilde{U}_k$ so that their condition numbers and the norms of their inverses are uniformely bounded above by a moderate number Θ and $\|R_0\|_\infty$ is very small, then the sequence $\{R_k\}_{k=1,2,\cdots}$ converges to zero matrix. So it is expected that MAIR-LUF has a larger region of convergence than AIR-LUF.

Let the matrix $\tilde{E}_k$ be defined by

$$\tilde{E}_k \equiv \tilde{L}_k^{-1} R_k \tilde{U}_k^{-1} \equiv \tilde{L}_k^{-1}(A - L_k U_k)\tilde{U}_k^{-1}, \qquad (38)$$

then we have the following lemma.

Lemma 8: If the initial pair (L_0, U_0) satisfies the inequality,

$$\left\|\tilde{E}_0\right\|_\infty \equiv \left\|\tilde{L}_0^{-1} R_0 \tilde{U}_0^{-1}\right\|_\infty \equiv \left\|L_0^{-1} R_0 U_0^{-1}\right\|_\infty < 1 \qquad (39)$$

the process of MAIR-LUF is well-defined and the inequality,

$$\left\|\tilde{E}_{k+1}\right\|_\infty < \frac{\left\|\tilde{E}_k\right\|_\infty^2 + 4\tau_k \left\|\tilde{E}_k\right\|_\infty}{4(1-\tau_k)^2 (1 - \left\|\tilde{E}_k\right\|_\infty)} \qquad (40)$$

holds for $k = 1, 2, \cdots, \infty$.

Corollary 9: For a uniformely bounded sequence $\{\tau_k\}$ which satisfies $\tau_k < \hat{\tau}$, if the initial pair (L_0, U_0) satisfies the inequality,

$$\left\|\tilde{E}_0\right\|_\infty < 1 - \frac{1 + 4\hat{\tau}}{4(1-\rho)(1-\hat{\tau})^2 + 1}, \qquad (41)$$

for a small positive number $\rho < 1$, then the inequality, $\left\|\tilde{E}_{k+1}\right\|_\infty < (1-\rho)\left\|\tilde{E}_k\right\|_\infty$ holds for $k = 1, 2, \cdots, \infty$. The monotonically decreasing sequence $\{\left\|\tilde{E}_k\right\|_\infty\}_{k=1,2,\cdots}$ converges to zero.

Proposition 10: Under the conditions, $\tau_k < \hat{\tau} = 0.025$ and $\left\|\tilde{E}_0\right\|_\infty \equiv \left\|L_0^{-1} R_0 U_0^{-1}\right\|_\infty < \frac{1}{2}$, the sequence of pairs $\{(L_k, U_k)\}_{k=1,2,\cdots}$ generated by AIR-LUF converges to (L^*, U^*) and satisfies the inequalities,

$$\|L_k - L^*\|_\infty \le \frac{\delta^k}{1-\delta} \|L_0\|_\infty \left\|\tilde{E}_0\right\|_\infty, \qquad (42)$$

$$\|U_k - U^*\|_\infty \leq \frac{\delta^k}{1-\delta}\|U_0\|_\infty \left\|\tilde{E}_0\right\|_\infty , \qquad (43)$$

where $\delta = 0.9625\cdots$.

6. Implementation of MAIR-LUF

We describe storage allocation and arithmetic control for the actual implementation of MAIR-LUF, in which the matrices L_k and U_k are not formed explicitly. Instead, when needed, they are computed respectively as the sums of matrices $[L_0]$ and $\{[\delta L_j]\}_{j=1,2,\cdots k}$, and $[U_0]$ and $\{[\delta U_j]\ \}_{j=1,2,\cdots k}$, each of which are stored in single(or double) precision, where $[X]_m$ indicates the matrix X stored in m-tuple precision and $[X] \equiv [X]_1$.

Particular choices are made of *design matrices*, $\tilde{L}_k$ and $\tilde{U}_k$ in the following algorithm. We assume that the matrix A is stored in single precision and that the basic computation is executed with single-precision arithmetic. Multiple-precision computation is essential only for the computation of the residual R_k, but the result is stored in single precision matrix $[R_k]$. Matrix computation $*$ with m-tuple precision is denoted by $\langle * \rangle_m$ and assignment operation by $\Leftarrow$.

MAIR-LUF (p, m) **Algorithm**: Choose p and m such that $p \leq m$. Typically, $p = 1$ (or 2) and $m = 2$ (or 1)
Step 1: Compute the LU factorization of the matrix $[A]$ and put the resulting triangular matrices to be the initial matrices, $[L_0]$ and $[U_0]$.
Step 2: Compute the residual matrix with double precision arithmetic and save it in a single precision storage:

$$[R_0]_m \Leftarrow \langle\ [A] - [L_0][U_0]\ \rangle_m \qquad (44)$$

Step 3: By putting $R_0 = [R_0]_m$, $\tilde{L}_0 = [L_0]$ and $\tilde{U}_0 = [U_0]$, solve Eq.(29) with an extended-precision-accumulated-inner-product to obtain $[\delta L_0]$ and $[\delta U_0]$.
Step 4: Compute the residual matrix with triple precision arithmetic and save it in a single precision storage:

$$\begin{aligned} [R_1]_m \ \Leftarrow\ & \langle\ [A] - [L_0][U_0] \\ - \ & [\delta L_0][U_0] - [L_0][\delta U_0] \\ - \ & [\delta L_0][\delta U_0])\ \rangle_{m+1} \end{aligned} \qquad (45)$$

Step 5: Choose $\tilde{L}_1$ and $\tilde{U}_1$ and assign them:

$$[\tilde{L}_1]_p \Leftarrow \langle\ [L_0] + [\delta L_0]\ \rangle_m \qquad (46)$$

$$[\tilde{U}_1]_p \Leftarrow \langle\ [U_0] + [\delta U_0]\ \rangle_m \qquad (47)$$

Step 6: By putting $R_1 = [R_1]$, $\tilde{L}_1 = [\tilde{L}_1]_p$ and $\tilde{U}_0 = [\tilde{U}_0]_p$, solve Eq.(29) with an extended precision accumulated inner product to obtain $[\delta L_1]$ and $[\delta U_1]$.
Step 7: Compute the residual matrix with quadruple precision arithmetic and save it in a single precision storage:

$$\begin{aligned} [R_2]_m \ \Leftarrow\ & \langle\ [A] - [L_0][U_0] \\ - \ & [L_0][\delta U_0] - [\delta L_0][U_0] \\ - \ & [L_0][\delta U_1] - [\delta L_0][\delta U_0] - [\delta L_1][U_0] \\ - \ & [\delta L_0][\delta U_1] - [\delta L_1][\delta U_0] \\ - \ & [\delta L_1][\delta U_1])\ \rangle_{m+2} \end{aligned} \qquad (48)$$

Step 8: Choose $\tilde{L}_2$ and $\tilde{U}_2$ and assign them:

$$[\tilde{L}_2]_p \Leftarrow \langle\ [L_0] + [\delta L_0] + [\delta L_1]\ \rangle_m \qquad (49)$$

$$[\tilde{U}_2]_p \Leftarrow \langle\ [U_0] + [\delta U_0] + [\delta U_0]\ \rangle_m \qquad (50)$$

Step 9: By putting $R_2 = [R_2]$, $\tilde{L}_2 = [\tilde{L}_2]_p$ and $\tilde{U}_2 = [\tilde{U}_2]_p$, solve Eq.(29) with an extended precision accumulated inner product to obtain $[\delta L_2]$ and $[\delta U_2]$.
Step $\star$: Continue with the similar steps in which we compute the residuals R_k with progressive multiple precisions incremental with k.

Alternatively we could choose $[\tilde{L}_k] \equiv [\tilde{L}_1](or\ [L_0])$, and $[\tilde{U}_k] \equiv [\tilde{U}_1](or\ [U_0])$, $(k = (1), 2, 3, \cdots)$, without actually evaluating such Eqs. as (49), (50), etc..

The standard LU factorization yields matrices, $[L_0]$ and $[U_0]$ which admit the residual estimate, $|[A] - [L_0][U_0]| \leq \gamma_n\, |[L_0]|\, |[U_0]|$, where $\gamma_n \equiv nu/(1-nu)$ (Higham[3]). Hence, roughly speaking, $[\delta L_0]$ and $[\delta U_0]$ are of the order of $\Theta^2\gamma_n\, \|[L_0]\|\, \|U_0]\|$ and the orders of the magnitude of $[\delta L_k]$ and $[\delta U_k]$ are rapidly decreasing as k increases, where Θ is defined in Lemma 7.

The author recommend that we apply a single cycle(i.e. (Step1,2 and 3) of MAIR-LUF algorithm with $p = m = 1$ even if multiple-precision arithmetic is not availavble.

Acknowledgments

The author would like to thank Professor Shinichi Oishi of Waseda University for his thought-provoking introducition to the filed of Numerical Verification Method.

References

[1] S.M. Rump, “Approximate inverses of almost singular matrices still contain useful information,” *Tecnical Report 90.1, Technical University Hamburg-Harburg*, 1990.

[2] S. Oishi and S.M. Rump, “Fast verification of solutions of matrix equations,” *Numer. Math.* vol.90, pp.755-773, 2002.

[3] X.Li.J. Demmel et.al., “Design, implementation and testing of extended and mixed precision BLAS,”*ACM Trans. Math. Soft.*, vol.28, pp.152-205, 2002.

[4] N.J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, 2002.

[5] T. Ogita, S.M. Rump and S. Oishi, “Accurate sum and dot product,” *SIAM J. Sci. Comput.*, vol.26, pp.1955-1988, 2005.

[6] T. Ohta, T. Ogita, S.M. Rump and S. Oishi, “Numerical Verification Method for Arbitrarily Ill-conditioned Linear Systems,” *Trans. JSIAM*, to appear.

[7] K. Tanabe, “Additive-Form Iterative Refinement of Cholesky Factorization,” *Manuscript*, 2005.

[8] K. Tanabe, “Iterative Refinement of QR Factorization,” *Manuscript*, 2005.