Algorithms Research 2013, 2(2): 50-57
DOI: 10.5923/j.algorithms.20130202.03

Solving Ill Conditioned Linear Systems Using the Extended Iterative Refinement Algorithm: The Forward Error Bound

Abdramane Sermé

Department of Mathematics, The City University of New York, New York, NY, 10007, USA

Abstract This paper aims to provide a bound of the forward error of the extended iterative refinement or improvement algorithm used to find the solution to an ill conditioned linear system. We use the additive preconditioning for preconditioner of a smaller rank r and the Schur aggregation to reduce the computation of the solution to an ill conditioned linear system to the computation of the Schur aggregate S. We find S by computing W the solution of a matrix system using an extension of Wikinson iterative refinement algorithm. Some steps of the algorithm are computed error free and other steps are computed with errors that need to be evaluated to determine the accuracy of the algorithm. In this paper we will find the upper bound of the forward error of the algorithm and determine if its solution W can be considered accurate enough.

Keywords Forward Error Analysis, Ill Conditioned Linear System, Sherman-Morrison-Woodbury (SMW) Formula, Preconditioning, Schur Aggregation, Iterative Refinement or Improvement, Algorithm, Singular Value Decomposition

1. Introduction

We find the solution $x = A^{-1}b$ of an ill conditioned linear system $Ax = b$ by transforming it using the additive preconditioning and the Schur aggregation. We use the Sherman-Morrison-Woodbury (SMW) formula $A^{-1} = (C - UVH)^{-1} = C^{-1} + C^{-1}U(I_r - VHC^{-1}U)^{-1}VHC^{-1}$ where $A = C - UV^H$ is an invertible square matrix and $S = I_r - V^H C^{-1}U$ to get new linear systems. The challenge in solving these new linear systems of smaller sizes with well conditioned coefficients matrices $V^H C^{-1}$, $C^{-1}U$ and $S = I_r - V^H C^{-1}U$ is the computation of the Schur aggregate *S*. The technique of (extended) iterative refinement or improvement for computing the Schur aggregate[14] and its application for solving linear systems of equations has been studied in a number of papers[3, 15, 18]. Its variant that we used allows us to compute W with high precision. The high precision is achieved by minimizing the errors in the computation. The bound of the forward error will allow us to determine if the computed solution is an accurate one. This paper is divided into three sections. The first section covers the concept of rounding errors, floating-point summation, matrix norms and convergence. The second section is devoted to the additive preconditioning, the Schur aggregation and how the iterative refinement or improvement technique is used with the SMW formula to transform the original linear system $Ax = b$ into better conditioned linear systems. The third section analyzes the forward error of the extended iterative refinement or improvement algorithm and provides a forward error bound.

* Corresponding author:
aserme@bmcc.cuny.edu (Abdramane Sermé)
Published online at http://journal.sapub.org/algorithms

2. Rounding Errors, Floating-point Summation, Matrix Norms and Convergence

2.1. Rounding Errors

Definition 2.1.1 Let $\hat{x}$ be an approximation of the scalar *x*. The absolute error in $\hat{x}$ approximating x is the number $\varepsilon = |\hat{x} - x|$.

Definition 2.1.2 Let $\hat{x}$ be an approximation of a scalar x. The absolute and relative errors of this approximation are the numbers $|\hat{x} - x|$ and $\rho = \frac{|\hat{x} - x|}{|x|}$, respectively. If $\hat{x}$ is an approximation to x with relative error ρ, then there is a number $\frac{\hat{x} - x}{x}$ such that 1) $|r| = \rho$ and 2) $\hat{x} = x(1 + r)$.

Remark 2.1.1 The relative error ρ is independent of scaling, that is the scaling $x \to \alpha x$ and $\hat{x} \to \alpha\hat{x}$ leave ρ unchanged.

Theorem 2.1[6] Assume that $\hat{x}$ approximates x with

relative error $\rho < 1$.

Then $\hat{x}$ is nonzero and $\rho = \frac{|\hat{x} - x|}{|x|} \le \frac{\rho}{1-\rho}$.

Remark 2.1.2 If the relative error of x with respect to $\hat{x}$ is ρ, then x and $\hat{x}$ agree to roughly $-\log_2(\rho)$ correct significant digits. For binary system, if x and $\hat{x}$ have relative error of approximately 2^{-t-1}, then x and $\hat{x}$ agree to about t bits.

Definition 2.1.3 The componentwise relative error is defined as: $Max_i = \frac{|x_i - \hat{x}_i|}{|x_i|}$ for $x = (x_x, x_2, ..., x_{i,}...)$ and is widely used in the error analysis and perturbation theory.

Remark 2.1.3 In numerical computation, one has three main sources of errors.

1. Rounding errors, which are unavoidable consequences of working in finite precision arithmetic.

2. Uncertainty in the input data, which is always a possibility when we are solving practical problems.

3. Truncation errors, which are constituted and introduced by omitted terms.

Rounding errors and Truncation errors are closely related to forward errors.

Definition 2.1.4 Precision is the number of digits in the representation of a real number. It defines the accuracy with which the computations and in particular the basic arithmetic operations $+, -, \times, /$ are performed. For floating point arithmetic, precision is measured by the unit roundoff or machine precision, which we denote u in single precision and ū in double precision. The values of the unit roundoff are given in Table 1.1 in Section 2.3.

Remark 2.1.4 Accuracy refers to the absolute or relative error of an approximation.

Definition 2.1.5 Let $\hat{y}$ be an approximation of $y = f(x)$ computed with a precision u where f is a real function of a real scalar variable.

$\min\{|\Delta x| : \hat{y} = f(x + \Delta x)\}$ is called the (absolute) backward error, whereas the absolute or relative errors of $\hat{y}$ are called forward errors.

Definition 2.1.6 For an approximation $\hat{x}$ to a solution of a linear system $Ax = b$ with $(A \in C^{n \times n}$ and $b \in C^n)$, the forward error is the ratio $\frac{\|x - \bar{x}\|}{\|x\|}$.

The process of bounding the forward error of a computed solution in terms of u is called forward error analysis. Δx is the perturbation of x.

Definition 2.1.7 An algorithm is called forward stable if it produces answers with forward errors of similar magnitude to those produced by backward stable method.

Definition 2.1.8 A mixed forward-backward error is defined by the equation

$\hat{y} + \Delta\hat{y} = f(x + \Delta x)$ where $|\Delta\hat{y}| \le \varepsilon |y|$, $|\Delta x| = \eta |x|$ with ε and η are small constants.

Remark 2.1.5 This definition implies that the computed value ŷ differs little from the value $\hat{y} + \Delta\hat{y}$ that would have been produced by an input $x + \Delta x$ little different from the actual input x. Simpler, $\hat{y}$ is almost the right answer for almost the right data.

Definition 2.1.9 An algorithm is called numerically stable if it is stable in the mixed forward and backward error sense.

Remark 2.1.6 A backward stability implies a forward stability but the converse is not true.

Remark 2.1.7 One may use the following rule of thumb;

Forward error $\le$ condition number $\times$ backward error, with approximate equality possible. Therefore the computed solution to an ill conditioned problem can have a large forward error even if the computed solution has a small backward error. This error can be amplified by the condition number in the transition to forward error. This is one of our motivations for reducing the condition number of the matrix A using the additive preconditioning method.

Definition 2.1.10 For a system of linear equations $Ax = b$, $\rho(x) = \frac{\|b - Ax\|}{\|A\|\|x\|}$ is called the relative residual. The relative residual gives us an indication on how closely Ax represents b and is scale independent.

2.2. Floating-point Number System

Definition 2.2.1 [6] A floating-point number system F is a subset of the real numbers whose elements have the form $y = \pm m \times \beta^{e-t}$. The range of the nonzero floating-point numbers in F is given by $\beta^{e_{\min}-1} \le |y| \le \beta^{e_{\max}(1-\beta^{-t})}$. Any floating-point number $y \in F$ can be written in the form

$$Y = \left(\frac{d_1}{\beta} + \frac{d_2}{\beta^2} + ... + \frac{d_t}{\beta^t}\right) \times (\pm\beta^e) = .d_1 d_2 ... d_t \times (\pm\beta^e)$$

where each digit d_i satisfies $0 \le d_i \le \beta - 1$ and $d_1 \ne 0$ for normalized numbers. d_1 is called the most significant digit and d_t the least significant digit.

2.3. Error-free Floating-point Summation

Here is a summation algorithm due to D.E. Knuth[7]. Algorithm 1. *Error-free transformation of the sum of two floating point numbers*

$$function[x, y] = Twosum(a + b)$$
$$x = fl(a + b)$$
$$z = fl(x - a)$$
$$y = fl((a - (x - z)) + (b - z))$$

The algorithm transforms two input-floating point numbers a and b into two output floating-point numbers x and y such that $a+b=x+y$ and $x=fl(a+b)$. The same solution is achieved using the Kahan-Babušhka's[11] and Dekker's[12] classical algorithm provided that $|a|\geq|b|$. It uses fewer ops but includes branches, which slows down the code optimization outputs.

Definition 2.3.1[8] The unit roundoff error u is the quantity $u=\frac{1}{2}\beta^{1-t}$. We write u and $\bar{u}$ to denote the operations performed in single precision and in double precision, respectively.

Table 1.1. The values of the unit roundoff

Machine and arithmetic	β	t	$e_{\min}$	$e_{\max}$	unit roundoff u
IEEE Single	2	24	-125	128	$2^{-24}\approx 5.96\times 10^{-8}$
IEEE Double	2	53	-1021	1024	$2^{-53}\approx 1.11\times 10^{-16}$

Remark 2.3.1 The following theorem shows that every real number x lying in F can be approximated by an element of F with a relative error no larger than u.

Theorem 2.2 If $x\in\Re$ lies in F then $fl(x)=x(1+\delta)$ with $|\delta|<u$

Theorem 2.2 says that $fl(x)$ is equal to x multiplied by a factor very close to 1.

Definition 2.3.2 From now on $fl(.)$, for an argument that is an arithmetic expression, denotes the computed value of that expression. op represents floating-point operation in F.

2.4. Matrix Norms

2.4.1. The Singular Value Decomposition (SVD)[3, 15]

Definition 2.4.1 The compact singular value decomposition or SVD of an $m\times n$ matrix A of a rank ρ is the decomposition: $A=S(\rho)\sum(\rho)T(\rho)H=\sum_{j=1}^{\rho}\sigma_j s_j t_j^H$ where $S^{(\rho)}=(s_j)_{j=1}^{\rho}$ and $T^{(\rho)}=(t_j)_{j=1}^{\rho}$ are unitary matrices, that is, $S^{(\rho)H}S^{(\rho)}=I_\rho$, $T^{(\rho)H}T^{(\rho)}=I_\rho$, $\sum^{(\rho)}diag(\sigma_j)_{j=1}^{\rho}$ is a diagonal matrix, s_j and t_j are m- and n-dimensional vectors, respectively, and $\sigma_1\geq\sigma_2\geq\ldots\geq\sigma_\rho>0$. σ_j or $\sigma_j(A)$ for $j=1,\ldots,\rho$ is the j^{th} largest singular value of the matrix A.

Definition 2.4.2 The condition number, $cond_2 A$ of a matrix A of a rank ρ is $cond_2 A=\sigma_1(A)/\sigma_\rho(A)=\|A\|_2\|A^{-1}\|_2$. A matrix is said to be ill conditioned if its condition number is large, that is if $\sigma_1(A)\rangle\rangle\sigma_\rho(A)$, and is called well conditioned otherwise.

Definition 2.4.3[6] The matrix 2-norm $\|A\|_2=\sup_{|x|=1}|Ax|=\sigma_1(A)=\sigma_{\max}(A)$, is also called the spectral norm. $\sigma_1(A)$ denotes the largest singular value of the matrix A.

Remark 2.4.1 The matrix 2-norm satisfies the relation

1. $\|A\|\leq B\Rightarrow\|A\|_2\leq\|A\|_2\leq\|B\|_2$ where $|A|=(|a_{ij}|)_{i=1,j=1}^{m,n}$, $|a_{i,j}|\leq b_{ij}$ for $i=1,\ldots,m$ and $j=1,\ldots,n$.
2. $\|A\|_2\leq\|A\|_1\leq\sqrt{n}\|A\|_2$.

Lemma 2.3 For a vector norm $\|\|$, suppose that $\tilde{\rho}=\frac{\|\tilde{x}-x\|}{\|\tilde{x}\|}<1$. Then $\frac{\|\tilde{x}-x\|}{\|\tilde{x}\|}\leq\frac{\tilde{\rho}}{1-\tilde{\rho}}$.

2.5. Numerical Nullity

Definition 2.5.1 The nullity of A, $nulA=n-rankA$, is the smallest integer r for which a rank r APC UV^H can define a nonsingular A-modification $C=A+UV^H$. The nullity of A, which is defined as the dimension of the null space can also be defined as the large integer r for which we have $AC^{-1}U=0$ or $V^HC^{-1}A=0$, provided C is a nonsingular matrix. In this case, $C^{-1}U$ and V^HC^{-1} are the right and left null matrix bases for the matrix A.

Definition 2.5.2 The numerical nullity of A is the number of its small singular values.

2.6. Convergence

There is a natural way to extend the notion of limit from C to C^n.

Definition 2.6.1 Let $\{x_k=(x_1^{(k)},x_2^{(k)},\ldots,x_n^{(k)})^T\}_1^\infty$ be a sequence of vectors in C^n and let $x\in C^n$. The sequence $\{x_k\}_1^\infty$ converges componentwise to x and we write $\{x_k\}_1^\infty\to x$ if $\lim_{k\to\infty}x_i^{(k)}=x_i$ for $i=1,2,\ldots,n$.

Here is another way to define convergence in C^n.

Definition 2.6.2 Let $\{x_k=(x_1^{(k)},x_2^{(k)},\ldots,x_n^{(k)})^T\}_1^\infty$ be a sequence of vectors in C^n and let $x\in C^n$. The sequence $\{x_k\}_1^\infty$ converges normwise to x, that is $\lim_{k\to\infty}x_k=x$ if and only if $\lim_{k\to\infty}\|x_k-x\|=0$

There is no compelling reason to expect the two notions of

convergence to be equivalent. In fact for infinite dimensional vector space, they are not.

Theorem 2.4[6] Let $P \in C^{nxn}$ and suppose that $\lim_{k\to\infty} P^k = 0$.

Then $I-P$ is nonsingular and $(I-P)^{-1} = \sum_{k=0}^{\infty} P^k$ (this sum is called Neumann sum). A sufficient condition for $P^k \to 0$ is that $\|P\| < 1$ in some consistent norm, in which case $\left\|(I+P+P^2+...+P^k)-(I-P^{-1})\right\| \le \frac{\|P\|^{k+1}}{1-\|P\|}$.

Corollary 2.5 If $P^k \to 0$, then $(I-|P|)^{-1}$ is nonnegative and $(I-P)^{-1} \le (I-|P|)^{-1}$.

Theorem 2.6[6] Let $\|.\|$ be a matrix norm on C^{nxn} consistent with a vector norm (also denoted $P\|.\|$) and let a matrix $X \in C^{nxn}$. Let P be a square matrix such that $\|P\| < 1$. Then

(i) the matrix $I-P$ is nonsingular,

(ii) $\left\|(I-P)^{-1}X\right\| \le \frac{\|X\|}{1-\|P\|}$, and

(iii) $\left\|(I-P)^{-1}-I\right\| \le \frac{\|P\|}{1-\|P\|}$.

The following corollary extends Theorem 2.6.

Corollary 2.7[6]

If $\|A^{-1}E\| \le 1$, then $\left\|(A+E)^{-1}\right\| \le \frac{\|A^{-1}\|}{1-\|A^{-1}E\|}$.

Moreover, $(A+E)^{-1}-A^{-1} = [(I-A^{-1}E)-I]A^{-1}$,

so that $\left\|(A+E)^{-1}-A^{-1}\right\| \le \frac{\|A^{-1}\|\|A^{-1}E\|}{1-\|A^{-1}E\|}$.

The corollary remains valid if all occurrences of $\|A^{-1}E\|$ are replaced by $\|EA^{-1}\|$.

Theorem 2.8 [8, 17] Let $\|.\|$ denote a matrix norm and a consistent vector norm. If the matrix A is nonsingular and $Ax=b$ and (*i*) $\tilde{A}\tilde{x}=b$, where $\tilde{x}$ is an approximated value of x, then

(*ii*) $\frac{\|\tilde{x}-x\|}{\|\tilde{x}\|} \le \|A^{-1}E\|$.

In addition if $\|A^{-1}E\| < 1$, then $\tilde{A}=A+E$ is nonsingular and

(*iii*) $\frac{\|\tilde{x}-x\|}{\|x\|} \le \frac{\|A^{-1}E\|}{1-\|A^{-1}E\|}$.

3. The Additive Preconditioning Method, the Schur Aggregation and the Extended Iterative Refinement or Improvement Algorithm

3.1. The Additive Preconditioning Method

Definition 3.1.1 For a pair of matrices U of size $m\times r$ and V of size $n\times r$, both having full rank $r>0$, the matrix UV^H of rank r is an additive preprocessor (APP) of rank r for any $m\times n$ matrix A. The matrix $C=A+UV^H$ is the A-modification. The matrices U and V are the generators of the APP, and the transition $A\to C$ is an A-preprocessing of rank r for the matrix A. An APP UV^H for a matrix A is an additive preconditioning (APC) and an A-preprocessing is an A-preconditioning if $cond_2 A \rangle\rangle cond_2 C$. An APP is an additive compressor (*AC*) and an A-preprocessing is an A-complementation if the matrix A is rank deficient, whereas the A-modification C has full rank. An APP UV^H is unitary if the matrices U and V are unitary.

Remark 3.1.1 Suppose UV^H has rank r. Then[3, 18], we expect $cond_2 C = \sigma_1(C)/\sigma_n(C)$ to be to the order of $\sigma_1(A)/\sigma_{n-r}(A)$, therefore small if the additive preconditioner UV^H is

i) random

ii) well conditioned, and

iii) properly scaled, that is $\|A\|/\|UV^H\|$ is not large and not small.

Additive preconditioning consists in adding a matrix UV^H of a small rank to the input matrix A, to decrease its condition number. The A-modification is supposed to generate a well conditioned matrix C. In practice, to compute the A-modification $C=A+UV^H$ error-free, we fill the generators U and V with short binary numbers.

3.2. The Schur Aggregation

The aggregation method consists of transforming an original linear system $Ax=b$ into linear systems of smaller sizes with well conditioned coefficients matrices $V^H C^{-1}$, $C^{-1}U$, and $S=I_r - V^H C^{-1} U$. The aggregation method is a well known technique[3, 14, 15, 18], but aggregation used here both decreases the size of the input matrix and improves its conditioning. One may remark that aggregation can be applied recursively until no ill conditioned matrix appears in the computation.

Definition 3.2.1 The Schur aggregation is the process of reducing the linear system $Ax = b$ by using the SMW (Sherman-Morrison-Woodbury) formula

$A^{-1} = (C - UV^H)^{-1} = C^{-1} + C^{-1}U(I_r - V^H C^{-1}U)^{-1}V^H C^{-1}$. The matrix $S = I_r - V^H C^{-1}U$, which is the Schur complement (Gauss transform) of the block C in the block matrix $\begin{bmatrix} C & U \\ V^H & I_r \end{bmatrix}$, is called the Schur aggregate. The A-modification $C = A + UV^H$ and the Schur aggregate S are well conditioned, therefore the numerical problems in the inversion of the matrix A are confined to the computation of the Schur aggregate $S = I_r - V^H C^{-1}U$.

3.3. The Iterative Refinement or Improvement Algorithm

Let $C = A + UV^H$. Then, we apply the Sherman-Morrison -Woodbury (SMW) formula to the original linear system $Ax = b$ and transform it into better conditioned linear systems of small sizes, with well conditioned matrices $V^H C^{-1}$, $C^{-1}U$ and $S = I_r - V^H C^{-1}U$. We solve the original linear system $Ax = b$ by post-multiplying $A^{-1} = C^{-1} + C^{-1}U(I_r - V^H C^{-1}U)^{-1}V^H C^{-1}$ by the vector b. We consider the case where the matrices C and S are well conditioned, whereas the matrices U and V have small rank r, so that we can solve the above linear systems with the matrices C and S faster and more accurately than the system with the matrix A. In this case the original conditioning problems for a linear system $Ax = b$ are restricted to the computation of the Schur aggregate S.

To compute the Schur aggregate $S = I_r - V^H C^{-1}U$ with precision, we begin with computing $W = C^{-1}U$ using the iterative refinement or improvement algorithm. We prove that we can get very close to the solution W of the linear system $CW = U$. We closely approximate it by working with numbers rounded to the IEEE standard double precision and using error-free summation. All norms used in this section are the 2-norm. The iterative refinement or improvement algorithm is a technique for improving the computed approximate solution $\hat{x}$ of a linear system $Ax = b$. Iterative refinement or improvement for the Gaussian Elimination (GE) was used in the 1940s on desk calculators, but the first thorough analysis of the method was given by Wilkinson in 1963. The process consists of three steps ([6],[7],[15]).

Algorithm 2. *Basic iterative refinement or improvement algorithm*

Input: An $n \times n$ *matrix* A*, a computed solution* $\hat{x} = x_1$ *to* $Ax = b$ *and a vector b.*

Output: A solution vector x_i *approximating* x *in* $Ax = b$ *and an error bound* $\frac{\|x - x_i\|}{\|x\|}$.

Initialize: $i \leftarrow 1$

Computations:

1) Compute the residual $r_i = b - Ax_i$ *in double precision (*$\overline{u}$*)*

2) Solve $Ad_i = r_i$ *in single precision (*u*) using the GEPP*

3) Update $x_{i+1} = x_i + d_i$ *in double precision (*$\overline{u}$*)*

$i \leftarrow i + 1$

Repeat stages 1-3 until x_i *is accurate enough.*

Output x_i *and an error bound.*

The iterative refinement or improvement algorithm can be rewritten as follows[6]. g_0 is the error in the computation of r_0, h_0 is the error in the computation of x_1 and $A_0 = A + E_0$, where E_0 is the perturbation to the matrix A. x_0 is a computed solution of the linear system $Ax = b$.

1) $r_0 = b - Ax_0 + g_0$
2) $d_0 = A_0^{-1} r_0$
3) $x_1 = x_0 + d_0 + h_0$

Repeat stages 1-3.

The algorithm yields a sequence of approximate solutions $x_0, x_1 \ldots$ which converges to $x = A^{-1}b$.

We use the extension of Wilkinson's iterative refinement or improvement to compute the matrix $W = C^{-1}U$ with extended precision. In its classical form, above the algorithm is applied to a single system $Cw = u$, where w and u are $n \times 1$ vectors. We applied it to the matrix equation $CW = U$, where the solution we are seeking is the matrix W. In its classical version, also the refinement stops when the matrix $W = C^{-1}U$ is computed with at most double precision. In order to achieve the high precision in computing W, we apply a variant of the extended iterative refinement or improvement where the residuals dynamically decrease, which is a must for us. We represent the output value as the sum of matrices with fixed-precision numbers.

3.4. The Extended Iterative Refinement or Improvement

Suppose A is an ill conditioned non-singular $n \times n$ matrix with $nnulA = r$ where $nnulA$ is the numerical nullity of the matrix A, UV^H is a random, well conditioned and properly scaled APC of rank $r < n$, and the well conditioned A-modification $C = A + UV^H$. We use the matrices U and V whose entries can be rounded to a fixed (small) number of bits to control or avoid rounding errors in computing the matrix $C = A + UV^H$.

Surely, small norm perturbations of the generators U and V, caused by truncation of their entrees, keep the matrix C well conditioned. We rewrite the iterative refinement or improvement algorithm to solve the linear system $CW = U$ with $U_0 = U$ and $W_0 = C^{-1}U_0 = X_0$ as follows.

Algorithm 3.

$$CW_k = U_k \tag{0.1}$$

$$U_{k+1} = U_k - CW_k \tag{0.2}$$

$$X_k = W_0 + \ldots + W_k \text{ for } k = 0,1,2,\ldots \tag{0.3}$$

The solution W of the linear system $CW = U$ is computed by means of Gaussian Elimination with Partial Pivoting (hereafter GEPP), which is a backward stable process. It is corrupted by rounding errors of the computation of W_k in (0.1), so that the computed matrix W_k (computed in single precision arithmetic u) turns into $(C + E_k)^{-1}U_k$. E_k is the perturbation to the matrix C. We can also say equivalently that there exists an error matrix E_k such that

$$(C + E_k)W_k = U_k \text{ where } \|E_k\| \le c(k)u\|C\| \tag{0.4}$$

that is, W_k is an exact solution for the approximated problem. $c(k)$ is a constant function of order k. Another source of error is the computation in (0.2) which, done using double precision arithmetic $\bar{u}$, turns numerically into

$$U_k = fl(U_{k-1} - CW_{k-1}) = U_{k-1} - CW_{k-1} + \Delta E_k \tag{0.5}$$

where $$\|\Delta E_k\| \le c_1(k)\bar{u}(\|C\|\|W_{k-1}\| + \|U_{k-1}\|) \tag{0.6}$$

$$X_k = fl(W_{k-1} + W_k) = W_{k-1} + W_k. \tag{0.7}$$

We recall that the summation (0.3) is done error free.

Algorithm 4. *Let us solve the linear system $CW = U$, derived from the ill conditioned linear system $Ax = b$, by applying the following extended iterative refinement or improvement algorithm.*

$$W_0 = C_0^{-1}U_0 \quad (U_0 = U \text{ and } C_0 = C + F_0)$$

$$W_k = (C + F_k)^{-1}U_k \tag{0.8}$$

$$U_{k+1} = U_k - CW_k + E_k \tag{0.9}$$

$$X_k = W_0 + \ldots + W_k, \text{ for } k = 0,1,2,\ldots$$

Let $F_k = C_k - C$.

Theorem 3.1 [1]

$$\text{If } \frac{\|C^{-1}F_k\|}{1 - \|C^{-1}F_k\|} \le \rho < 1 \text{ and} \tag{0.10}$$

$$\|E_k\| \le \gamma_k \text{ for } k = 0,1,\ldots, \tag{0.11}$$

then

$$\|X_k - X\| \le \rho^k \|X_0 - X\| + (1+\rho)\|C^{-1}\|(\gamma_k + \rho\gamma_{k-1} + \ldots + \rho^{k-1}\gamma_1).$$

In other words, $\|X_k - X\|$ is bounded by $O(\gamma_k)$ for a certain integer k.

Proposition 3.2 Let $C \in C^{n\times n}$ be nonsingular and let consider the linear system $CW_k = U_k$. If $c(k)condCu < \rho < 1$, then the matrix $(C + E_k)$ is nonsingular and

$$(C + E_k)^{-1} = (I + F_k)C^{-1} \text{ where}$$

$$\|F_k\| \le \frac{c(k)condCu}{1 - c(k)condCu} \tag{0.12}$$

or equivalently,

$$(C + E_k)^{-1} = C^{-1}(I + F_k).$$

4. Forward Error Analysis

Our forward error analysis of the extended iterative refinement or improvement algorithm results with the following proposition.

Proposition 4.1 (Forward error bound) Let $CW = U$ be a linear system derived from an ill conditioned linear system $Ax = b$ where U is $m \times r$, V is $n \times r$ both with full rank $r > 0$, and $C = A + UV^H$ is a well conditioned A-modification. If $CW = U$ is solved using the extended iterative refinement or improvement algorithm, Algorithm (4.) then for sufficiently large k the forward error $\frac{\|X - X_k\|}{\|X\|}$ is bounded by $\frac{4condCc_1(k)\bar{u}}{1 - \alpha_1}$. That is

$$\frac{\|X - X_k\|}{\|X\|} \le \frac{4condCc_1(k)\bar{u}}{1 - \alpha_1}$$

where $c_1(k)$ and $\alpha_1 = \left(\frac{c(k)}{1 - c(k)condCu} + 4c_1(k)\right)condCu$ are constant functions of order k. Furthermore we have

$$\frac{\|X - X_k\|}{\|X\|} \le O(u).$$

The forward error is bounded by a constant in the order of u which is an important result.

Proof: We obtain from equation (0.4),

$W_k = (C + E_k)^{-1}U_k$.

Since $X - X_k = X - W_{k-1} - W_k$, we get

$X - X_k = X - W_{k-1} - (C + E_k)^{-1}U_k$ by using (0.4).

$X - X_k = X - W_{k-1} - (C + E_k)^{-1}(U_{k-1} - CW_{k-1} + \Delta E_k)$

by using (0.5),

$X - X_k = X - W_{k-1} - (I + F_k)C^{-1}(U_{k-1} - CW_{k-1} + \Delta E_k)$

by using (0.12) in Proposition (3.2),

$X - X_k = X - W_{k-1} - (I + F_k)(C^{-1}U_{k-1} - W_{k-1} + C^{-1}\Delta E_k)$,

$X - X_k = X - W_{k-1} - (I + F_k)(C^{-1}U_k + C^{-1}\Delta E_k)$ by using (0.2).

We have,

$X_k = W_{k-1} + W_k$,

$X_k = W_{k-1} + C^{-1}U_k$,

$X_k - W_{k-1} = C^{-1}U_k$, so that

$X - X_k = X - W_{k-1} - (I + F_k)(X_k - W_{k-1} + C^{-1}\Delta E_k)$,

$X - X_k = X - W_{k-1} - (X_k - W_{k-1}) - F_k(X_k - W_{k-1}) - (I + F_k)C^{-1}\Delta E_k$

Consequently

$X - X_k = X - W_{k-1} + X_k + W_{k-1} - F_k(X_k - W_{k-1}) - (I + F_k)C^{-1}\Delta E_k$

$X - X_k = -(X - X_k) - F_k(X_k - W_{k-1}) - (I + F_k)C^{-1}\Delta E_k$, so

$2(X - X_k) = -F_k(X_k - W_{k-1}) - (I + F_k)C^{-1}\Delta E_k$.

Without loss of generality we can assume that $X - X_k = -F_k(X_k - W_{k-1}) - (I + F_k)C^{-1}\Delta E_k$.

Recall that F_k < 1, and take the norm on both sides, to get

$\|X - X_k\| \le \|F_k\|\|X_k - W_{k-1}\| + 2\|C^{-1}\|\|\Delta E_k\|$.

Recalling (0.6) and (0.7), we deduce that

$\|X - X_k\| \le \|F_k\|\|X_k - W_{k-1}\|$

$+2\|C^{-1}\|c_k(k)\overline{u}(\|C\|W_{k-1}\| + \|U_{k-1}\|\|)$

We recall the following inequalities (0.12),

$\|F_k\| \le \dfrac{c(k)condCu}{1 - c(k)condCu} < 1.$

From (0.7) $X_k = W_{k-1} + W_k$,

Therefore $W_k = X_k - W_{k-1}$,

$W_k = X - X_{k-1} + W_k - X + X_{k-1}$

$\|W_k\| \le \|X - X_{k-1}\| + \|X\|$.

So $\|X_k - W_{k-1}\| \le \|W_k\| \le \|X - X_{k-1}\| + \|X\|$.

We recall (0.6) $\|\Delta E_k\| \le c_1(k)\overline{u}(\|C\|\|W_{k-1}\| + \|U_{k-1}\|)$.

We also have

$\|W_{k-1}\| \le \|X - X_{k-1}\| + \|X\|$

$\|U_{k-1}\| \le \|C\|\|W_k\|$

$\|U_{k-1}\| \le \|C\|\|X - X_{k-1}\| + \|C\|\|X\|$.

$\|X - X_k\| \le \dfrac{c(k)condCu}{1 - c(k)condCu}(\|X - X_{k-1}\|)$

$+2\|C^{-1}\|c_1(k)\overline{u}(\|C\|\|X - X_{k-1}\|)$

$+\|C\|\|X\| + \|C\|\|X - X_{k-1}\| + \|C\|\|X\|$

$\|X - X_k\| \le \left[\dfrac{c(k)u}{1 - c(k)condCu} + 2\|C^{-1}\|\|C\|c_1(k)\overline{u}.2\right]\|X - X_{k-1}\|$

$+4\|C^{-1}\|\|C\|c_1(k)\overline{u}\|X\|$

$\|X - X_k\| \le \left[\dfrac{c(k)condCu}{1 - c(k)condCu} + 4condCc_1(k)\overline{u}\right]\|X - X_{k-1}\|$

$+4condCc_1(k)\overline{u}\|X\|$

$\|X - X_k\| \le \left[\dfrac{c(k)condCu}{1 - c(k)condCu} + 4condCc_1(k)\overline{u}\right]\|X - X_{k-1}\|$

$+4condCc_1(k)\overline{u}\|X\|$

$\|X - X_k\| \le \alpha_1\|X - X_{k-1}\| + \alpha_2\|X\|$ where

$\alpha_1(k) = \left(\dfrac{c(k)}{1 - c(k)condCu} + 4c_1(k)\right)condCu$ and

$\alpha_2(k) = 4condCc_1(k)\overline{u}$.

$\alpha_1(k), \alpha_2(k)$ and $c(k)$ are constant functions of order k.

$|\alpha_1(k)|$ < 1 and $\dfrac{c(k)condC\overline{u}}{1 - c(k)condCu}$ < 1 since C is well conditioned, we deduce that

$\|X - X_{k-1}\| \le \alpha_1(k)\|X - X_{k-2}\| + \alpha_2(k)\|X\|$.

Therefore, $\|X - X_k\| \le \alpha_1^2(k)\|X - X_{k-2}\|$
$+\alpha_1(k)\alpha_2(k)\|X\| + \alpha_2(k)\|X\|$.

We also recall that

$\|X - X_{k-2}\| \le \alpha_1(k)\|X - X_{k-3}\| + \alpha_2(k)\|X\|$,

so that

$\|X - X_k\| \le \alpha_1^3(k)\|X - X_{k-3}\| + \alpha_1^2(k)\alpha_2(k)\|X\| + \alpha_2(k)\|X\|$

and $\|X - X_k\| \le \alpha_1^4(k)\|X - X_{k-4}\| + \alpha_1^3(k)\alpha_2(k)\|X\| \ldots$
$+\alpha_1(k)\alpha_2(k)\|X\| + \alpha_2(k)\|X\|$

$\|X - X_k\| \le \alpha_1^{k-1}(k)\|X - X_1\| + (\alpha_1^{k-2}(k)$

$+\alpha_1^{k-3}(k) + \ldots + 1)\alpha_2(k)\|X\|$

$\|X - X_k\| \le \alpha_1^{k-1}(k)\|X - X_0\| + (1 + \alpha_1(k) + \ldots$

$+\alpha_1^{k-3}(k) + \alpha_1^{k-2}(k))\alpha_2(k)\|X\|$

$\|X - X_k\| \le \alpha_1^{k-1}(k)\|X - X_0\| + \dfrac{1 - \alpha_1^{k-1}(k)}{1 - \alpha_1(k)}\alpha_2(k)\|X\|$.

Therefore, $\lim\limits_{x\to\infty}\|X - X_k\| \le \dfrac{\alpha_2(k)}{1 - \alpha_1(k)}\|X\|$

$\lim\limits_{x\to\infty}\|X - X_k\| \le \dfrac{4condCc_1(k)\overline{u}}{1 - \alpha_1(k)}\|X\|$

$\lim\limits_{x\to\infty}\dfrac{\|X - X_k\|}{\|X\|} \le \dfrac{4condCc_1(k)\overline{u}}{1 - \alpha_1(k)}$.

Therefore for sufficiently large k, we have

$\dfrac{\|X - X_k\|}{\|X\|} \le \dfrac{4condCc_1(k)\overline{u}}{1 - \alpha_1(k)}$. Moreover,

$\dfrac{\|X - X_k\|}{\|X\|} \le \dfrac{4condCc_1(k)\overline{u}}{1 - \alpha_1(k)} \le \dfrac{1}{1 - \alpha_1(k)}.u$. So,

$\frac{\|X - X_k\|}{\|X\|} \leq \beta(k).u$ where $\beta(k) = \frac{1}{1-\alpha_1(k)}$ and $\alpha_1(k)$ are constant functions of order k. Therefore, $\frac{\|X - X_k\|}{\|X\|} \leq \mathrm{O}(u)$.

5. Conclusions

We use the concepts of additive preconditioning and Schur aggregation along with the extended iterative refinement or improvement algorithm to reduce the computation of $x = A^{-1}b$ to the computation of the Schur aggregate $S = I_r - V^H C^{-1} U$. We solve the linear system $W = C^{-1}U$ with high precision using the extended iterative refinement or improvement algorithm. We proved in our forward error analysis that the forward error $\frac{\|X - X_k\|}{\|X\|}$ is bounded by $\frac{4condCc_1(k)\overline{u}}{1-\alpha_1}$. The forward error can further be bounded by $\mathrm{O}(u)$, a constant of order u. These results are in line with Higham's results ([6], page 234, Theorem 11.1) and constitute another way to prove the convergence of the extended iterative refinement or improvement to a more accurate solution

$$\begin{aligned} & A^{-1}b \\ & = (C - UV^H)^{-1}b \\ & = C^{-1}b + C^{-1}U(I_r - V^H C^{-1}U)^{-1}V^H C^{-1}b \end{aligned}.$$

ACKNOWLEDGEMENTS

The author would like to thank Professor Victor Pan, Distinguished professor of The City University of New York for his support and advice. The author would also like to thank his wife Lisa C. Serme for her support.

REFERENCES

[1] A. Serme, J. W. Richard, The Schur Aggregation and Solving Ill Conditioned Linear Systems: The convergence theorem, Afrika Matematika DOI: 10.1007/s13370-012-0066-x, March 2012.

[2] A. Serme, On Iterative Refinement/Improvement of the Solution to an Ill Conditioned Linear System, Ph.D. thesis under the surpervision of Professor Victor Y. Pan, CUNY Ph.D. Program in Mathematics, Graduate Center, The City University of New York, 2008.

[3] V. Y. Pan et al., Additive preconditioning and aggregation in matrix computations, Computers and mathematics with applications, 55(8), 1870-1886, 2008.

[4] V. Y. Pan, Structured Matrices and Polynomials: Unified Superfast Algorithms, Boston/New York: Birkhäuser/ Springer, 2001.

[5] V. Y. Pan, Y. Yu, Certification of Numerical Computation of the Sign of the Determinant of a Matrix, Algorithmica, vol. 30, 708-724, 2001.

[6] G. W. Stewart, Matrix Algorithms, Vol. I: Basic Decompositions, Philadelphia: SIAM, 1998.

[7] D. E. Knuth, The Art of Computer Programming: Volume 2, Seminumerical Algorithms, Reading, Massachusetts: Addison-Wesley, 1969 (first edition), 1981 (second edition), 1998 (third edition).

[8] N. J. Higham, Accuracy and Stability in Numerical Analysis, Philadelphia: SIAM, 2002 (second edition).

[9] T. Ogita, S. M. Rump, S. Oishi, Accurate Sum and Dot Product, SIAM Journal on Scientific Computing, 26(6), 1955-1988, 2005.

[10] S. M. Rump, T. Ogita, S. Oishi, Accurate Floating-Point Summation, Tech. Report 05.12, Faculty for Information and Communication Sciences, Hamburg University of Technology, November 2005.

[11] J I. Babuška, Numerical Stability in Mathematical Analysis, Information Processing, 68 (Proc. of IFIP Congress), North-Holland, Amsterdam, pp. 11-23, 1969.

[12] T. J. Dekker, A Floating-Point Technique for Extending the Available Precision, Numerische Math., vol. 18, pp. 224-242, 1971.

[13] N. J. Higham, The Accuracy of Floating Point Summation, SIAM Journal on Scientific Computing, vol. 14, pp. 783-799, 1993.

[14] J. Demmel, Y. Hida, W. Kahan, X. S. Li, Soni Mukherjee, E. J. Riedy, Error Bound from Extra Precise Iterative Refinement, Computer Science Division, Technical Report UCB//CSD-04-1344, University of California, Berkeley, February 2005.

[15] W. L. Miranker, V. Y. Pan, Methods of Aggregations, Linear Algebra and Its Application, vol. 29, pp. 231-257, 1980.

[16] G. H. Golub, C. F. Van Loan, Matrix Computations, 3rd edition, Baltimore, Maryland: The Johns Hopkins University Press, 1996.

[17] G. W. Stewart, Matrix Algoritms, Vol II: Eigensystems, Philadelphia: SIAM, 1998.

[18] V. Y. Pan, D. Ivolgin, B. Murphy, R. E. Rosholt, M. Tabanjeh, The Schur aggregation for solving linear systems of equations, SNC'07, July 25-27, 2007.