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

Numerical Verification Method for Dense Linear Systems with Arbitrarily Ill-conditioned Matrices

Takahisa OHTA[†], Takeshi OGITA[*,†], Siegfried M. Rump[‡], and Shin'ichi OISHI[†,*]

†Faculty of Science and Engineering, Waseda University
3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555, Japan
∗ CREST, Japan Science and Technology Agency
‡Inst. f. Computer Science III, Hamburg University of Technology
Schwarzenbergstr. 95, Hamburg 21071, Germany,
Email: tohta@fuji.waseda.jp, ogita@waseda.jp, rump@tu-harburg.de, oishi@waseda.jp

Abstract—This paper is concerned with the problem of verifying an accuracy of a computed solution of linear systems with an arbitrarily ill-conditioned coefficient matrix. In this paper, a method of obtaining an accurate computed solution of such linear systems and its verified error bound is proposed. The proposed method is based on the accurate computation of dot product and IEEE standard 754 arithmetic. A verified and accurate computed solution with a desired tolerance can be obtained by the proposed method with iterative refinement. Numerical results are presented for illustrating the effectiveness of the proposed method.

1. Introduction

We are concerned with numerical verification method for a computed solution $\tilde{x}$ of a linear system

$$Ax = b, \tag{1}$$

where A is a real $n \times n$ matrix and b is a real n-vector. Condition number of A is defined by

$$\kappa(A) := \|A\| \cdot \|A^{-1}\|. \tag{2}$$

When A is nonsingular and b is perturbed to $b + \Delta b$, the exact solution is perturbed to of the order $\kappa(A) \cdot \|\Delta b\|$, i.e. for a perturbed linear system $A(x + \Delta x) = b + \Delta b$ it holds that

$$\frac{\|\Delta x\|}{\|x^*\|} \leq \kappa(A) \frac{\|\Delta b\|}{\|b\|}. \tag{3}$$

When using double precision floating-point arithmetic defined by IEEE standard 754, the relative precision of mantissa of a floating-point number is $2^{-53} = 1.11 \cdots \times 10^{16}$. Therefore if condition number is larger than of the order 10^{16}, then the computed solution, which has rounding errors, has only one or no correct digit in mantissa, so that it is the limit of calculation in double precision. In such case, we can use higher precision arithmetic. However, if all calculations are done using multiple-precision arithmetic, it requires significant computational cost.

To overcome this, we develop a fast method of calculating an accurate approximate inverse of A whose condition number can be arbitrarily large. In the proposed method, higher precision arithmetic is used only for dot product including matrix-vector product and matrix-matrix product. Using the method, we propose a fast verification algorithm for extremely ill-conditioned linear systems without multiple-precision arithmetic except for dot product.

2. Accurate Dot Product

Let $\mathbb{F}$ be a set of IEEE 754 double precision floating-point numbers. For vector $x, y \in \mathbb{F}^n$ calculating a dot product

$$x^T y = \sum_{i=1}^{n} x_i y_i$$

plays a prominent role in scientific computing, so that several algorithms of calculating the dot product have been developed, among them

- super-long accumulator for dot product [1]
- doubly compensated summation [6, 2]
- XBLAS [3]
- fast arbitrary precision sum and dot product [4]

To know details, see references. In particular, Higham's book [2, Chapter 4] is very readable.

In general, computational speed of multiple-precision arithmetic is some hundred times slower than that of double precision arithmetic because such multiple-precision arithmetic is implemented by software simulation while the double precision arithmetic can be executed by hardware instructions. Limited to the computation of dot product, however, higher precision arithmetic can be executed in reasonable computational cost only some ten times slower than the double precision arithmetic.

In this paper we assume that a function

`DotExact`(*expression*, *parameter*)

calculates the *expression* term as if computed in exact arithmetic and rounded to (sum of) double precision number(s) according to the *parameter* term. For example, for

$A \in \mathbb{F}^{m\times p}$, $B \in \mathbb{F}^{p\times n}$, the operation

$$C_{1:k} = \texttt{DotExact}(AB, k)$$

calculates $C_i \in \mathbb{F}^{m\times n}$, $i = 1, 2, \ldots, k$ such that

$$\left| \sum_{i=1}^{k} C_i - A \cdot B \right| \leq \max(2^{-52}|C_k|, realmin \cdot E), \qquad (4)$$

where $|C_i| \geq 2^{52}|C_{i+1}|$ for $i = 1, 2, \ldots, k-1$, $realmin := 2^{-1022}$ which is the smallest positive normalized floating-point number of IEEE 754 double precision and E is an $m \times n$ matrix of all ones. These are necessary for rigorousness of the result verification including the presence of underflow.

3. Proposed Method

In this section we propose a fast verification method of calculating a componentwise error bound of a computed solution of a linear system $Ax = b$ with A being an arbitrarily ill-conditioned matrix.

3.1. Componentwise Verification Method

Yamamoto's theorem [9] is useful as calculating a componentwise error bound of a computed solution of $Ax = b$.

Theorem 1 (Yamamoto [9]) *Let A be a real $n \times n$ matrix and b a real n-vector. If there exists a real $n \times n$ matrix R for $G := RA - I$ with the $n \times n$ identity matrix I such that*

$$\|G\|_\infty < 1, \qquad (5)$$

then A is nonsingular and for an arbitrary real n-vector $\tilde{x}$

$$|\tilde{x} - A^{-1}b| \leq |R(A\tilde{x} - b)| + \frac{\|R(A\tilde{x} - b)\|_\infty}{1 - \|G\|_\infty} t, \qquad (6)$$

where t satisfies

$$\left(\sum_{j=1}^{n} |G_{1j}|, \ldots, \sum_{j=1}^{n} |G_{nj}| \right)^T \leq t. \qquad (7)$$

In practice, an approximate inverse of A and an approximate solution of $Ax = b$ are chosen as R and as $\tilde{x}$, respectively. In our proposed method, we set $\tilde{x}$ as an approximate solution of $Ax = b$ using iterative refinement through an accurate computation of residual $A\tilde{x} - b$ by `DotExact` introduced in Section 2. A preconditioner R is calculated with the product-type iterative method proposed in next section for satisfying $\|RA - I\|_\infty < 1$ by use of `DotExact`.

3.2. Accurate Approximate Inverse

When condition number of A is greater than 10^{16}, it is difficult to apply the verification methods (e.g. [5]) with the IEEE 754 double precision computation because it might become $\|RA - I\|_\infty \geq 1$. To overcome this we will propose a method of calculating R which satisfies $\|RA - I\|_\infty < 1$ using higher precision arithmetic.

In order to utilize the property of IEEE 754 double precision floating-point arithmetic, we consider to express R as the sum of double precision floating-point matrices such as

$$R_{1:k} := \sum_{i=1}^{k} R_i \quad \text{for } R_i \in \mathbb{F}^{n\times n},$$

where $|R_i| \geq 2^{52} \cdot |R_{i+1}|$, $i = 1, 2, \ldots, k-1$. Here we propose the following algorithm which is an extended version of the method presented in [8]. In this paper we express algorithms by a MATLAB-like style.

Algorithm 1 Calculation of an accurate approximate inverse $R = \sum_{i=1}^{k} R_i$ of a real square matrix A:

```
function R^(k)_{1:k} = AccInv(A, k)
R^(1)_1 = inv(A)
for i = 2 : k
    C = DotExact(R^(i-1)_{1:i-1} · A, 1)
    T = inv(C)
    R^(i)_{1:i} = DotExact(T · R^(i-1)_{1:i-1}, i)
end
```

We observe that Algorithm 1 can be applied to the problem with condition number $\kappa(A)$ satisfying

$$\kappa(A) \lesssim 10^{16(k-1)} \sim 10^{16k}. \qquad (8)$$

We will confirm it by numerical experiments in Section 4.

Normally, the condition number $\kappa(A)$ is not known in advance, so that it is preferable to calculate R satisfying $\|RA-I\|_\infty < 1$ by iterative refinement automatically adapted to the condition of A. In fact, it is possible to design such an algorithm with a stopping criterion, say $fl(\|C - I\|_\infty) \leq 10^{-3}$, for the iterative refinement. In addition, using (4) and rounding mode controlled computation [5], an inclusion of $RA - I$ and therefore an upper bound of $\|RA - I\|_\infty$ can be calculated. Thus, we can verify the nonsingularity of A and obtain an accurate approximate inverse R of A.

3.3. Verification Method for Linear Systems

Let r be a residual for an approximate solution $\tilde{x}$ of a linear system $Ax = b$ as $r := A\tilde{x} - b$. If A is nonsingular, error of $\tilde{x}$ for the exact solution $x^* := A^{-1}b$ is given by a solution $p^* := A^{-1}r$ of a linear system $Ap = r$ and $x^* = \tilde{x} - p^*$. Here, we assume that $\|r\| \neq 0$ because if A is nonsingular and elements of r are all zeros, then $\tilde{x}$ is the exact solution of $Ax = b$. Thus the iterative refinement method can be written as follows.

1. Calculate $A\tilde{x} - b$ with higher precision. ($\tilde{r} \leftarrow A\tilde{x} - b$)
2. Solve $Ap = \tilde{r}$ using R. ($\tilde{p} \leftarrow R \cdot \tilde{r}$)
3. Update $\tilde{x}$. ($\tilde{x}_{new} \leftarrow \tilde{x} - \tilde{p}$)

We combine this iterative refinement method with the proposed verification method, i.e. we propose a method which calculates $\tilde{x} \in \mathbb{F}^n$ and $y \in \mathbb{F}^n$ satisfying

$$\left| \frac{\tilde{x}_i - x_i^*}{\tilde{x}_i} \right| \leq tol, \quad (\tilde{x}_i \neq 0) \tag{9}$$

with a desired tolerance *tol* and

$$|\tilde{x} - x^*| \leq y. \tag{10}$$

Here, we list characteristics of the proposed method.

- If we do not know in advance which precision is sufficient to calculate an accurate inverse R, the method calculates R adaptively by iterative calculations.
- Higher precision arithmetic is necessary in only matrix product and matrix-vector product.
- Therefore a fast algorithm can be implemented in terms of measured computing time.

We will confirm the usefulness of the proposed method by numerical experiments in the next section.

4. Numerical Experiments

In this section we present some results of numerical examples to illustrate the performance of the proposed method. We used a PC with Pentium 4 2.53GHz CPU. All calculations were done on MATLAB 7 with IEEE 754 double precision.

4.1. Hilbert Matrix

First we consider the Hilbert matrix as an example of a linear system whose condition number is larger than 10^{16}. The (i, j)-element of the $n \times n$ Hilbert matrix $H = (h_{ij})$ is

$$h_{ij} = \frac{1}{i + j - 1}. \tag{11}$$

To avoid the rounding errors in constructing a coefficient matrix, we set $A := s \cdot H$ where s is a common multiplier from 2 to $2n - 1$. This technique is valid for at least $n \leq 20$. Let $n = 20$ and $b := A \cdot z$ where $z_i = (-1)^i$ for $1 \leq i \leq 20$. Since calculation of b causes no rounding error, the exact solution x^* of $Ax = b$ becomes exactly z. We set $n = 20$ and $tol = 10^{-9}$. When $n = 20$, we find $\kappa(A) = 2.45 \cdots \times 10^{28}$. The result is as follows:

```
>> n=20; [A,b,cnd]=myhilb(n); cnd
cnd =
    2.452156585815303e+028
>> [x,y]=AccVerLin(A,b,1e-9); [x,y]
*** calculation of approximate inverse ***
k =   2
*** verification for a linear system ***
***    with iterative refinement     ***
loop =   1, max. rel. error = 1.095537e-004
loop =   2, max. rel. error = 8.874301e-010
ans =
   -1.000000000000000e+000    8.617077e-025
    1.000000000000000e+000    9.640417e-024
   -1.000000000000000e+000    1.062221e-021
    1.000000000000000e+000    4.941914e-020
   -9.999999999999992e-001    7.780127e-016
    9.999999999999695e-001    3.054625e-014
   -9.999999999994678e-001    5.324606e-013
    9.999999999996297e-001    3.709047e-013
   -9.999999999960786e-001    3.925436e-012
    9.999999999602059e-001    3.980825e-011
   -1.000000000120782e+000    1.208331e-010
    1.000000000595174e+000    5.953339e-010
   -9.999999998351981e-001    1.649530e-010
    9.999999994154054e-001    5.849744e-010
   -1.000000000637741e+000    6.379809e-010
    9.999999991127588e-001    8.874301e-010
   -9.999999999849613e-001    1.513451e-011
    9.999999999542837e-001    4.574802e-011
   -9.999999999890732e-001    1.093710e-011
    9.999999999997991e-001    2.018749e-013
```

The first column of `ans` displays a computed solution $\tilde{x}$, and the second column the right-hand side of inequality (10), i.e. an error bound y of $\tilde{x}$. Therefore it means $\tilde{x} - y \leq x^* \leq \tilde{x} + y$. When `loop` = 1, we did not apply the iterative refinement method. After `loop` = 2, we apply (`loop` − 1) times iterative refinements. Moreover, `max. rel. error` means the maximum relative error of $\tilde{x}$, an upper bound of $\max_{1 \leq i \leq n} |\tilde{x}_i - x_i^*| / |\tilde{x}_i|$.

Next we change b and *tol*, i.e. set $b = (1, 1, \cdots, 1)^T \in \mathbb{R}^{20}$ and $tol = 10^{-12}$. Then the result is as follows:

```
>> b=ones(n,1);
>> [x,y]=AccVerLin(A,b,1e-12); [x,y]
*** calculation of approximate inverse ***
k =   2
*** verification for a linear system ***
***    with iterative refinement     ***
loop =   1, max. rel. error = 9.044724e-004
loop =   2, max. rel. error = 3.366434e-009
loop =   3, max. rel. error = 1.366729e-014
ans =
   -3.743263442685729e-015    5.116025e-029
                    ⋮
    2.579979360165119e-004    2.070311e-020
```

4.2. Larger Condition Number Matrix

To construct an example of a floating-point $n \times n$ matrix A with larger condition number, we use Rump's method [7] of generating arbitrarily ill-conditioned matrices. We set a right-hand side vector as $b := (1, 1, \ldots, 1)^T \in \mathbb{R}^n$. We also set $n = 100$, $\kappa(A) \approx 10^{100}$ and $tol = 10^{-12}$. Then the result of the proposed verification method is as follows:

```
>> n=100; A=randmat(n,1e+100); b=ones(n,1);
>> [x,y]=AccVerLin(A,b,1e-12);
*** calculation of approximate inverse ***
k = 8
*** verification for a linear system ***
***    with iterative refinement     ***
loop = 1, max. rel. error = 6.93e-06
loop = 2, max. rel. error = 4.79e-11
loop = 3, max. rel. error = 4.27e-16
elapsed time is 8.109000 sec.
```

Finally we consider a larger size problem and change conditions, i.e. we set $n = 500$, $\kappa(A) \approx 10^{50}$ and $tol = 10^{-12}$. Then the result is as follows:

```
>> n=500; A=randmat(n,1e+50); b=ones(n,1);
>> tic; [x,y]=AccVerLin(A,b,1e-12); toc
*** calculation of approximate inverse ***
k =   5
*** verification for a linear system ***
***    with iterative refinement     ***
loop =   1, max. rel. error = 3.776758e-009
loop =   2, max. rel. error = 1.023496e-016
elapsed time is 190.500000 sec.
```

From these results, it turns out that we can verify the nonsingularity of A and calculate a verified and accurate approximate solution $\tilde{x}$ of a linear system $Ax = b$ whose condition number is extremely large by use of the proposed method and the iterative refinement method.

Acknowledgments

This research was partially supported by CREST program, Japan Science and Technology Agency (JST), 21st Century COE Program (Productive ICT Academia Program, Waseda University) from the Ministry of Education, Science, Sports and Culture of Japan and Individual Research of Waseda University Grant for Special Research Projects (No. 2005A-884).

References

[1] U. W. Kulisch, W. L. Miranker, "The arithmetic of the digital computer: A new approach," *SIAM Review*, 28 (1986), 1–40.

[2] N. J. Higham, "Accuracy and Stability of Numerical Algorithms," 2nd ed., *SIAM Publications*, Philadelphia, PA, 2002.

[3] X. Li, et al., "Design, implementation and testing of extended and mixed precision BLAS," *ACM Trans. Math. Softw.*, 28 (2002), 152–205.

[4] T. Ogita, S. M. Rump, S. Oishi, "Accurate sum and dot product," *SIAM J. Sci. Comput.*, 26:6 (2005), 1955–1988.

[5] S. Oishi, S. M. Rump, "Fast verification of solutions of matrix equations," *Numer. Math.* 90:4 (2002), 755–773.

[6] D. M. Priest, "On Properties of Floating Point Arithmetics: Numerical Stability and the Cost of Accurate Computations," *PhD thesis*, University of California at Berkeley, 1992.

[7] S. M. Rump, "A class of arbitrarily ill-conditioned floating-point matrices," *SIAM J. Matrix Anal. Appl.*, 12:4 (1991), 645–653.

[8] S. M. Rump, "Approximate inverses of almost singular matrices still contain useful information, Forschungsschwerpunktes Informations- und Kommunikationstechnik," Technical Report 90.1, Hamburg University of Technology, 1990.

[9] T. Yamamoto, "Error bounds for approximate solutions of systems of equations," *Japan J. Appl. Math.*, 1:1 (1984), 157–171.