Verified Solutions of Sparse Linear Systems by LU factorization

TAKESHI OGITA
CREST, Japan Science and Technology Agency (JST), and Faculty of Science and Engineering, Waseda University, Robert J. Shillman Hall 802, 3-4-19 Okubo, Shinjuku-ku, Tokyo 169-0072, Japan, e-mail: ogita@waseda.jp

SIEGFRIED M. RUMP
Institut für Informatik III, Technische Universität Hamburg-Harburg, Schwarzenbergstraße 95, Hamburg 21071, Germany, e-mail: rump@tu-harburg.de

SHIN'ICHI OISHI
Faculty of Science and Engineering, Waseda University, Robert J. Shillman Hall 802, 3-4-19 Okubo, Shinjuku-ku, Tokyo 169-0072, Japan, e-mail: oishi@waseda.jp

Abstract. A simple method of calculating an error bound of computed solutions of general sparse linear systems is proposed. It is well known that the verification for sparse linear systems is still difficult except for the case where it is known in advance that the coefficient matrix has special structures such as M-matrix. The new verification algorithm is based on direct methods such as LU factorization. Results of numerical experiments are presented for illustrating that computational cost of calculating an error bound of an obtained computed solution is acceptable in practice.

Keywords: self-validating method, verified computation, sparse linear systems

1. Introduction

In this paper, we are concerned with the accuracy of a computed solution of a linear system

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

where A is a real $n \times n$ matrix and b is a real n-vector. Our goal is to verify the nonsingularity of A and to estimate an error bound ϵ of a computed solution $\widetilde{x}$ of (1) for the exact solution $x^* = A^{-1}b$ such that

$$\|\widetilde{x} - x^*\|_\infty \le \epsilon. \tag{2}$$

Recently, fast verification methods (cf., for example, [9, 12]) have been developed to calculate rigorous and tight bounds for (2) on computers abiding by IEEE standard 754 for floating point arithmetic.

However, it is well known that the verification for *sparse* linear systems is still difficult except for the case where we know *in advance* that the coefficient matrix A belongs to a certain special matrix class, e.g., diagonally dominant matrix, M-matrix, H-matrix and totally nonnegative matrix. The reason of difficulty for sparse systems is mainly due to the destruction of its sparsity which occurs in the verification process. Thus the verification for sparse linear systems with interval coefficients becomes one of the open problems in *Grand Challenges and Scientific Standards in Interval Analysis* [8] presented by Neumaier. In this paper, we consider sparse systems including banded systems.

In the present state, fast verification algorithms for sparse system with

- monotone matrix [10] including M-matrix,
- H-matrix (deduced from the case of M-matrix),
- symmetric positive definite matrix [9, 15, 17],
- symmetric matrix [9, 16] and
- general matrix [15]

have already been known. Here, the first and the second cases can be treated with a favorable iterative method such as Gauss-Seidel, SOR or conjugate gradient method, and the others require the direct method such as LU or Cholesky factorization. The conventional verification algorithms for sparse matrices via direct methods are based on the estimation of the smallest singular value of A. For example, when A is a symmetric positive definite matrix, the conventional algorithms [9, 15] use Cholesky or $\mathrm{LDL}^{\mathrm{T}}$ factorization of a shifted matrix $A - \alpha I$ with positive constant α. This approach needs some inverse iterations for estimating a suitable α. Moreover, if A is a general matrix, the algorithm becomes more complicated and needs more computational resources.

The purpose of this paper is to develop a new verification algorithm for a general sparse matrix using LU factorization as one of the direct methods. In verification process of the proposed method, it is not necessary to bound the smallest singular value of A. Only the method requires is an LU factorization of A, which can be obtained to calculate a computed solution of (1). The proposed method is very simple, so that it is easy to understand and use it. We think this is very important point for practical use and implementations, especially in case of treating sparse matrices because it is not necessary for users to implement a lot of additional routines, i.e., only we need is the existing direct solver for $U^T x = b$ (hopefully sparse right-hand side is possible) and $L^T x = b$. A main goal of the article is to show that it is possible to compute tight upper bounds for (2).

We shall present results of numerical experiments which document that the computational cost of calculating an error bound of an obtained computed solution is acceptable in practical use.

2. Verification theory

Let $A = (a_{ij})$ be a real $n \times n$ matrix and $Y = (y_{ij})$ an approximate inverse of A. Let also b be a real n-vector and $\widetilde{x}$ an approximate solution of $Ax = b$. It is well known that if it holds that

$$\|YA - I\| < 1, \tag{3}$$

where I stands the $n \times n$ identity matrix, then A is nonsingular,

$$\left\|A^{-1}\right\| \leq \frac{\|Y\|}{1 - \|YA - I\|}, \tag{4}$$

and

$$\left\|\widetilde{x} - A^{-1}b\right\| \leq \frac{\|Y(A\widetilde{x} - b)\|}{1 - \|YA - I\|}. \tag{5}$$

Based on this, we present a theorem to verify the nonsingularity of A and to bound the maximum norm of its inverse, $\left\|A^{-1}\right\|_\infty$, and the error bound of the approximate solution, $\left\|\widetilde{x} - A^{-1}b\right\|_\infty$.

THEOREM 1. *Let A be a real $n \times n$ matrix and b a real n-vector. Let also $\widetilde{x}$ be an approximate solution of $Ax = b$. Let further $e^{(j)} = (e_1^{(j)}, \ldots, e_n^{(j)})^T$ be a unit n-vector corresponding to the j-th column of the $n \times n$ identity matrix $I = (e_{ij})$, i.e.,*

$$e_i^{(j)} := \begin{cases} 1 & (i = j) \\ 0 & (i \neq j) \end{cases},$$

and $y^{(j)} = (y_1^{(j)}, \ldots, y_n^{(j)})^T$ an n-vector corresponding to the transpose of the j-th row of an approximate inverse $Y = (y_{ij})$ of A, i.e., $y_i^{(j)} := y_{ji}$. If α satisfies

$$\max_{1 \leq j \leq n} \left\|A^T y^{(j)} - e^{(j)}\right\|_1 \leq \alpha < 1, \tag{6}$$

then A is nonsingular,

$$\left\|A^{-1}\right\|_\infty \leq \frac{\max_{1 \leq j \leq n} \left\|y^{(j)}\right\|_1}{1 - \alpha} \tag{7}$$

and

$$\left\|\widetilde{x} - A^{-1}b\right\|_\infty \leq \frac{\max_{1 \leq j \leq n} \left|(A\widetilde{x} - b)^T y^{(j)}\right|}{1 - \alpha}. \tag{8}$$

Proof. It follows by $y_{ij} = y_j^{(i)}$ and $e_{ij} = e_i^{(j)} = e_j^{(i)}$ that

$$\|Y\|_\infty = \max_{1\le i\le n} \sum_{j=1}^{n} |y_{ij}| = \max_{1\le i\le n} \sum_{j=1}^{n} \left|y_j^{(i)}\right| = \max_{1\le i\le n} \left\|y^{(i)}\right\|_1 \tag{9}$$

and

$$\begin{aligned}\|YA - I\|_\infty &= \max_{1\le i\le n} \sum_{j=1}^{n} \left|\sum_{k=1}^{n} y_{ik}a_{kj} - e_{ij}\right| = \max_{1\le i\le n} \sum_{j=1}^{n} \left|\sum_{k=1}^{n} y_k^{(i)} a_{kj} - e_j^{(i)}\right| \\ &= \max_{1\le i\le n} \sum_{j=1}^{n} \left|\left(A^T y^{(i)} - e^{(i)}\right)_j\right| \\ &= \max_{1\le i\le n} \left\|A^T y^{(i)} - e^{(i)}\right\|_1 . \end{aligned} \tag{10}$$

Combining (3), (4), (9) and (10) proves (7). Moreover,

$$\|Y(A\tilde{x} - b)\|_\infty = \max_{1\le j\le n} \left|(A\tilde{x} - b)^T y^{(j)}\right|$$

and (5) proves (8). □

3. Verification using LU factorization

Suppose L, U and P are given by LU factorization (with partial pivoting) of A in floating-point arithmetic such that $PA \approx LU$. Consider the following matrix equation

$$YA = I$$

for Y. This is equivalent to

$$A^T y^{(j)} = e^{(j)} \quad \text{for } j = 1, \dots, n.$$

Therefore, if L and U are the exact LU factors of A, then

$$(P^T LU)^T y^{(j)} = e^{(j)}$$

and

$$y^{(j)} = P^T L^{-T} U^{-T} e^{(j)}.$$

We now present an algorithm of calculating an error bound on $\|\tilde{x} - A^{-1}b\|_\infty$ based on the fast verification algorithm proposed by Oishi and Rump [12, 13].

ALGORITHM 1. *Calculation of an error bound on* $\|\tilde{x} - A^{-1}b\|_\infty$*:*

function $[\widetilde{x}, \mathrm{err}] = \mathrm{vlin\_lu}(A, b)$
$[L, U, P] = \mathrm{lu}(A)$; % *LU factorization:* $PA \approx LU$
$\widetilde{x} = \mathrm{fl}\,(U \backslash (L \backslash (Pb)))\,;$
setround(-1);
$\underline{r} = \mathrm{fl}\,(A\widetilde{x} - b)\,;$
setround$(+1)$;
$\overline{r} = \mathrm{fl}\,(A\widetilde{x} - b)\,;$
$r_{\mathrm{mid}} = \mathrm{fl}\,((\underline{r} + \overline{r})/2)\,; \quad r_{\mathrm{rad}} = \mathrm{fl}\,(r_{\mathrm{mid}} - \underline{r})\,;$
$\alpha = 0; \quad \beta = 0;$
for $j = 1 : n$
  setround(0);
  $t = \mathrm{fl}\left(U^T \backslash e^{(j)}\right);$ % *Solve* $U^T t = e^{(j)}$ *for* t
  $y = \mathrm{fl}\left(P^T (L^T \backslash t)\right);$ % *Solve* $L^T P^T y = t$ *for* y
  setround(-1);
  $\underline{t} = \mathrm{fl}\left(A^T y - e^{(j)}\right); \quad \underline{\phi} = \mathrm{fl}\left(y^T r_{\mathrm{mid}}\right);$
  setround$(+1)$;
  $\overline{t} = \mathrm{fl}\left(A^T y - e^{(j)}\right); \quad \overline{\phi} = \mathrm{fl}\left(y^T r_{\mathrm{mid}}\right);$
  $\overline{t} = \max(|\underline{t}|, |\overline{t}|);$
  $\overline{\phi} = \mathrm{fl}\left(\max(\left|\underline{\phi}\right|, \left|\overline{\phi}\right|) + |y|^T r_{\mathrm{rad}}\right);$
  $\alpha = \max(\alpha, \mathrm{fl}\,(\|\overline{t}\|_1)); \quad \beta = \max(\beta, \overline{\phi});$
  if $\alpha \geq 1$
    error(`'verification failed.'`)
  end
end
$\mathrm{err} = \mathrm{fl}\,(\beta / -(\alpha - 1))\,;$ % $\mathrm{fl}_{\triangle}\,(\beta/-(\alpha-1)) \geq \beta/(1-\alpha)$

Note that the verification method can be used with the ordering strategies for sparse matrix, e.g., the (approximate) minimum degree permutation and the reverse Cuthill-McKee ordering (cf., for example, [4, 5]), i.e., our algorithm does not depend on the process of obtaining the LU factors. Even if one uses not only a row permutation P but also a column permutation Q in the LU factorization, it is easy to modify the algorithm by considering $PAQ \approx LU$ instead of $PA \approx LU$. In addition, the algorithm can be done by blockwise for j with adapting the block size to the computer environment. Of course, this requires more memory space, but may achieve less computing time in practical use.

4. Numerical examples

To illustrate that the proposed verification method gives a tight error bound of a computed solution of a linear system $Ax = b$, we shall report results of numerical experiments. We have used a PC with Intel Pentium IV 3.46GHz CPU and Matlab 7.0.4 [18]. This computer environment satisfies the IEEE 754 standard. To solve sparse linear systems $Ax = b$ by LU factorization, Matlab uses UMFPACK [2]. Moreover, we set block size 100 for Algorithm 1.

The coefficient matrices used in the numerical experiments are taken from the Harwell-Boeing collection [3]. Although these matrices may be symmetric and even positive definite, we treat such matrices as general matrices because we purely want to evaluate the performance of the proposed method. We put the right-hand side vector $b = (1, \ldots, 1)^T$ if b is not provided by the example.

Next, we prepared some coefficient matrices and right-hand side vectors (NW$**$)[1] whose origins are the problems computing an approximate minimum eigenpair of Orr-Sommerfield equations for Poiseuille flow by Newton-Raphson iteration with piecewise cubic Hermite base function. These matrices are unsymmetric.

In Tables I and II, the results of the numerical experiments are displayed. Here, nnz(A) for a sparse matrix A means the number of nonzero elements of A. In the column labeled by α, an upper bound on $\|RA - I\|_\infty$ is given. The quantity ϵ refers to the maximum error bound of the computed solution of $Ax = b$, i.e. $\|\widetilde{x} - A^{-1}b\|_\infty \le \epsilon$. In the column labeled by t, elapsed time (sec.) for the verification is given. The notation n/a in the table means that the data are not available because the verification of the computed solutions failed ($\alpha \ge 1$). We omit the results of problems when the problem size is small (roughly less than 1,000) or the problem is similar to the one listed in Table I. We also omit the case where an approximate solution cannot be obtained by UMFPACK on Matlab, which usually means the problem is extremely ill-conditioned. These results document that if we can obtain computed solutions, then we can also get their verified error bounds in almost all cases.

In conclusion, it turns out that verified error bounds of approximate solutions of sparse linear systems by the proposed verification method. It seems that the verification method is useful in case where it is not known that the coefficient matrix of a linear system has structures such as M-matrix or symmetric positive definite. Therefore, we can construct the hybrid verification method for linear system; for small dimensions

[1] Thanks to Profs. M. T. Nakao and Y. Watanabe for providing the test matrices.

Table I. Verification results for sparse linear systems with Harwell-Boeing collection. The notation n/a means that the data are not available.

Problem	n	nnz(A)	nnz($L+U$)	α	ϵ	t (sec.)
1138_BUS	1,138	4,054	5,392	6.9e-11	7.0e-11	2.51
BCSSTK09	1,083	18,437	115,447	6.7e-12	6.1e-12	3.56
BCSSTK13	2,003	83,883	529,869	6.6e-10	6.7e-10	20.8
BCSSTK15	3,948	117,816	1,225,222	2.0e-10	1.5e-10	86.4
BCSSTK17	10,974	428,650	2,076,228	2.1e-09	1.5e-09	515.
BCSSTK25	15,439	252,241	2,847,888	2.0e-08	2.1e-08	987.
BLCKHOLE	2,132	14,872	189,045	5.6e-10	3.7e-12	13.7
GEMAT11	4,929	33,108	60,387	4.2e-10	3.5e-10	52.6
GEMAT12	4,929	33,044	61,473	9.2e-10	4.3e-10	51.9
LNSP3937	3,937	25,407	277,332	1.9e-02	9.7e-10	41.2
LSHP1009	1,009	6,865	74,881	1.4e-10	1.4e-12	2.98
LSHP2233	2,233	15,337	210,778	3.7e-10	5.2e-12	16.2
LSHP3466	3,466	23,896	420,469	3.7e-09	2.6e-11	45.3
MAHINDAS	1,258	7,682	14,736	1.2e-08	3.0e-09	1.29
NNC1374	1,374	8,588	49,954	1.5e-01	9.1e-02	4.21
ORANI678	2,529	90,158	111,060	2.6e-12	8.7e-13	12.0
ORSREG_1	2,205	14,133	154,159	1.3e-12	1.2e-12	12.8
PLAT1919	1,919	32,399	132,605	8.6e+01	n/a	10.3
PSMIGR_1	3,140	543,160	5,821,707	4.5e-09	4.5e-09	244.
PSMIGR_2	3,140	540,022	6,714,444	5.3e-09	1.5e-09	271.
PSMIGR_3	3,140	543,160	5,821,707	8.0e-13	3.5e-12	244.
SAYLR4	3,564	22,316	294,830	1.1e-09	8.4e-10	37.6
SHERMAN1	1,000	3,750	18,180	1.2e-12	1.8e-12	1.03
SHERMAN3	5,005	20,033	187,091	9.5e-12	6.9e-12	22.7
SHERMAN5	3,312	20,793	126,962	4.8e-13	4.1e-13	8.23
WATT_1	1,856	11,360	99,762	3.4e-13	3.3e-13	8.02
WATT_2	1,856	11,550	105,589	1.3e-12	1.3e-12	8.23
WEST1505	1,505	5,414	8,262	2.0e-09	2.5e-09	2.80
WEST2021	2,021	7,310	10,879	2.4e-09	2.9e-09	5.17

the method for dense matrix is available. When the coefficient is symmetric matrix, then we can try the super-fast verification method for positive definite matrix in [17]. If it can be proved that the coefficient is H-matrix, e.g. using iterative criterion [6], then the fast verification method [10] can be utilized. Otherwise, the proposed method in this paper may become a fallback algorithm.

Table II. Verification results for sparse linear systems from Orr-Sommerfield equations for Poiseuille flow. The notation n/a means that the data are not available.

Problem	n	nnz(A)	nnz($L+U$)	α	ϵ	t (sec.)
NW398	398	5,508	7,845	5.4e-09	5.0e-09	0.35
NW1998	1,998	27,908	31,901	9.4e-06	9.4e-06	10.2
NW3998	3,998	55,908	63,901	1.6e-04	1.6e-04	44.2
NW7998	7,998	111,908	128,134	3.1e-03	3.4e-03	186.
NW19998	19,998	279,908	324,113	5.2e-01	1.1e+00	1,179.
NW29998	29,998	419,908	544,197	1.34+00	n/a	2,668.

References

1. Alefeld, G., Herzberger, J.: *Introduction to Interval Computations*, Academic Press, New York, 1983.
2. Davis, T. A.: Algorithm 832: UMFPACK, an unsymmetric-pattern multifrontal method, *ACM Trans. Math. Softw.*, 30:2 (2004), 196–199.
3. Duff, I. S., Grimes, R. G., and Lewis, J. G.: User's guide for Harwell-Boeing sparse matrix test problems collection, Technical Report RAL-92-086, Computing and Information Systems Department, Rutherford Appleton Laboratory, Didcot, UK, 1992.
4. George, A., Liu, J.: The evolution of the minimum degree ordering algorithm, *SIAM Rev.*, 31 (1989), 1–19.
5. Gilbert, J. R., Moler, C. and Schreiber, R.: Sparse matrices in MATLAB: design and implementation, *SIAM J. Matrix Anal. Appl.*, 13 (1992), 333–356.
6. Li, L.: On the iterative criterion for generalized diagonally dominant matrices, *SIAM J. Matrix Anal. Appl.*, 24:1 (2002), 17–24.
7. Li, X. S.: An overview of SuperLU: Algorithms, implementation, and user interface, *ACM Trans. Math. Softw.*, 31:3 (2005), to appear.
8. Neumaier, A.: Grand challenges and scientific standards in interval analysis, *Reliable Computing*, 8 (2002), 313–320.
9. Ogita, T., Oishi, S. and Ushiro, Y.: Computation of sharp rigorous componentwise error bounds for the approximate solutions of systems of linear equations, *Reliable Computing*, 9:3 (2003), 229–239.
10. Ogita, T., Oishi, S., Ushiro, Y.: Fast verification of solutions for sparse monotone matrix equations, *Topics in Numerical Analysis (Computing, Supplement 15, G. Alefeld and X. Chen eds.)*, 175–187, Springer WienNewYork, Austria, 2001.
11. Oishi, S.: Fast enclosure of matrix eigenvalues and singular values via rounding mode controlled computation, *Linear Alg. Appl.*, 324 (2001), 133–146.
12. Oishi, S., Rump, S. M.: Fast verification of solutions of matrix equations, *Numer. Math.*, 90:4 (2002), 755–773.
13. Rump, S. M.: Fast and parallel interval arithmetic, *BIT*, 39:3 (1999), 539–560.

14. Rump, S. M.: INTLAB – INTerval LABoratory, *Developments in Reliable Computing (T. Csendes, ed.)*, 77–104, Kluwer Academic Publishers, Dordrecht, 1999. http://www.ti3.tu-harburg.de/~rump/intlab/
15. Rump, S. M.: Verification methods for dense and sparse systems of equations, *Topics in Validated Computations (J. Herzberger, ed.)*, Elsevier, Amsterdam, 1994, 63–135.
16. Rump, S. M.: Verified computation of the solution of large sparse systems, *ZAMM*, 75 (1995), S439–S442.
17. Rump, S. M., Ogita, T.: Super-fast validated solution of linear systems, *J. Comp. Appl. Math.*, to appear.
18. The MathWorks Inc.: MATLAB Users Guide, Version 7, 2004.