# New Arithmetic Filter for the Sign of the Determinant of a Matrix *†

Victor Y. Pan
Department of Mathematics and Computer Science
Lehman College, CUNY Bronx, NY 10468
VPAN@ALPHA.LEHMAN.CUNY.EDU

June 9, 1999

**Abstract**

Certified sign of the determinant of a matrix is very frequently required in various geometric and algebraic computations. The computations go faster when they are performed numerically, with floating point and rounding to a single precision. In this case, however, the certification by the known methods is practically difficult because the magnitude of the determinant of an integer input matrix $A$ may vary dramatically, from 1 to $||A||^n$, and the roundoff error bound of the determinant computation varies proportionally. Because of such a variation, slower computation of det $A$ with a high precision is frequently required to ensure that the output error bound is smaller than the magnitude of the determinant, and by experiments, the single precision computations tend to produce correct answer even where the above certification fails. We observe, however, that our certification task of determining only a single bit of det $A$, that is, the bit carrying the sign, does not require to estimate the latter roundoff error. Instead, we solve a much simpler task of computing numerically the factorization of a matrix by Gaussian elimination with pivoting (which is a subroutine of LAPACK and LINPACK) and of estimating the minimum distance $1/||A^{-1}||$ from $A$ to a singular matrix. Such an estimate gives us a desired range for the roundoff error of the factorization such that the invariance of the sign of det $A$ is ensured as long as the error varies in this range. Based on such observations, we improve substatially the known arithmetic filters for the certification of the sign.

**Key words:** the sign of matrix determinant, arithmetic filters, certified geometric computations, convex hull, distance to a singularity.

## 1 Introduction. The Problem of the Certification of the Sign of Matrix Determinant

Computation of the sign of the determinant of a $k \times k$ matrix, $A$, is a major task in practive of geometric and algebraic computations. Its solution is required for such fundamental problems as the computation of the convex hulls and Voronoi diagrams, of the arrangement of lines and line segments, and of the orientation of a polyhedron or an algebraic variety. In the case of computing the orientation for convex optimization problems in statistical physics and chemistry, $k$ is frequently large, ranging from 100 to 500, and the entries of $A$ are short integers (represented, say, with 5 to 10 bits). For the other cited tasks of practical

*Supported by NSF Grants CCR 9625344 and CCR 9732206 and PSC CUNY Awards 668365 and 669363.

†The results of this paper have been presented at the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA99), Maryland, January 1999.

geometric computations, $k$ is not greater than 12 and typically is below 5, but then the input values are given with a high precision (by "long" binary numbers).

In many cases, it is crucially important to have the *certified sign*, and this prompted intensive work of several researchers on the design of sophisticated algorithms that turn the entries of the input matrix $A$ into integers (via chopping and scaling them) and then compute det $A$ symbolically, with no errors [Wi], [KP91], [KP92], [C92], [BP94], [K95], [ABDPY97], [BEPP97], [BEPP99].

On the other hand, according to the current state of the art, the IEEE 754 standard floating point computations [IEEE87] clearly outperform symbolic techniques applied to yield computations with no errors. The certification of the result of numerical computation of sign (det $A$), however, is a rather hard problem. For example, let $A = \begin{pmatrix} 10^h & 10^h - 1 \\ 10^h + 1 & 10^h \end{pmatrix}$. In this case, det $A = 10^h - (10^h - 1)(10^h + 1) = 1$, that is , already for a moderately large $h$ (say, $h = 10$) the representation of the intermediate term $-(10^h - 1)(10^h + 1)$ requires more than 53 bits allowed by the single precision of the IEEE standards, even though the integer output value of det $A$ is small.

Instead of the above direct computation of det $A$, one may rely on the auxiliary factorization of $A$ into the product of two triangular matrices, obtained by Gaussian elimination. The overflow problem can be avoided by means of pivoting, but the estimated roundoff errors would still be of order $10^{2h}u$, $u$ being the *unit roundoff* (also called *machine epsilon* and equal to the maximum relative error of a roundoff of a real number for a fixed computer precision); already for a moderately large $h$ the value $10^{2h}u$ may exceed det $A = 1$, and then sign (det $A$) remains uncertified. Generally, $|\det A|$ can be as large as $||A||^k$, and the estimated roundoff error, $e_d$, tends to be of order $||A||^k u$. For a large class of input matrices, $|\det A|$ exceeds such an estimate, and then sign(det $A$) is certified. Otherwise, $e_d \geq |\det A|$, and then the shift to higher precision numerical computation or to the more expensive symbolic computation of det $A$ (with no error) is in order.

Thus the floating point computation of det $A$ with rounding to a single precision acts as an *arithmetic filter* that solves the problem for a large class of input matrices. For other input matrices, the filter leaves the solution to slower computations with higher (or infinite) precision. Intensive research effort was spent in the recent years with the goal to devise effective arithmetic filters that would filter out a larger class of input instances for which already the floating point computations with single precision produce the certified sign. Then more expensive symbolic techniques would be invoked only for a smaller class of unfiltered input instances. Such a topic received much attention in the literature, regarding the computation of both sign of $\det A$ and some other algebraic and geometric predicates [BKMNSU95], [YD95], [FvW96], [ABDPY97], [PYS97].

In the case of det $A$, however, the output of the roundoff errors tend to be greatly overestimated, which frequently does not allow us to certify the sign even where it is computed correctly. Such a phenomenon can be well understood in the light of the above consideration and was confirmed by extensive experiments (cf. e.g. [PYS97]).

**Acknowledgement.** Comments of the referee helped me to clarify the results and to make the paper more accessible to the readers.

## 2 A New Arithmetic Filter for the Certifications of the Sign of the Determinant

Our main result is a new arithmetic filter that greatly improves the certification of sign(det $A$). Our point of departure is the observation that we only need to certify a single bit, that is, sign(det $A$), and this turns out to be a simpler task than the certified computation of several leading bits of det $A$, which we avoid. Let us describe our algorithm in terms of the standard definitions in the area of matrix computations (cf. [GL96]), that is, for a matrix $W = (w_{i,j})$, the matrix of the absolute values $|w_{i,j}|$ is denoted by $|W|$; $||\cdot||$ is any fixed operator matrix norm, e.g. $||W|| = ||W||_1 = \max_j \sum_i |w_{i,j}|$ or $||W|| = ||W||_\infty = \max_i \sum_j |w_{i,j}|$; $u$ is the unit roundoff (machine epsilon). We will assume that a matrix norm $||\cdot||$, the unit roundoff $u$, and the value $\gamma_k = ku/(1-ku)$ are fixed (cf. [H96]).

Our arithmetic filter consists of the following steps:

1. Apply Gaussian elimination with partial pivoting (that is, with row permutations) to obtain a permutation matrix $P$ and to compute numerically two matrices, $L$ and $U$ (where $L$ is a unit lower triangular matrix, and $U$ is an upper triangular matrix), such that $A = PLU$, det $L = 1$, det $A = (\det P)\det U$. As soon as this is done, det $P$ and numerical approximations to det $U$ and, therefore, det $A$ are readily available. The computed approximations to $L, U$ and $\tilde{d} = \det A$ will be denoted by $\tilde{U}, \tilde{L}$ and $\tilde{d}$ , respectively, and we will write $e_d = |\tilde{d} - \det A|$.

2. Apply some known simple arithmetic filters to compute an upper estimat $e^+$ on the roundoff errore $e_d$ of computing det $A$ (cf. e.g. [PYS97], [PY98], [PY99]). If $e^+ < |\tilde{d}|$, then output the sign of det $A$ given by

$$\text{sign } \tilde{d} = \text{sign}(\det A). \tag{2.1}$$

3. Otherwise compute

$$e = ||(|\tilde{L}| \cdot |\tilde{U}|)||\gamma_k \tag{2.2}$$

and $N = \|\tilde{U}^{-1}\tilde{L}^{-1}\|$. If $eN < 1$, output sign($\tilde{d}$)=sign(det $A$). Otherwise output $|\tilde{d}| + e^+$ and FAILURE.

In principle, one may skip stage 2, because stage 3 alone would suffice as a filter, but the computations of stage 2 are simpler and still filter out a large class of input matrices $A$ for which floating point numerical computation of det $A$ with single precision produces the certified sign. Note that stage 3 avoids computing det $A$ and proceeds without estimating the roundoff error of its computation but still certifies sign (det $A$) for a class of matrices $A$ that is sbustatially large than the one filtered out at stage 2. In the rare cases where FAILURE is output the algorithm also produces an upper bound $|\tilde{d}| + e$ and $\det A$, which can be useful in the case of subsequent application of the algorithms of [BEPP97], [BEPP99]. The resulting algorithms amount to invocation of some simple and widely available subroutines of LAPACK, LINPACK and MATLAB, which can be effectively parallelized. The reader is referred to [PY98], [PY99] on various details, ameliorations and the results of some numerical tests.

## 3 Proof of the Correctness of the Algorithm.

The proof of the correctness of the algorithm relies on the following observations:

a) The classical techniques of the backward error analysis immediately gives us the following fundamental result (see. e.g. [H96], theorem 9.3 page 175):

*$||E|| \leq e$ for $e$ of (2.2) and for the matrix $E = A - P\tilde{L}\tilde{U}$, that is, the $P\tilde{L}\tilde{U}$ factorization of $A$ computed with a machine precision $u$ gives us exact factorization of a matrix $A - E$ with $E$ lying in the $e$-neighborhood of $A$.*

b) If some continuous perturbation of $A$ within its $e$-neighborhood changes the sign of det $A$, then, by continuity, such a perturbation must pass through a singular matrix (whose determinant vanishes); consequently, (2.1) holds if $e$ is less than the distance $\delta$ from $\tilde{A} = P\tilde{L}\tilde{U}$ to a closest singular matrix.

c) Such a distance is readily computable because $1/\delta = N = \|\tilde{A}^{-1}\| = \|\tilde{U}^{-1}\tilde{L}^{-1}\|$ by a well-known theorem [EY39], [K66].

## 4 Further Comments on the Performance and the Implementation of the Algorithm

To compute numerically the $PLU$ factorization of $A$ and the value $N$, it suffices to invoke the standard subroutines of LAPACK, LINPACK and MATLAB (that is, ones for Gaussian elimination with partial pivoting and for solving $2k$ triangular linear systems of equations, respectively). For the experiments reported in [PY98], [PY99], the codes for the computation of the $PLU$ factors and of the norm $N$ were created directly from the algorithms description in [GL96].

The subsequent computation of $e$ is even simpler than that because $||(|\tilde{L}| \cdot |\tilde{U}|)||_\infty = ||(|\tilde{L}|(u_i))||_\infty$, $||(|\tilde{L}| \cdot |\tilde{U}|)||_1 = ||((l_j)^T|\tilde{U}|)||_1$, where $(l_j)$ and $(u_i)$ are two vectors with the components $l_j = \sum_i |\tilde{l}_{i,j}|$ and $u_i = \sum_j |\tilde{u}_{i,j}|$, provided that $\tilde{L} = (\tilde{l}_{i,j})$, $\tilde{U} = (\tilde{u}_{i,j})$,

The entire computation at stages 1–3 requires fewer arithmetic operations than the known exact solution algorithms of [C92], [ABDPY97], [BEPP97], [BEPP99], with our further benefit stemmed from performing these operations with single IEEE precision.

The major advantage of our algorithm, versus the known arithmetic filters, is that it produces the certified sign det $A$ even where the estimated roundoff error of the determinant computation is large. In this case, the computations of stage 3 are invoked, where the bound $e$ on the roundoff errors of Gaussian elimination tends to be proportional to $\gamma_n||A||$ according to the decades of practical experience [GL96], versus the much higher order of $\gamma_n||A||^n$, for the roundoff error of computing det $A$. Such a superiority of our filter versus the ones of [PYS97] (specialized for the computation of det $A$) was confirmed by the results of numerical experiments reported in [PY98], [PY99]. The latter papers also contain some further details of the algorithm its modifications and extensions, some recipes in the case of its failure, and its theoretical and experimental comparison with the known arithmetic filters for det $A$.

## References

[ABDPY97] F. Avnaim, J.-D. Boissonnat, O. Devillers, F. P. Preparata, M. Yvinec, Evaluating Signs of Determinants Using Single-Precision Arithmetic, **Algorithmica**, 17, 111-132, 1997.

[BEPP97] H. Brönnimann, I. Z. Emiris, V. Y. Pan, S. Pion, Computing Exact Geometric Predicates Using Modular Arithmetic with Single Precision, **Proc. 13th Ann. ACM Symp. on Computational Geometry**, 174-182, ACM Press, New York, 1997.

[BEPP99] H. Brönnimann, I. Z. Emiris, V. Y. Pan, S. Pion, Sign Determination in Residue Number Systems, **Theoretical Computer Science**, 210, 1, 173-197, 1999.

[BKMNSU95] C. Burnikel, J. Könnemann, K. Mehlhorn, S. Näher, S. Schirra, C. Uhrig, Exact Geometric Computation in *LEDA*, **Proc. 11th Ann. ACM Symp. on Computational Geometry**, C18– C19, 1995.
Package available at *http://www.mpi- sb.mpg.de/LEDA/leda.html.*

[BP94] D. Bini, V. Y. Pan, **Polynomial and Matrix Computations, vol. 1: Fundamental Algorithms**, Birkhäuser, Boston, 1994.

[C92] K.L. Clarkson, Safe and Effective Determinant Evaluation, **Proc. 33rd Annual IEEE Symp. on Foundations of Computer Science**, 387-395, IEEE Computer Society Press, 1992.

[EY39] C. Eckart, G. Young, A Principal Axis Transformation for Non-Hermitian Matrices, **Bull. Amer. Math. Society** (New Series), 45, 118–121, 1939.

[FvW93] S. Fortune, C. J. Van Wyk, Efficient Exact Arithmetic for Computational Geometry, **Proc. 9th Ann. ACM Symp. on Computational Geometry**, 163– 172, 1993.

[FvW96] S. Fortune, C. J. Van Wyk, Static Analysis Yields Efficient Exact Integer Arithmetic for Computational Geometry, **ACM Trans. Graph,** 15, 3, 223–248, July 1996.

[GL96] G. H. Golub, C. F. Van Loan, **Matrix Computations**, Johns Hopkins Univ. Press, Baltimore, MD, 1996 (third edition).

[H96] N. J. Higham, **Accuracy and Stability of Numerical Algorithms**, SIAM Publications, Philadelphia, 1996.

[IEEE87] IEEE Standard 754-1985 for Binary Floating Point Arithmetic, Reprinted in **SIGPLAN Notices**, 22, 2, 9–25, 1987.

[K66] W. Kahan, Numerical Linear Algebra, **Canadian Math. Bull.**, 9, 757–801, 1966.

[K95] E. Kaltofen, Analysis of Coppersmith's Block Wiedemann Algorithm for the Parallel Solution of Sparse Linear Systems, **Math. of Comput.,** 64, 210, 777-806, 1995.

[KP91] E. Kaltofen, V. Y. Pan, Processor Efficient Parallel Solution of Linear Systems over an Abstract Field, **Proc. 3rd Ann. ACM Symp. on Paralled Algorithms and Architectures,** 180-191, ACM Press, New York, 1991.

[KP92] E. Kaltofen, V. Y. Pan, Processor Efficient Parallel Solution of Linear Systems II. The Positive Characteristic and Singular Cases, **Proc. 33rd Ann. IEEE Symp. on Foundation of Computer Science,** 714-723, IEEE Computer Society Press, 1992.

[PYS97] V. Y. Pan, Y. Yu, C. Stewart, Algebraic and Numerical Techniques for the Computation of Matrix Determinants, **Computers & Math. (with Applications)**, 34, 1, 43-70, 1997.

[PY98] V. Y. Pan, Y. Yu. Certified Numerical Computation of the Sign of a Matrix Determinant, MSRI Preprint No. 1998-070, **Math. Science Research Institute**, Berkeley, California, 1998, accepted for a special issue of **Algorithmica** on Algorithm Engineering.

[PY99] V. Y. Pan, Y. Yu, Certified Computation of the Sign of a Matrix Determinant, **Proc. 10th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA99)**, 715–724, ACM press, New York, and SIAM Publications, Philadelphia, 1999.

[Wi] D. H. Wiedemann, Solving Sparse Linear Equations Over Finite Fields, **IEEE Trans. on Inf. Theory**, IT-32, 1 , 54-62, 1986.

[YD95] C. K. Yap, T. Dubhe, The Exact Computation Paradigm, D. Du and F. Hwang, editors, **Computing in Euclidean Geometry**. World Scientific Press, 1995.