Certified Numerical Computation of the Sign of a Matrix Determinant *, †

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

Yanqiang Yu
Ph.D. Program in Computer Science, CUNY
New York, NY 10036

December 14, 1998

Abstract

Certified computation of the sign of a matrix determinant is a central problem in computational geometry. 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, high precision computation of det A is required to ensure that the error bound is smaller than the magnitude of the determinant. 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 factorizaion of a matrix by Gaussian elimination with pivoting 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 these observations, we show the resulting simplified methods for the certified computation of the sign, compare them with other approaches, observe the possibility of effective combination of our methods with some known symbolic methods for this problem, and confirm the efficiency of our techniques by some numerical tests.

*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.

1 Introduction.

1.1 The problem and the background.

The classical problem of computing det A, the determinant of an $n \times n$ matrix A, has long history (see e.g. [Mu], [Ma], [Du], [Fo], [R52], [E67], [B68], [P88], [BP]). Recently, it turned out that some of the most fundamental problems of computational geometry (such as the computation of convex hulls and Voronoi diagrams) are reduced to the computation of $\det A$ or, more precisely, its sign, that is, testing whether $\det A = 0$, $\det A > 0$, or $\det A < 0$ [BEPP97], [BEPP98], [BKMNSU], [EC], [Em], [FVW], [Y97], [YD].

In many areas of computational geometry, lower dimensional problems must be solved, and then n ranges between 2 and 10, usually staying below 5. In this class of applications, the matrix A is filled with "long" numbers, representing the real data with a high precision (and thus allowing to treat the important case of a nearly singular input). In another major class of applications [DL94], [DL97], [ES], [AF], [FR], [MA], n is large (say, in the range from 100 to 500), whereas the matrix A is filled with relatively short integers (say, represented with 5 to 10 bits). Such applications include the computation of the orientation of a polyhedron or an algebraic variety in a high-dimensional space (for instance, such computations are required in the area of convex optimization in statistical physics and chemistry).

In both cases we may apply the well known methods to compute $\det A$ based on the triangular ($PLUP_1$) or orthogonal (QR) factorization of the matrix A. High speed of these computations is ensured as they are performed numerically, with a fixed (single or double) precision, which currently has much faster computer implementation than rational, integer, and multiple precision arithmetics. The major problem, however, is to certify that the output is correct in the presence of roundoff errors.

Substantial advance in this area was the paper [C], though the correctness certification of the output of the proposed algorithm (based on the modified Gram-Schmidt method) complicated and slowed down the computation. The algorithm of [ABDPY] competes with one of [C] for $n \leq 4$ but does not work well for larger n.

Recent progress reported in [BEPP97], [BEPP98] relies on using symbolic algorithm that computes $\det A$ modulo several primes $p_1, \ldots, p_k$ such that their product exceeds $|\det A|$. The papers propose effective algorithms for the recovery of the sign of $\det A$ from these data, based on some novel application and extensions of the Chinese remainder algorithm, which [BEPP97] and [BEPP98] reduce to single or double precision computation. The algorithms of [BEPP97] and [BEPP98] seem to be among the currently best ones for the problem. Their bottleneck is the relatively expensive computation of $(\det A) \bmod m_i$, for $i = 1, \ldots, k$. The number k of the pairwise relatively prime moduli m_i involved and, consequently, the computational cost decrease if $|\det A|$ is shown to be smaller.

The algorithms of [PYS] complement ones of [BEPP97], [BEPP98] by computing

$\det A$ numerically. The correctness of the output is certified unless the algorithm establishes a relatively small upper bound on $|\det A|$. This would be an ideal example of effective combination of symbolic and numerical techniques, but numerical experiments show that the techniques of [PYS] give too rough bounds, greatly exceeding the actual value of $|\det A|$. Such a phenomenon occurs because the range for the values of $|\det A|$ is huge (from 0 to D^+ with D^+ on the level of Hadamard's determinant bound, which can be as large as $||A||^n$), and the known techniques of error analysis only guarantee roundoff error bounds of order $D^+ n^2 \epsilon$, ϵ being the *machine epsilon* (allso called *unit roundoff*). Such bounds can be large even where $|\det A|$ is actually small (cf. section 8.3). Therefore, the roundoff error bounds for computing $\det A$ may exceed the value $|\det A|$ substantially.

1.2 Our results.

In the present paper we propose some novel techniques for certified computation of the sign of $\det A$ and devise new effective algorithms based on these techniques and promising the be practical. Our numerical experiments confirm the efficiency of these techniques and algorithms and their substantial superiority to the ones [PYS]. Our main algorithm 4.1 uses some novel techniques to computes $\det A$ numerically and either certifies that the sign of $\det A$ has been computed correctly or shows which increase of the precision of computing should yield the certified output. If the required increase of the precision is too large to be handled by available computer support, then $|\det A|$ must be relatively small, and the transition to the symbolic approach is appropriate. Motivated by this observation, we also show how to compute or estimate from above the value of $|\det A|$, which is an input value for the symbolic approach.

All our computations involve order of n^3 (single precision) flops and essentially amount to the invocation of some widely available subroutines for matrix computations, which we select and combine in an appropriate order (cf. sections 4, 6 and 7). A major step of our algorithm is the estimation of the distance to a closest singular matrix, and we describe several known techniques for this task, depending, in particular, on allowing or not allowing randomization. We also briefly examine modifications of our approach in section 8.

The major technical novelty behind our improvement versus [PYS] is the certification of the sign of $\det A$ without estimating the roundoff error of computing $\det A$. Instead, we estimate the minimum distance from the matrix $\tilde{L}\tilde{U}$ to a singular matrix, $\tilde{L}$ and $\tilde{U}$ being the computed approximations to the factors L and U of the triangular ($PLUP_1$) factorization of A. By a well-known theorem [EY], [K66], this distance is equal to $1/N$, for $N = ||\tilde{U}^{-1}\tilde{L}^{-1}||$.

Due to the celebrated techniques of backward error analysis, the roundoff errors of computing $\tilde{L}$ and $\tilde{U}$ can be equivalently represented by the errors caused by perturbation of A by some matrix E whose norm is easily estimated from above. On the other hand, we deduce easily that $\det(A+E)$ does not change its sign when E ranges in the ball of radius $1/N = 1/||\tilde{U}^{-1}\tilde{L}^{-1}||$ centered in the origin. Now, since the matrices $\tilde{L}$ and $\tilde{U}$ are two available triangular matrices, it is not hard to obtain a certified and quite tight upper bound on N at the cost of performing $O(n^3)$ arithmetic operations and comparisons. The proposed approach to certified computation of the sign of the determinant seems to be novel in spite of its relative simplicity and efficiency.

Also, our combination of numerical and algebraic (residue) computations proposed in section 8.5 seems to be technically novel. Such a combination enables rapid certified computation of sign$(\det A)$ by using symbolic computation of $\det A$ modulo a small prime already in the case where $|\det A|$ (rather that $\det A$) is closely approximated numerically. This may be also of some practical interest because one may effectively approximate $|\det A|$ by computing numerically the LDL^T factorization of the matrix A^TA, A^T denoting the transpose of A.

1.3 The order of our presentation.

We present our results in the following order. In the next section, we state some definitions. In section 3, we recall some known estimates for the errors of Gaussian elimination due to roundoff. In section 4, we relate the certification of the sign of $\det A$ in the presence of roundoff errors to the minimum distance from A to a singular matrix. Then we propose our main algorithm for computing and certifying the sign of $\det A$ based on this relation. In section 5, we show how to use the computed information in the case where the algorithm does not produce a certified correct output. In section 6, we elaborate the stage of estimating the minimum distance to a singular matrix, which is a major block of our algorithm of section 4. In section 7, we complement the algorithm by presenting some techniques for computing or estimating from above $|\det A|$. In section 8, we comment on some variations of our approach and some alternatives. In particular, we indicate some reasons for the deficiency of an approach of [PYS] and of one based on the Barrlund and Sun theorem; we also point out the modifications of our approach that use Gauss-Jordan, $PLDM^TP_1$, QR (rather than $PLUP_1$) factorizations of A or the LDL^T factorization of A^TA. In the appendix, we recall some techniques for computing $\tilde{U}^{-1}\tilde{L}^{-1}$ and present the results of our numerical experiments.

Sections 1-8 and Appendix A are due to the first author, Appendix B on numerical tests to both authors.

Acknowledgement. Victor Pan thanks Dario Bini for the preliminary discussions on some alternative approaches to certified numerical computation of the sign of matrix determinant.

2 Some definitions.

Hereafter, $\mathbf{A}_k$ denotes the k-th column vector of a matrix A. $w_{i,j}$ denotes the (i,j)-th entry of a matrix $W = (w_{i,j})$. $\|.\| = \|.\|_h$ denotes a fixed operator matrix norm, in particular we will use the 2-norm $\|.\|_2$, the row norm $\|W\|_\infty = \max_i \sum_j |w_{i,j}|$, and the column norm $\|W\|_1 = \max_j \sum_i |w_{i,j}|$ for a matrix $W = (w_{i,j})$ (cf. [GL96] or [H96]). The transpose of W is denoted $W^T = (w_{j,i})$, whereas $|W|$ denotes the matrix $(|w_{i,j}|)$. We write $|W| \le |V|$ iff $V = (v_{i,j})$ and $|w_{i,j}| \le |v_{i,j}|$ for all i and j. I denotes the $n \times n$ identity matrix. $\mathrm{diag}(w_{i,i})$ denotes the diagonal matrix with the diagonal entries $w_{i,i}$, $i = 1, \dots, n$. $\det W$ and $\mathrm{sign}(\det W)$ denote the determinant of a square matrix W and its sign, respectively. A triangular matrix will be called *unit* (respectively, *proper*) triangular if its diagonal is filled with ones (respectively, zeros).

3 Roundoff errors of matrix factorization by Gaussian elimination.

$\det A$ and its sign for a given matrix A can be immediately obtained from the triangular $(PLUP_1)$ factorization of A, but the problem is to analyze the effect of the roundoff errors when the factorization is computed numerically. In this analysis we will apply some known estimates, which we will recall in this section.

Theorem 3.1 (cf. [H96], Theorem 9.3, page 175). *Let*

$$A = PA'P_1, \;\; \tilde{A} = A' + E = \tilde{L}\tilde{U}, \tag{3.1}$$

where A, A', $\tilde{A}$, E, P, P_1, $\tilde{L}$, and $\tilde{U}$ are $n \times n$ matrices, P and P_1 are permutation matrices, $\tilde{L} = (\tilde{l}_{i,j})$ is a unit lower triangular matrix (so that $\det \tilde{L} = 1$), and $\tilde{U} = (\tilde{u}_{i,j})$ is an upper triangular matrix, $\tilde{L}$ and $\tilde{U}$ are computed numerically, with unit roundoff ϵ (machine epsilon), by means of Gaussian elimination (with complete pivoting) applied to the matrix A. Then

$$|E| \le \gamma_n |\tilde{L}| \cdot |\tilde{U}|, \;\; \gamma_n = n\epsilon/(1 - n\epsilon).$$

Two special cases of this result cover Gaussian elimination with partial pivoting (for $P_1 = I$) and with no pivoting (for $P_1 = P = I$).

By (3.1) we have

$$\det A = (\det P)(\det A') \det P_1 \;, \tag{3.2}$$

$$\det \tilde{A} = \det \tilde{U} = \tilde{u}_{1,1}\tilde{u}_{2,2} \cdots \tilde{u}_{n,n} \;. \tag{3.3}$$

Since $\det P$ and $\det P_1$ are readily available (they equal 1 or -1), the equations (3.2) and (3.3) define $\mathrm{sign}(\det A)$ provided that the diagonal entries of $\tilde{U}$ are available and that ϵ is sufficiently small to guarantee that

$$\mathrm{sign}(\det A') = \mathrm{sign}(\det \tilde{A}) \;. \tag{3.4}$$

In the next section we will show how to verify (3.4) by using the following corollary of theorem 3.1.

Corollary 3.1 *Under the assumptions of theorem 3.1, we have*

$$\|E\| \leq e = ||(|\tilde{L}| \cdot |\tilde{U}|)||\gamma_n \tag{3.5}$$

for any fixed operator matrix norm.

The computation of the row and column norms of $|\tilde{L}| \cdot |\tilde{U}|$ involves only $O(n^2)$ arithmetic operations 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 $u_i = \sum_j |\tilde{u}_{i,j}|$.

By corollary 3.1, the rounding error of the computation of the $PLUP_1$ factorization of A is bounded in terms of γ_n and the norm of the matrix $|\tilde{L}| \cdot |\tilde{U}|$. It is known that in the case of using complete pivoting, the (k, j)-th entries $\tilde{a}_{k,j}, \tilde{l}_{k,j}$ and $\tilde{u}_{k,j}$ of the matrices $\tilde{A}, \tilde{L}$ and $\tilde{U}$ of (3.1), respectively, satisfy the bounds

$$|\tilde{l}_{k,j}| \leq 1, \ \ |\tilde{u}_{k,j}| \leq \rho_k \max_{g,h} |\tilde{a}_{g,h}|,$$

for all k and j, where $\rho_k \leq k^{1/2}(2 \cdot 3^{1/2} \cdots k^{1/(k-1)})^{1/2} \leq 1.8k^{(\ln k)/4}$ (cf. [GL96], p.119). The same bounds are known in the case of partial pivoting, but theoretically only for $\rho_k \leq 2^{k-1}$. In practice, however, $\rho_k = O(k)$ even in the case of using partial pivoting (cf. [GL96], p.116). Some improvement of the worst case error bound (even versus the case of complete pivoting) can be obtained by means of symmetrization (see section 8.5).

4 Certification of the sign in terms of the smallest distance to a singular matrix.

The following sufficient condition for (3.4),

$$|\det \tilde{U}| = |\det \tilde{A}| > |(\det A) - \det(P\tilde{A}P_1)| = e_d,$$

can be verified based on the straightforward crude estimate :

$$e_d \leq n^2 e_+ D_+,$$

where e_+ denotes the maximum absolute value of the entries of E, and $D_+ = \prod_{k=1}^n (||\mathbf{A}_k||_2 + ne_+)$. This estimate is based on Hadamard's bound,

$$|\det A| \leq \prod_{k=1}^{n} ||\mathbf{A}_k||_2 \tag{4.1}$$

(cf. [H96], p.287). It is not easy to compute e_+, but we may replace e_+ by its upper bound $e^* = \gamma_n \ \max_{i,j}(|\tilde{L}| \cdot |\tilde{U}|)_{i,j}$ implied by theorem 3.1. Here $(|\tilde{L}| \cdot |\tilde{U}|)_{i,j}$ denotes the (i, j)-th entry of the matrix $|\tilde{L}| \cdot |\tilde{U}|$. Then we obtain the following estimate:

$$e_d \leq e_d^+ = D^+ n^2 e^*, \;\; D^+ = \prod_{k=1}^{n} (||\mathbf{A}_k||_2 + ne^*). \tag{4.2}$$

Our more refined techniques for the verification of equation (3.4) will rely on the next two results.

Theorem 4.1 *For two given matrices W and $W+\Delta$, for a fixed matrix norm $\|.\|$ and for all singular matrices S, let*

$$\max\{\|W-S\|, \|W+\Delta-S\|\} > \|\Delta\|. \tag{4.3}$$

Then

$$\text{sign}(\det W) = \text{sign}(\det(W+\Delta)) \;. \tag{4.4}$$

Proof Unless (4.4) holds, $S = W + t\Delta$ is a singular matrix for some real t, $0 \leq t \leq 1$. Clearly, for such a matrix S, we have

$$\|W-S\| = t\|\Delta\| \leq \|\Delta\|,$$

$$\|W+\Delta-S\| = (1-t)\|\Delta\| \leq \|\Delta\|,$$

and (4.3) is violated. □

The next theorem was first proved by Eckart and Young, in 1939, [EY], in the special case of the matrix norm $||\cdot|| = \|.\|_2$ and much later, by Gastinel (cf. [K66]), in the general case of any operator matrix norm $||\cdot||$.

Theorem 4.2 *For any fixed nonsingular matrix W and any fixed operator matrix norm $\|.\|$, we have $\frac{1}{\min\|W-S\|} = \|W^{-1}\|$, where the minimum is over all singular matrices S.*

Combining theorems 4.1 and 4.2 implies the next result.

Corollary 4.1 *Under the assumptions of theorem 4.1, if $1/\min\{\|W^{-1}\|, \;\|(W+\Delta)^{-1}\|\} > \|\Delta\|$, then (4.4) holds.*

Apply corollary 4.1 to $W = A'$ and $\Delta = \tilde{A} - A'$ to obtain the next result.

Corollary 4.2 *The equation (3.4) holds if $\|E\| < 1/\min\{\|(A')^{-1}\|, \;\|\tilde{A}^{-1}\|\}$.*

Now we are ready to propose an algorithm for computing $\text{sign}(\det A)$, where at stage 2 we rely on simple bound (4.2), which suffices for a large class of inputs, and at the next stage we rely on corollary 4.2.

Algorithm 4.1.

Input: an $n \times n$ matrix A, a fixed matrix norm ($||\cdot||_\infty$ or $||\cdot||_2$), and a unit roundoff ϵ.

Output: either the certified value of $\text{sign}(\det A)$ or FAILURE (cf. section 5).

Computations.

1. Apply Gaussian elimination (with complete, partial, or no pivoting) using the unit roundoff ϵ, to compute the matrices $\tilde{L}$ and $\tilde{U}$ of (3.1).

2. Compute an upper bound on e_d (in particular, we may use the simple crude bound e_d^+ of (4.2) and check if $|\det \tilde{U}|$ exceeds this bound. If so, compute and output sign $(\det A)$ based on (3.2)-(3.4).

3. Otherwise, estimate the norm $N = \|\tilde{A}^{-1}\| = \|\tilde{U}^{-1}\tilde{L}^{-1}\|$ from above and/or below (see section 6) to decide whether

$$eN < 1. \tag{4.5}$$

If the latter inequality is verified, compute and output sign $(\det A)$ based on (3.2)-(3.4). Otherwise output FAILURE.

The complexity of the computations by the algorithm is dominated by the estimated complexity of its stages 1 and 3. We refer the reader to [GL96] and [BP] on the complexity of stage 1 and to section 6 on the complexity of stage 3. In both cases we need $O(n^3)$ arithmetic operations. At stage 1, we may also need $O(n^3)$ or $O(n^2)$ comparisons for complete or partial pivoting, respectively.

5 Some recipes in the case of FAILURE.

Suppose that algorithm 4.1 has output FAILURE and that an upper bound N^+ on N and the value $f = eN^+$ are available. Then we have several options:

1. Repeat the computation but with the unit roundoff $\epsilon_{new} = c\epsilon_{old}/f$ for some heuristic choice of $c < 1$. The value c can be adapted dynamically, depending on the resulting change of the value eN^+. If the latter value changes proportionally to ϵ_{new} (as can be expected unless A is a very ill-conditioned matrix), then (4.5) holds for $\epsilon = \epsilon_{new} = c\epsilon_{old}/f$ and for any $c < 1$. Recomputation of $\tilde{L}$ and $\tilde{U}$ for ϵ_{new} can be simplified because several leading digits in the representation of the computed values stay invariant, and only the remaining trailing digits must be recomputed (compare [EPY]).

2. Unless it is known already that $N^- \geq 1/e$, try to improve the upper estimate N^+ for $N = \|\tilde{U}^{-1}\tilde{L}^{-1}\|$ at stage 3 (see the next section).

3. If the value f is too large so that numerical computations with a unit roundoff $\epsilon_{new} < \epsilon_{old}/f$ become too expensive, shift to the symbolic algorithms of [BEPP97] and [BEPP98]. In this case, one needs an a priori upper bound on $|\det A|$. Hadamard's bound (4.1) or the bound $|\det A| \leq e_d^+ + |\det \tilde{U}|$ can be used, but it may pay to refine these bounds by performing some additional computations (see sections 7 and 8.5).

6 Estimating the minimum distance to a singular matrix.

6.1 Estimating the distance from above.

To estimate from above the minimum distance from the matrix $\tilde{A}$ to a singular matrix or, equivalently, to estimate $N = \|\tilde{U}^{-1}\tilde{L}^{-1}\|$ from below, we may apply the simple iterative algorithm of [CMSW79], which, according to [GL96], "produces a good order-of-magnitude " lower bound N^- on N at the cost of performing $O(jn^2)$ arithmetic operations in j iterations (practically, j is much smaller than n). If (3.5) does not hold even for N replaced by N^-, then we may apply the recipes of section 5, for some heuristic choice of N^+.

6.2 Two-sided estimates with randomization.

Recall that $N = \|\tilde{A}^{-1}\|_2 = \|\tilde{U}^{-1}\tilde{L}^{-1}\|_2 = \sigma_1(\tilde{U}^{-1}\tilde{L}^{-1}) = 1/\sigma_n(\tilde{L}\tilde{U}) = 1/\sigma_n(\tilde{A})$, $\sigma_k(W)$ denoting the k-th largest singular value of a matrix W. This reduces our probem to estimating the smallest singular value $\sigma_n(\tilde{A}) \geq 0$, which is a well known problem of matrix computations whose solution does not require full computation of the singular value decomposition (SVD) of $\tilde{A}$. The problem can be solved by using the inverse power iteration or, better, Lanczos algorithm ([GL96], sect. 8.2.2 and ch.9). Both of these algorithms require an initial vector, which can be chosen at random. Both use $O(n^2)$ flops per iteration, but Lanczos algorithm converges faster (cf. [GL96], p.477). (The cost of each iteration in our case is dominated by the cost of the solution of four linear systems of equations with the already available triangular coefficient matrices $\tilde{L}^T, \tilde{U}^T$, $\tilde{U}$ and $\tilde{L}$.) The probability of obtaining accurate approximation to $\sigma_n(\tilde{A})$ depends on the number of iterative steps. Dixon in [D83] shows that, with a probability at least $P_{I.POWER}(k,\ \Theta) = 1 - 0.8\Theta^{-k/2}n^{1/2}$, the lower bound $N_k^- = (\mathbf{x}^T(\tilde{A}\tilde{A}^T)^{-1}\mathbf{x})^{1/2k} \leq ||\tilde{A}^{-1}||_2 = 1/\sigma_n(\tilde{A})$ on N is also an upper bound within the factor $\Theta > 1$, that is, $\Theta N_k^- \geq ||\tilde{A}^{-1}||_2$, for $k \geq 1$ and a random choice of a vector $\mathbf{x}$ on the unit sphere $S_n = \{\mathbf{x} : \mathbf{x}^T\mathbf{x} = 1\}$, under the uniform probability distribution on S_n. Dixon deduced this estimate for the inverse power method applied to approximation of the smallest eigenvalue of $\tilde{A}\tilde{A}^T$, which is the smallest singular value of $\tilde{A}$.

An alternative approach produces two-sided estimates for N by means of Lanczos algorithm. Lanczos algorithm computes σ_n^* satisfying $\sigma_n^* \geq \sigma_n(\tilde{L}\tilde{U})$. The estimates for the probability $P_{LANCZOS}(l, \Theta)$ that $\sigma_n^*/\sigma_n(\tilde{L}\tilde{U}) \leq \Theta$, for a fixed $\Theta > 1$ and nonsingular $\tilde{L}$ and $\tilde{U}$, can be expressed as functions in the number l of Lanczos iterations for a random choice of the initial vector (cf. [KW92], [KW94]).

6.3 Deterministic lower estimates for the distance.

Section 8.3 of [H96], pages 159-161, shows some techniques for rapid computation of some crude upper bounds on $||T^{-1}||$ for triangular matrices T, and this can be immediately translated into rapid computation of some crude upper bounds on $N = \|\tilde{U}^{-1}\tilde{L}^{-1}\| \leq \|\tilde{U}^{-1}|| \cdot ||\tilde{L}^{-1}\|$. To yield more refined upper bounds on N, one may actually compute the inverses $\tilde{U}^{-1}$ and $\tilde{L}^{-1}$, then their product X and finally (an upper bound on) the norm $N = \|\tilde{U}^{-1}\tilde{L}^{-1}\|$. Detailed presentation of such computations can be found in [H96], sections 13.2-13.3, pages 265-275. For reader's convenience, we sketch some of these algorithms in appendix A.

Assumming the computations with unit roundoff ϵ, [H96] presents the estimates for the residual norms $\|\tilde{A}X - I\| = r(X)$, $\|X\tilde{A} - I\| = r^*(X)$, and the error norms $\|\tilde{A}^{-1} - X\| = e(X)$ for the computed approximations X to $\tilde{U}^{-1}\tilde{L}^{-1} = \tilde{A}^{-1}$. The estimates are given in the form

$$r(X) \leq c_n\epsilon||\tilde{L}|| \cdot ||\tilde{U}|| \cdot ||X||,$$

$$r^*(X) \leq c_n^*\epsilon||\tilde{L}|| \cdot ||\tilde{U}|| \cdot ||X||,$$

$$e(X) \leq c_n'\epsilon||\tilde{L}|| \cdot ||\tilde{U}|| \cdot ||X|| \cdot ||\tilde{A}^{-1}||.$$

Here c_n, c_n^* and c_n' are constants independent of ϵ and $\tilde{A}$, which can be elaborated by using the error analysis techniques of [H96].

Instead of applying these estimates, we may directly compute $\tilde{A}X - I$ or $X\tilde{A} - I$ (in $O(n^3)$ operations) and arrive at the residual norms $r(X)$ or $r^*(X)$ within the roundoff error bound $\gamma_n||\tilde{A}|| \cdot ||X||$, $\gamma_n = \epsilon n/(1 - \epsilon n)$ (cf. [H96], page 78). For computing the row and column norms of the matrix, we may apply recursive pairwise summation, whose relative roundoff error is at most $\gamma_{\lceil \log_2 n \rceil}$ (cf. [H96], page 92). If $r = \min\{r(X), r^*(X)\} < 1$, we immediately estimate that $e(X) \leq r||X||/(1 - r)$.

Remark 6.1 To improve the approximation to $\tilde{A}^{-1}$ by $X = X_0$ one may apply Newton's iteration, $X_{i+1} = X_i(2I - AX_i)$, whose i-th iterative step for every i squares the residual matrices $\tilde{A}X_i - I$ and $X_i\tilde{A} - I$ and consequently squares the upper bounds on their norms (cf. [PS]).

7 Estimating the magnitude of the determinant.

One may refine the upper bounds of section 5 on $|\det A|$ by relying on the following well known fact (cf. section 8.5 for an alternative way to the refinement):

Fact 7.1

$$|\det A| = \prod_{i=1}^{k} \sigma_i \ ,$$

where $\sigma_1, \ldots, \sigma_r$ are the singular values of A, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$, $\sigma_{r+i} = 0$ for $i = 1, \ldots, n - r$, $r = \operatorname{rank} A$.

Consequently, $d^+ = \prod_{i=1}^n \sigma_i^+ \geq |\det A|$ if $\sigma_i^+ \geq \sigma_i$, $i = 1, \ldots, n$, and our problem is reduced to approximating $\sigma_1, \ldots, \sigma_n$ from above. The known SVD algorithms ([GL96], section 8.6 and p.254) for the latter task use $7n^3/3 + O(n^2)$ flops. A faster though cruder solution may rely on approximating from above a few smallest singular values $\sigma_n, \sigma_{n-1}, \ldots$ by means of Lanczos algorithm and applying the readily available upper bound $\sigma = \min\{\|\tilde{A}\|_1, \|\tilde{A}\|_\infty\}\|$ on all other σ_i.

8 Some modifications and alternative approaches.

8.1 LDM^T factorization.

Instead of LU factorization of A', one may rely on its LDM^T factorization, where L and M are unit lower triangular matrices and D is a diagonal matrix. One may immediately obtain LDM^T factorization by extending LU factorization as follows: compute the matrices $D = \operatorname{diag}(u_{i,i})$ and $M^T = D^{-1}U$ and then substitute $U = DM^T$ into the LU factorization. Alternatively, one may compute the LDM^T factorization directly, by means of the Gauss-Jordan transformation. In both cases our analysis can be easily extended (cf. [H96], pp.275-281).

8.2 Solution by estimating the magnitude of the diagonal perturbation.

Consider the relative roundoff errors of the diagonal entries of the factor U of $A' = LU$. To prove that equation (3.4) holds, it suffices to verify that all these errors are less than 1. The theorem of Barrlund and Sun (see [H96], page 194) gives us the following sufficient condition: $\|\tilde{G}\|_2 < 1$ and simultaneously $\operatorname{diag}(|\tilde{G}|(I - |\tilde{G}|)^{-1}|\tilde{U}|) < \operatorname{diag}(|\tilde{U}|)$ for $\tilde{G} = \tilde{L}^{-1}(\tilde{A} - A')\tilde{U}^{-1}$.

This approach has some similarity to one of sections 4–5 (in particular $\|\tilde{G}\|$ can be estimated similarly to N^+ of section 6 and appendix A) but requires stronger assumptions and involves a more complicated expression, $|\tilde{G}|(I - |\tilde{G}|)|U|$, whose computation has larger roundoff errors.

8.3 The straightforward approach of [PYS].

Unlike our approach, the paper [PYS] relies on the following equations:

$$\delta = \det(A' + E) - \det A' = \sum_{i,j} e_{i,j} \tilde{D}_{i,j}, \tag{8.1}$$

where $E = (e_{i,j}) = \tilde{A} - A'$, $\tilde{D}_{i,j} = \det \tilde{A}_{i,j}$, $\tilde{A}_{i,j}$ is the $(n-1) \times (n-1)$ matrix obtained by replacing the first $j-1$ columns of A' by ones of $\tilde{A}$ and by deleting the i-th row and the j-th column of the resulting matrix, for $i, j = 1, \ldots, n$. Then the relations $|\delta| \leq n^2 e_+ \max_{i,j} |\tilde{D}_{i,j}| \leq n^2 e_+ (\|A\| + ne_+)^{n-1}$ for a fixed matrix norm are deduced. This upper estimate for $|\delta|$, however, is overly pessimistic because it relies on the rough bound $|\tilde{D}_{i,j}| \leq (\|A\| + ne_+)^{n-1}$ (which could be only slightly improved by using Hadamard's bound (4.1)) and on ignoring possible cancellation in the summation in (8.1). Generally, it is hard to obtain a sharp estimate for δ, and our approach benefits of proposing a solution that avoids estimating δ.

8.4 QR factorization versus $PLUP_1$ factorization.

The proposed algorithms and their analysis can be immediately extended based on the QR rather than $PLUP_1$ factorization of A. Some estimates for the cost of computing the factorization and for the perturbation of the input, which would accomodate the roundoff errors, can be taken from [PYS] but refined by carefully estimating from above the norm $||R^{-1}Q^T||$ (cf. [H96], chapter 18). This analysis gives upper hand to Householder transformations as a method of choice for the computation of the QR factorization. (Actually, besides the sign of $\det Q$, which is usually readily available, we only need the diagonal entries of R for our purpose of the sign computation.) Relying on the QR factorization has advantage over using triangular factorizations when the sign of $\det A$ must be computed for a dynamically updated matrix A (see [GGMS], [PYS]). To decrease the roundoff errors, one may apply a scaled version of QR factorization, making it free of square root computation (cf. [C]).

8.5 Sign determination via LDL^T factorization of A^TA.

Computation of the LDL^T factorization of the symmetric matrix A^TA is simple and has very good numerical stability (cf. [GL96], pp.138-139, and [H96], pp.207-209). Namely, the two matrices of the roundoff errors of computing A^TA and the LDL^T factors of A^TA have norms bounded from above by $\gamma_n||A|| \cdot ||A^T||$ and $\frac{\gamma_{n+1}}{1-\gamma_{n+1}} \sum_i a_{i,i}^2$, respectively ([H96], p.78). Thus, such a factorization may serve as a basis for good numerical approximation of $(\det A)^2 = \det(A^TA) = \prod_i D_{i,i}$ and then of $|\det A|$.

Let us extend this observation to a certified algorithm for sign $(\det A)$. We will assume that we have computed $\tilde{d} \geq 0$ and a relatively small $\Delta > 0$ such that

$$\tilde{d} - \Delta < |\det A| < \tilde{d} + \Delta.$$

If $\tilde{d} < \Delta$, then $|\det A| < 2\Delta$, and the symbolic algorithms of [BEPP97], [BEPP98] effectively compute sign $(\det A)$. If $\tilde{d} \geq \Delta$, we choose some prime $p \geq 4\Delta$ lying close to 4Δ and compute $d_p = (\det A) \bmod p$ defined as a unique value in the semi-open interval $[-p/2, p/2)$ such that p divides $d_p - \det A$. Similarly we define $\tilde{d}_p = \tilde{d} \bmod p$. (To obtain d_p, we first compute (modulo p) a PLU factorization of A and sign$(\det P)$ and then compute $d_p = ((\prod_i (u_{i,i} \bmod p))\text{sign}(\det P)) \bmod p$. Note that the computation modulo p only requires $2\lceil \log_2 p \rceil$-bit precision.)

Finally, let us explain the transition from Δ, p, d_p and $\tilde{d}_p$ to sign $(\det A)$ Clearly, we have either

$$-\Delta < \det A - \tilde{d} = (d_p - \tilde{d}_p) \bmod p < \Delta$$

or

$$-\Delta < \det A + \tilde{d} = (d_p + \tilde{d}_p) \bmod p < \Delta.$$

These two cases cannot occur simultaneausly since $(2\tilde{d}) \bmod p = (2\tilde{d}_p) \bmod p \geq 2\Delta$, and we may easily check which of them actually occurs. In the former case, $\det A > \tilde{d} - \Delta \geq 0$, and we output $\text{sign}(\det A) = 1$. Otherwise, $\det A < \Delta - \tilde{d} \leq 0$, and we output $\text{sign}(\det A) = -1$.

References

[ABDPY] 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.

[AF] D. Avis, K. Fukuda, A Pivoting Algorithm for Convex Hulls and Vertex Enumeration of Arrangements and Polyhedra, **Discrete Comput. Geometry**, 8, 295–313, 1992.

[B68] E. H. Bareiss, Sylvester's Identity and Multistep Integer-Preserving Gaussian Elimination, **Math. of Comp.**, 22, 565–578, 1968.

[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.

[BEPP98] H. Brönnimann, I. Z. Emiris, V. Y. Pan, S. Pion, Sign Determination in Residue Number Systems, **Theoretical Computer Science**, 210, 1, 173-197, 1998.

[BKMNSU] 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.*

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

[C] 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.

[CMSW79] A. K. Cline, C. B. Moler, G. W. Stewart, J. H. Wilkinson, An Estimate for the Condition Number of a Matrix, **SIAM J. Numerical Analysis**, 16, 368–375, 1979.

[D83] JJ. D. Dixon, Estimating Extremal Eigenvalues and Conditional Numbers of Matrices, **SIAM J. on Numerical Analysis**, 20, 4, 812-814, 1983.

[DL94] M. Deza, M. Laurent, Applications of Cut Polyhedra, **J. of Computational and Applied Math.**, 55, 1, 191–216, and 55, 2, 217–247, 1994.

[DL97] M. M. Deza, M. Laurent, **Geometry of Cuts and Metrics**, Springer, Berlin, 1997.

[Du] E. Durand, **Solutions Numériques des Équations Algébriques. Vol. II: Systèmes de Plusieurs Équations. Valeurs Propres des Matrices**, Masson et Cie, Paris, 1961.

[E67] J. Edmonds, Systems of Distinct Representatives and Linear Algebra, **J. Res. Nat. Bur. Standards**, Sect. B, 71, 4, 241–245, 1967.

[EC] I.Z. Emiris, J.F. Canny, A General Approach to Removing Degeneracies, **SIAM J. Computing**, 24, 3, 650–664, 1995.

[Em] I.Z. Emiris, A Complete Implementation for Computing General Dimensional Convex Hulls, **Intern. J. Computational Geom. & Applications**, to appear.

[EPY] I. Z. Emiris, V. Y. Pan, Y. Yu, Modular Arithmetic for Linear Algebra Computations in the Real Field, **J. of Symbolic Computation**, 21, 1-17, 1998.

[ES] R. M. Erdahl, V. H. Smith (editors), **Density Matrices and Density Functionals**, Proc. of the A. John Coleman Symp., Reidel, Dordrecht, 1987.

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

[Fo] L. Fox, **An Introduction to Numerical Linear Algebra**, Clarendon Press, Oxford, 1964.

[FR] K. Fukuda, V. Rosta, Combinatorial Face Enumeration in Convex Polytopes, **Computational Geometry, Theory and Applications**, 4, 191–198, 1994.

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

[GGMS] P.E. Gill, G.H. Golub, W. Murray, M.A. Saunders, Methods for Modifying Matrix Factorizations, **Math. of Computation**, 28, 505-535, 1974.

[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.

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

[KW92] J. Kuczyński, H. Woźniakowski, Estimating the Largest Eigenvalue by the Power and Lanczos Algorithms with a Random Start, **SIAM J. on Matrix Analysis and Applications**, 13, 4, 1094-1122, 1992.

[KW94] J. Kuczyński, H. Woźniakowski, Probabilistic Bounds on the Extremal Eigenvalues and Condition Number by the Lanczos Algorithm, **SIAM J. on Matrix Analysis and Applications**, 15, 2, 672-691, 1994.

[Ma] R. H. Macmillan, A New Method for the Numerical Evaluation of Determinants, **J. Roy. Aeronaut. Soc.**, 59, 772ff, 1955.

[MA] R. E. M. Moore, I. O. Angell, Voronoi Polygons and Polyhedra, **J. of Computational Physics**, 105, 301–305, 1993.

[Mu] T. Muir, **The Theory of Determinants in Historical Order of Development**, four volumes bound as two (I, 1693–1841; II, 1841–1860; III, 1861–1880; IV, 1881–1900), Dover, New York, 1960; **Contributions to the History of Determinants**, 1900–1920, Blackie and Son, London, 1930 and 1950.

[P88] V.Y. Pan, Computing the Determinant and the Characteristic Polynomial of a Matrix via Solving Linear Systems of Equations, **Information Processing Letters**, 28, 2, 71-75, 1988.

[PS] V. Y. Pan, R. Schreiber, An Improved Newton Iteration for the Generalized Inverse of a Matrix, with Applications, **SIAM J. Sci. Stat. Comput.,** 12, 5, 1109-1131, 1991.

[PYS] 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.

[R52] J. B. Rosser, A Method of Computing Exact Inverses of Matrices with Integer Coefficients, **J. Res. Na. Bur. Standards**, Sect. B, 49, 349–358, 1952.

[Y97] C. Yap, Towards Exact Geometric Computation, **Computational Geometry, Theory and Applications,** 7, 3–23, 1997.

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

Appendix A. Computation of the inverse.

To compute the inverse $\tilde{A}^{-1} = \tilde{U}^{-1}\tilde{L}^{-1}$, we may rely on the following simple observations (cf. [H96], pp.170-174).

Theorem A.1 *Let $T = (t_1, \ldots, t_n)$ be an $n \times n$ proper lower (respectively, upper) triangular matrix (with zeros on its diagonal). Let T_i denote the matrix obtained by zeroing all the columns of T except for its i-th (respectively, $(n+1-i)$-th) column, $i = 1, \ldots, n$. Then*

$$I + T = (I + T_1)(I + T_2) \cdots (I + T_{n-1}),$$

$$(I + T_i)^{-1} = I - T_i, \quad i = 1, \ldots, n-1.$$

Corollary A.1 *Under the assumptions of proposition 2, we have*

$$(I + T)^{-1} = (I - T_{n-1})(I - T_{n-2}) \cdots (I - T_1). \tag{A.1}$$

Based on corollary A.1, we may compute $\tilde{L}^{-1}$ and $\hat{U}^{-1}$ where $\hat{U}$ is a unit triangular matrix obtained from $\tilde{U}$ by scaling the rows and/or columns of $\tilde{U}$. The overall roundoff error estimate for computing $\tilde{L}^{-1}$ depends on the order in which we multiply the matrices $I - T_i$ in (A.1) for $T = \hat{L} - I$ (e.g. from left to right, from right to left or according to some mixed policy) and similarly for $T = \hat{U} - I$. (At the latter stage the variation also depends on the choice of the scaling of $\tilde{U}$, which defines the transition to the matrix $\hat{U}$.)

Appendix B. Numerical Experiments.

In this section, we compare the results of the numerical experiments performed in [PYS] with those based on algorithm 4.1 The comparison clearly favors the latter algorithm.

In the experiments, we compute numerically the determinants of $n \times n$ matrices A for $2 \leq n \leq 12$, based on computing the LU, PLU, and $PLUP_1$ factorizations of A. The input matrices A have been composed by using the following steps to produce random non-singular matrices either with determinants 1 or with relatively small known determinants:

1. For an auxiliary pair of lower and upper triangular matrices $L^{(0)}$ and $U^{(0)}$, respectively, let their non-zero off-diagonal entries be random integers in the interval $(-10, 10)$.

2. Either let the diagonal entries $l_{i,i}^{(0)}$ and $u_{i,i}^{(0)}$ be also chosen at random in the same way (see table 1) or set $l_{i,i}^{(0)} = u_{i,i}^{(0)} = 1$ for all i (see table 2). In the former case, if $l_{i,i}^{(0)} = 0$ or $u_{i,i}^{(0)} = 0$, for some i, $1 \leq i \geq n$, then set the entry to 1 to avoid arriving at a singular matrix.

3. Compute $A = L^{(0)}U^{(0)}$.

4. Swap a random pair of rows in the matrix A.

5. m times repeat step 4, where m is a random integer in the range $[0, n)$.

The algorithms have been implemented with C++ and built as a console application with Microsoft Visual C++ 5.0 compiler and linker. All numerical operations have been performed with double precision floating point arithmetic. The double precision representation of a number uses 64 bits: 1 for the sign, 11 for the exponent, and 52 for the mantissa. Its range is $\pm 1.7 \times 10^{308}$ with at least 15 decimal digits of precision. The test results have been collected on a Pentium-100MHz PC, running under Windows 95's DOS session. The system pseudo-random number generator functions **srand()** and **rand ()** have been used to generate input matrices.

Tables 1 and 2 present the results of our experiments. 1000 random matrices of each size (from 2 to 12 in the case of small determinants and from 2 to 10 in the case of determinants 1) have been tested. The relative errors $e_0 = \frac{|\det\ \tilde{A} - \det\ A'|}{|\det\ A'|}$, $e_{pys} = \frac{n^2 a^+ \epsilon}{|\det\ A'|}$ and $e_{new} = eN$ have been evaluated in each case, and their average values have been presented in the tables. The values of $|\det\ A|$ and $\frac{1}{||\tilde{A}^{-1}||} = \frac{1}{||\tilde{U}^{-1}\tilde{L}'^{-1}||}$ (average over the results of both approaches [PYS] and algorithm 4.1) have also been computed and presented in tables 1 and 2. Two integer counters V_{pys} and V_{new} have been used to keep track of the number of cases where the [PYS] algorithm and our new algorithm failed to verify the computation results, respectively. More specifically, whenever division by zero occured in the [PYS] algorithm or whenever we observed that $e_{pys} \geq 1$, the counter V_{pys} was incremented by one, and this case was excluded from the average count of the data represented in the first five columns of the tables. Likewise, whenever division by zero occured in algorithm 4.1 or whenever we observed that $e_{new} \geq 1$, the counter V_{new} was incremented by one and the case was similarly excluded from the average count of the data.

The tables show that the number of cases rejected by the new algorithm are consistently smaller than that of the [PYS] algorithm. For the accepted cases, the relative error estimates obtained by the new algorithm are much closer to but no smaller than the "true" relative errors.

Algorithm	e_0	e_{pys}	e_{new}	$\lvert\det\ A\rvert$	$\frac{1}{\lvert\lvert A^{-1}\rvert\rvert}$	NV_{pys}	NV_{new}
Size 2							
LU	1.958e-016	1.707e-013	5.906e-014	8.781e+002	9.036e+000	102	102
PLU	2.625e-016	4.454e-014	6.266e-014	8.845e+002	9.262e+000	3	3
$PLUP_1$	1.935e-016	3.189e-014	5.084e-014	8.834e+002	9.253e+000	0	0
Size 3							
LU	2.636e-015	2.322e-010	4.828e-012	2.400e+004	3.360e+000	76	72
PLU	2.404e-015	4.011e-011	1.705e-012	2.408e+004	3.438e+000	3	3
$PLUP_1$	2.184e-015	1.077e-011	1.160e-012	2.402e+004	3.441e+000	0	0
Size 4							
LU	2.265e-014	4.030e-004	3.705e-010	6.943e+005	1.483e+000	62	64
PLU	1.363e-014	1.756e-009	4.177e-011	6.991e+005	1.481e+000	1	1
$PLUP_1$	1.041e-014	1.678e-009	1.525e-011	6.984e+005	1.480e+000	0	0
Size 5							
LU	1.797e-013	3.935e-003	3.761e-009	1.563e+007	5.724e-001	56	40
PLU	8.941e-014	5.184e-006	2.116e-010	1.530e+007	5.652e-001	0	0
$PLUP_1$	7.952e-014	6.080e-007	1.437e-010	1.530e+007	5.652e-001	0	0
Size 6							
LU	9.590e-013	1.980e-002	1.255e-008	4.585e+008	2.615e-001	134	37
PLU	1.078e-012	5.420e-005	2.438e-009	4.097e+008	2.395e-001	0	0
$PLUP_1$	9.341e-013	1.005e-005	1.132e-009	4.097e+008	2.395e-001	0	0
Size 7							
LU	7.345e-013	5.393e-002	4.752e-008	3.433e+010	1.802e-001	317	37
PLU	9.060e-012	5.940e-003	3.327e-008	2.473e+010	1.423e-001	4	0
$PLUP_1$	1.046e-011	7.431e-004	1.502e-008	2.463e+010	1.417e-001	0	0
Size 8							
LU	4.674e-012	1.175e-001	1.751e-005	6.615e+011	8.039e-002	619	29
PLU	3.221e-011	2.448e-002	1.061e-006	5.662e+011	6.086e-002	37	0
$PLUP_1$	2.304e-011	1.404e-002	5.653e-008	5.541e+011	5.957e-002	16	0
Size 9							
LU	9.134e-013	2.053e-001	1.586e-008	2.701e+013	6.165e-002	839	19
PLU	1.055e-011	6.904e-002	5.593e-008	1.201e+013	2.812e-002	160	0
$PLUP_1$	1.356e-011	3.917e-002	6.251e-008	1.108e+013	2.596e-002	89	0
Size 10							
LU	9.726e-013	2.240e-001	7.454e-009	1.963e+015	6.443e-002	964	21
PLU	5.562e-012	1.304e-001	2.899e-008	8.986e+014	1.949e-002	425	0
$PLUP_1$	9.713e-012	1.013e-001	5.244e-008	7.089e+014	1.551e-002	271	0
Size 11							
LU	4.879e-013	4.320e-001	9.386e-009	5.658e+016	7.637e-002	995	32
PLU	4.061e-012	2.047e-001	1.454e-008	1.308e+016	1.431e-002	740	0
$PLUP_1$	1.578e-011	1.700e-001	5.002e-008	8.611e+015	9.995e-003	595	0
Size 12							
LU	NA	NA	NA	NA	NA	1000	24
PLU	6.190e-012	2.802e-001	5.254e-008	3.347e+018	3.118e-002	912	0
$PLUP_1$	4.520e-012	2.429e-001	2.767e-008	1.494e+018	1.581e-002	802	0

Table 1: Average estimated errors of 1,000 random matrices.

Algorithm	e_0	e_{pys}	e_{new}	$\|\det\ A\|$	$\frac{1}{\|\|A^{-1}\|\|}$	NV_{pys}	NV_{new}
Size 2							
LU	**7.904e-016**	**1.809e-012**	**1.656e-013**	**1.000e+000**	**1.000e+000**	**92**	**92**
PLU	**9.345e-016**	**1.036e-013**	**5.165e-013**	**1.000e+000**	**1.000e+000**	**2**	**2**
$PLUP_1$	**9.268e-016**	**1.034e-013**	**5.155e-013**	**1.000e+000**	**1.000e+000**	**0**	**0**
Size 3							
LU	**2.326e-014**	**2.800e-008**	**3.315e-010**	**1.000e+000**	**1.000e+000**	**85**	**76**
PLU	**3.583e-014**	**1.064e-009**	**4.950e-011**	**1.000e+000**	**1.000e+000**	**4**	**3**
$PLUP_1$	**2.828e-014**	**6.154e-010**	**3.819e-011**	**1.000e+000**	**1.000e+000**	**1**	**0**
Size 4							
LU	**1.004e-012**	**1.924e-003**	**5.165e-008**	**1.000e+000**	**1.000e+000**	**54**	**53**
PLU	**7.900e-013**	**1.285e-005**	**5.243e-009**	**1.000e+000**	**1.000e+000**	**0**	**0**
$PLUP_1$	**6.655e-013**	**4.342e-007**	**2.049e-009**	**1.000e+000**	**1.000e+000**	**0**	**0**
Size 5							
LU	**2.608e-011**	**3.990e-002**	**1.504e-005**	**1.000e+000**	**1.000e+000**	**152**	**52**
PLU	**2.507e-011**	**9.849e-004**	**3.504e-007**	**1.000e+000**	**1.000e+000**	**1**	**0**
$PLUP_1$	**2.143e-011**	**2.059e-004**	**1.021e-007**	**1.000e+000**	**1.000e+000**	**0**	**0**
Size 6							
LU	**2.809e-010**	**2.083e-001**	**3.227e-005**	**1.000e+000**	**1.000e+000**	**626**	**41**
PLU	**4.969e-010**	**1.319e-001**	**1.051e-005**	**1.000e+000**	**1.000e+000**	**104**	**0**
$PLUP_1$	**4.670e-010**	**8.723e-002**	**3.762e-006**	**1.000e+000**	**1.000e+000**	**21**	**0**
Size 7							
LU	**2.245e-009**	**3.712e-001**	**4.674e-004**	**1.000e+000**	**1.000e+000**	**992**	**41**
PLU	**4.464e-009**	**4.782e-001**	**1.818e-003**	**1.000e+000**	**1.000e+000**	**937**	**0**
$PLUP_1$	**2.534e-009**	**4.393e-001**	**3.227e-005**	**1.000e+000**	**1.000e+000**	**885**	**0**
Size 8							
LU	**NA**	**NA**	**NA**	**NA**	**NA**	**1000**	**114**
PLU	**NA**	**NA**	**NA**	**NA**	**NA**	**1000**	**8**
$PLUP_1$	**NA**	**NA**	**NA**	**NA**	**NA**	**1000**	**0**
Size 9							
LU	**NA**	**NA**	**NA**	**NA**	**NA**	**1000**	**349**
PLU	**NA**	**NA**	**NA**	**NA**	**NA**	**1000**	**54**
$PLUP_1$	**NA**	**NA**	**NA**	**NA**	**NA**	**1000**	**34**
Size 10							
LU	**NA**	**NA**	**NA**	**NA**	**NA**	**1000**	**713**
PLU	**NA**	**NA**	**NA**	**NA**	**NA**	**1000**	**281**
$PLUP_1$	**NA**	**NA**	**NA**	**NA**	**NA**	**1000**	**242**

Table 2: Average estimated errors of 1,000 random matrices with ± 1 determinants. (NA stands for "not available" in the case where computation of $\det A$ by the algorithm of [PYS] failed in all cases.)