# Certification of the QR Factor R and of Lattice Basis Reducedness

Gilles Villard*
CNRS, Laboratoire LIP (CNRS, ENSL, INRIA, UCBL)
École Normale Supérieure de Lyon, France
http://perso.ens-lyon.fr/gilles.villard

## ABSTRACT

Given a lattice basis of $n$ vectors in $\mathbb{Z}^n$, we propose an algorithm using $12n^3 + O(n^2)$ floating point operations for checking whether the basis is LLL-reduced. If the basis is reduced then the algorithm will hopefully answer "yes". If the basis is not reduced, or if the precision used is not sufficient with respect to $n$, and to the numerical properties of the basis, the algorithm will answer "failed". Hence a positive answer is a rigorous certificate. For implementing the certificate itself, we propose a floating point algorithm for computing (certified) error bounds for the $R$ factor of the $QR$ factorization. This algorithm takes into account all possible approximation and rounding errors. The certificate may be implemented using matrix library routines only. We report experiments that show that for a reduced basis of adequate dimension and quality the certificate succeeds, and establish the effectiveness of the certificate. This effectiveness is applied for certifying the output of fastest existing floating point heuristics of LLL reduction, without slowing down the whole process.

**Categories and Subject Descriptors:** I.1[**Symbolic and Algebraic Manipulation**]: Algorithms; F.2.1[**Analysis of Algorithms and Problem Complexity**]: Numerical Algorithms and Problems—*Computations on matrices.*

**General Terms:** Algorithms.

**Keywords:** linear algebra, QR factorization, lattice basis reducedness, verification algorithm.

## 1. INTRODUCTION

Our motivation is to develop a certificate for lattice basis reducedness that may be used in cooperation with—possibly non certified—numerical reduction heuristics such as those described in [22, 32, 34]. Indeed, floating-point approaches are crucial for reducing higher-dimensional lattice bases for example in cryptography [11, 13], cryptanalysis [2], or in computational group theory. A reducedness certificate is necessary when using reduction for proving lattice properties, or establishing mathematical results. In applications where only short vectors are expected however, a certificate may be added if the overhead is neglectible. Hence our two main constraints are speed and effectiveness. The certificate has to be fast enough for not slowing down the whole process, and the answer should be relevant ("yes") on a large class of inputs such as those successfully treated by the reduction heuristic. Our approach relies on error bounds for the $R$ factor of the $QR$ factorization that we discuss first.

**Bounding errors for the factor $R$.** Let $A$ be an $n \times n$ invertible integer matrix. The $QR$ factorization (see [12, Ch. 19]) of $A$ is a factorization $A = QR$ in which the factor $R \in \mathbb{R}^{n\times n}$ is an upper triangular matrix, and the factor $Q \in \mathbb{R}^{n\times n}$ is orthogonal ($Q^TQ$=I). We take the unique factorization such that the diagonal entries of $R$ are positive. Let $\mathbb{F}$ denote a set of floating point numbers such that the arithmetic operations in $\mathbb{F}$ satisfy the IEEE 754 arithmetic standard [1]. Assume that an approximate floating point and upper triangular factor $\widetilde{R} \in \mathbb{F}^{n\times n}$ is given. In §5 we propose an algorithm for computing a componentwise error bound for $|\widetilde{R} - R|$ using operations in $\mathbb{F}$ only. For a matrix $A = (a_{i,j})$, $|A|$ denotes $(|a_{i,j}|)$. Our error bound for $|\widetilde{R} - R|$ is given by a matrix $H \in \mathbb{F}^{n\times n}$ such that $|\widetilde{R} - R| \leq H|\widetilde{R}|$. Since floating point numbers are rational numbers, when $\widetilde{R}$ and $H$ are known, the latter inequality provides a rigourous mathematical bound for the error.

For understanding the behaviour of the error bounding algorithm better, we recall in §2 some existing numerical pertubation analyses for the $QR$ factorization. The necessary background material may be found in Higham's book [12]. Then in §3 and §4, we give the mathematical foundations of our approach. We focus on the componentwise bounds of [38] that allow us to derive an algorithm based on the principles of verification (self-validating) methods. On the latter methods we refer to the rich surveys of Rump [27, 28]. As various experiments in §5 and §6 will demonstrate, the error bounding algorithm is effective in practice. It provides relevant bounds for input matrices with appropriate numerical properties. The matrix $\widetilde{R}$ is computed by the modified Gram-Schmidt orthogonalization—MGS for short—(see [12, Alg. 19.12]). The cost of the certificate is only 5 times more than a numerical $QR$ factorization, we mean $10n^3 + O(n^2)$ operations in $\mathbb{F}$. For efficiency, the error bounds are them-

*This material is based on work supported in part by the French National Research Agency, ANR Gecko.
A more detailed text version [39] is available at:
http://hal.archives-ouvertes.fr/hal-00127059/en


*ISSAC'07,* July 29–August 1, 2007, Waterloo, Ontario, Canada.

selves calculated using floating point operations, nevertheless, they take into account all possible numerical and rounding errors. The reducedness certificate require $2n^3 + O(n^2)$ additional operations. Most of the $12n^3$ operations actually correspond to the evaluation of matrix expressions. An efficient implementation may thus rely on fast matrix routines such as the BLAS [8].

**The LLL-reducedness certificate.** The effectiveness of the error bound on $|\widetilde{R} - R|$ allows us to address the second topic of the paper. To an $n \times n$ integer matrix $A$ we associate the Euclidean lattice $\mathcal{L}$ generated by the columns $(a_j)$ of $A$ (for definitions and on algorithmic aspects of lattices we refer for instance to [6]). From $(a_j)$, the LLL algorithm computes a reduced basis [15], where the reduction is defined via the GS orthogonalization of $a_1, a_2, \ldots, a_n \in \mathbb{Z}^n$. The GS orthogonalization determines the associated orthogonal basis $a_1^*, a_2^*, \ldots, a_n^* \in \mathbb{Q}^n$ by induction, together with factors $\mu_{ij}$, using $a_i^* = a_i - \sum_{j=1}^{i-1} \mu_{ij} a_j^*$, and $\mu_{ij} = \langle a_i, a_j^* \rangle / \|a_j^*\|_2^2$, $1 \leq j < i$. Vectors $a_1, a_2, \ldots, a_n$ are said proper for $\eta \geq 1/2$ if their GS orthogonalization satisfies

$$|\mu_{ij}| \leq \eta, \ 1 \leq j < i \leq n. \tag{1}$$

In general one considers $\eta = 1/2$. The basis $a_1, a_2, \ldots, a_n$ of $\mathcal{L}$ is called LLL-reduced with factors $\delta$ and $\eta$ if the vectors are proper, and if they satisfy the Lovász conditions:

$$(\delta - \mu_{i+1,i}^2)\|a_i^*\|_2^2 \leq \|a_{i+1}^*\|_2^2, \ 1 \leq i \leq n-1, \tag{2}$$

with $1/4 < \delta \leq 1$ and $1/2 \leq \eta < \sqrt{\delta}$. If $A = QR$ is the $QR$ factorization of $A$ then we have

$$\begin{cases} \|a_i^*\|_2 = r_{ii}, \ 1 \leq i \leq n, \\ \mu_{ij} = r_{ji}/r_{jj}, \ 1 \leq j < i \leq n. \end{cases} \tag{3}$$

Assuming multiplication of $b$ bit integers within $O(b^2)$ bit operations, a basis of vectors of Euclidean length $2^\beta$ can be LLL reduced in $O(n^5\beta + n^4\beta^3)$ [31], or $O(n^6\beta + n^5\beta^2)$ (quadratic in fixed dimension) bit operations [21]. Relaxed notions of reducedness and assuming that integers are multiplied within $b^{1+o(1)}$ operations may lead to much better cost bounds such as $(n^{3.5}\beta^2)^{1+o(1)}$ [36] or $(n^3\beta(n+\beta))^{1+o(1)}$ bit operations [32].

We see from (3) that if an approximation $\widetilde{R}$ of $R$ with error bounds are known, then it may be possible to check whether (1) and (2) are satisfied. All the above draws the reducedness certificate that we propose in §6. We fix a set $\mathbb{F}$ of floating point numbers, and perform operations in $\mathbb{F}$ only. For certifying the reducedness of the column basis associated to $A$ the certificate works in three steps:

I: Numerical computation of a $R$ factor $\widetilde{R}$: $A \approx \widetilde{Q}\widetilde{R}$;

II: Certified computation of $F \in \mathbb{F}^{n\times n}$: $|\widetilde{R} - R| \leq F$;

III: Certified check of properness and Lovász conditions.

Following the principles of verification algorithms [28], Step I is purely approximation, and our implementation of Steps II and III is independent of the algorithm used for computing $\widetilde{R}$. For taking into account all possible numerical and rounding errors, Steps II and III use certified computing techniques (see §5).

In linear algebra, few things are known about the complexity of computing error bounds. A main result in [7] shows that the problem of computing a certified estimation of $\|A^{-1}\|$ (for a consistent norm) is as difficult as testing whether the product of two matrices is zero. Since with randomization the matrix product could be verified in $O(n^2)$ operations [9], the equivalence of the two problems is unclear. Verification methods have been developped in [25, 23, 30] for computing certified error bounds for linear system solution. In [23] the error bound (normwise) is computed in twice the time of Gaussian elimination. The verification approach [27, 28] gives an effective alternative to interval arithmetic whose exponential overestimation of the error would not be appropriate [28, §10.7]. In the same spirit, a verification approach using $O(n^3)$ floating point operations is proposed in [24] for the sign of the determinant (see [14] for a survey on the topic). We refer also to the verification of positive definiteness [29], or on eigenvalue verification [18, 26].

We will use several matrix norms (see [12, Ch. 6]) such as the Frobenius norm $\|\cdot\|_F$, the 2-norm $\|\cdot\|_2$, or the infinity norm $\|\cdot\|_\infty = \max_{1\leq i\leq n} \sum_{j=1}^n |a_{ij}|$. For $A = BC$ we have $\|A\|_\infty \leq \|B\|_\infty \|C\|_\infty$, and if $h = \|A\|_\infty$ then $|A| \leq H$ with $h_{ij} = h$. For a nonsingular matrix $A$, the matrix condition number is defined by $\kappa_p(A) = \|A\|_p \|A^{-1}\|_p$ with $p = 2, F$ or $\infty$ [12, Th. 6.4]. With the infinity norm we will also use the Bauer-Skeel condition number $\mathrm{cond}(A) = \||A^{-1}||A|\|_\infty \leq \kappa_\infty(A)$ [12, § 7.2]. Let $A^T$ and $A^{-T}$ denote the transpose matrices of $A$ and $A^{-1}$.

## 2. PERTURBATION ANALYSES FOR QR

A fixed precision computation of the $QR$ factorization leads to an approximate $\widetilde{R}$. The errors in $\widetilde{R}$ with respect to $R$ are called the *forward errors*. The matrix $\widetilde{R}$ is seen as the $QR$ factor of a perturbation $\widetilde{A} = A + E$, where $E$ is called the *backward error*. The link between backward and forward errors is made using the condition number of the problem, hence the condition number for the problem of computing $R$. The (relative) *condition number* of the problem measures the relative change in the output for a relative change in the input. A useful tool for estimating the accuracy of the solution to a problem, is the rule of thumb [12, p. 9]:

$$\text{forward error} \lesssim \text{condition number} \times \text{backward error}. \tag{4}$$

The condition number for $R$ may be defined theoretically, but it is non trivial to derive expressions that can be used in practice. Nevertheless, various formulae are proposed in the literature providing quantities that can be thought as a condition number for $R$, we refer for instance to [4]. These quantities may be very effective in practice in a matrix norm setting.

Let $A = QR$ and $A + E = \widetilde{A} = \widehat{Q}\widetilde{R}$ be $QR$ factorizations. Note that in general for a floating point factorization $A \approx \widetilde{Q}\widetilde{R}$, $\widetilde{Q}$ is not orthogonal hence $\widehat{Q} \neq \widetilde{Q}$. Let $\widetilde{R} = R + F$. For a sufficiently small backward error $E$, consider the normwise relative error $\epsilon = \|\widetilde{A} - A\|_F / \|A\|_2$. Then Sun's [37, Rem. 3.5] perturbation bounds (see also [35]) give

$$\|\widetilde{R} - R\|_F / \|R\|_2 \leq \sqrt{2}\kappa_2(A)\epsilon + O(\epsilon^2). \tag{5}$$

An improved bound is given by Zha [Theorem 2.1][40] (see also [4, § 5] and [12, §19.9]) under a componentwise model of perturbation that we simplify here. Let $|\widetilde{A} - A| = |E| = \epsilon|A|$, then for sufficiently small $\epsilon$ we have:

$$\|\widetilde{R} - R\|_\infty / \|R\|_\infty \leq c_n \mathrm{cond}(R^{-1})\epsilon + O(\epsilon^2) \tag{6}$$

where $c_n$ is a constant depending on $n$. Hence the Bauer-Skeel condition number of $R^{-1}$ can be considered as a condition number for the problem of calculating $R$. This indicates

that one may potentially loose significant digits (in the result) linearly with respect to the increase of $\log \mathrm{cond}(R^{-1})$, which is actually a typical behaviour in practice.

Identities (5) and (6) provide first order estimations of the errors. They could be extended for giving strict bounds on the forward error. Hence they are essential for estimating the normwise loss of accuracy. Nevertheless, the loss of accuracy on individual entries (needed for the certificate) may not be deduced from these identities. Matrix norms may largely overestimate the actual componentwise error in most cases. Normwise bounds much sharper than (5) and (6) may be found, especially in [5, 4]. It would be interesting to study how they could lead to practical componentwise bounds.

# 3. COMPONENTWISE BOUNDS FOR R

We now present the mathematical view and justification of the error bounding algorithm of §5. Given $A \in \mathbb{R}^{n\times n}$ invertible, and an upper triangular matrix $\widetilde{R} \in \mathbb{R}^{n\times n}$, we bound $|\widetilde{R} - R|$ where $R$ is the unknown $QR$ factor of $A$.

Rather than on the rule of thumb of previous section, our error bounding algorithm will be based on the componentwise bounds of Sun [38]. Our use of Sun's results is in the spirit of the verification methods. In particular, the error bounding algorithm is oblivious of the algorithm that is used for computing $\widetilde{R}$. Our bound computation may be appended to any numerical $QR$ algorithm, and does not rely on backward error bounds that would be have been needed in (4). Indeed, backward error bounds are known for specific $QR$ algorithms such as Householder or GS (see Theorems 19.4 and 19.13 in [12]), but may not be available in the general case. Note that the strict componentwise analysis of Sun [38, § 4] for $QR$ also relies on the backward error.

For being oblivious of the algorithm that has produced $\widetilde{R}$, we rather resort to Sun's study of the Cholesky factorization [38, § 4]. If $B \in \mathbb{R}^{n\times n}$ is symmetric positive definite, then there is a unique upper triangular $R \in \mathbb{R}^{n\times n}$ with positive diagonal entries, such that $B = R^T R$. This factorization is called the Cholesky factorization [12, Th. 10.1]. It holds that $A = QR$ is a $QR$ factorization if and only if $B = A^T A = R^T R$ is a Cholesky factorization. One has $\kappa_2(A^T A) = (\kappa_2(A))^2$, however we implement the certificate of §6 using $QR$ for computing $\widetilde{R}$, and use the Cholesky point of view only for computing the error bound.

For a matrix $A \in \mathbb{R}^{n\times n}$, the spectral radius $\rho(A)$ is the maximum of the modules of the eigenvalues of $A$. We denote by $\mathrm{triu}(A)$ the upper triangular part of $A$, we mean that $\mathrm{triu}(A) = (t_{ij})$ with $t_{ij} = a_{ij}$ if $i \le j$, and $t_{ij} = 0$ otherwise. The following theorem is [38, Th. 2.1].

THEOREM 3.1. *For $B, \widetilde{B} \in \mathbb{R}^{n\times n}$ symmetric positive definite matrices, let $R$ and $\widetilde{R}$ be the Cholesky factors of $B$ and $\widetilde{B}$. Let $E = \widetilde{B} - B$, and $G = |\widetilde{R}^{-T} E \widetilde{R}^{-1}|$. Then if $\rho(G) < 1$ we have $|\widetilde{R} - R| \le \mathrm{triu}(G(I-G)^{-1})|\widetilde{R}|$.*

Let us apply Theorem 3.1 with $B = A^T A$ and $\widetilde{B} = \widetilde{A}^T \widetilde{A}$. Using $\widetilde{A} = \widehat{Q}\widetilde{R}$ and $\widehat{Q}^T\widehat{Q} = I$, we get:

$$G = |\widetilde{R}^{-T}(\widetilde{A}^T\widetilde{A} - A^T A)\widetilde{R}^{-1}| = |\widetilde{R}^{-T} A^T A \widetilde{R}^{-1} - I|.$$

Going back to the $R$ factor of the $QR$ factorization we then have the following corollary to Theorem 3.1

THEOREM 3.2. *For $A \in \mathbb{R}^{n\times n}$ an invertible matrix, let $R$ be the $QR$ factor of $A$. Let $\widetilde{R} \in \mathbb{R}^{n\times n}$ be upper triangular and invertible, and $G = |\widetilde{R}^{-T} A^T A \widetilde{R}^{-1} - I|$. Then if $\rho(G) < 1$, we have*

$$|\widetilde{R} - R| \le \mathrm{triu}(G(I-G)^{-1})|\widetilde{R}|. \tag{7}$$

PROOF. Since $\widetilde{R}$ is invertible, $\widetilde{B} = \widetilde{R}^T\widetilde{R}$ is positive definite, the same holds for $B = A^T A$. By construction $R$ and $\widetilde{R}$ are the Cholesky factors of $B$ and $\widetilde{B}$. It suffices to apply Theorem 3.1 for concluding. □

Few things are known about the (mathematical) quality of bound (7) over $\mathbb{R}$. Furthermore, both additional method (for bounding $\widetilde{R}^{-1}$ and $\mathrm{triu}(G(I-G)^{-1})$) and arithmetic errors will be introduced for the finite precision evaluation of the bound. We produce an error bounding algorithm that is not fully analyzed, the experiments of §5 will however give a precise idea of its practical behaviour and effectiveness. For illustrating bound (7) over $\mathbb{R}$, let us consider some examples.

The calculations have been done in Maple [19], either exactly or with high precision. Let $H = \mathrm{triu}(G(I-G)^{-1})$ such that (7) is $|\widetilde{R} - R| \le H|\widetilde{R}|$. On random matrices (`randsvd` type [12, Ch. 28], $n = 200$), with $\widetilde{R}$ computed with double precision floating point numbers (64 bit numbers) via the MGS algorithm, we typically get the following. If $A$ with $\mathrm{cond}(R^{-1}) \approx 10^5$ then $\|H\|_\infty \approx 2\cdot 10^{-9}$. This leads to the knowledge that $\widetilde{R}$ approximates $R$ with (relative) accuracy $\approx 10^{-10}$ ($10^{-13}$ for the diagonal entries that play a key role for the Lovász test). If $\mathrm{cond}(R^{-1}) \approx 4\cdot 10^{13}$ then $\|H\|_\infty \approx 3\cdot 10^{-3}$, and $R$ is known with accuracy of about $10^{-2}$ ($2\cdot 10^{-5}$ on the diagonal). The ratio between the estimation and the true error is less than 4 on the diagonal. Consider also the matrix quoted from [4, Eq. 5.4]:

$$A_1 = \begin{bmatrix} 1 & 1-10^{-10} \\ 1 & 1+10^{-10} \end{bmatrix},$$

with $\mathrm{cond}(R^{-1}) \approx 2\cdot 10^{10}$. We compute the matrix $\widetilde{R}$ in Matlab [17], and obtain over $\mathbb{R}$ the error bound:

$$|\widetilde{R} - R| \le \begin{bmatrix} 3.5\cdot 10^{-12} & 3.5\cdot 10^{-12} \\ 0 & 7.4\cdot 10^{-17} \end{bmatrix}. \tag{8}$$

The matrix $R$ is known with accuracy of about $2.5\cdot 10^{-12}$ on the first row, and $5.25\cdot 10^{-7}$ for $r_{22}$. On the first row the error is overestimated by a factor of about $3.6\cdot 10^4$. Notwithstanding the fact that the accuracy of the bound produced by Theorem 3.1 is penalized by the particular form of the matrix, the estimation of the accuracy of $\widetilde{R}$ remains very good. We refer to [39] for some additional examples.

# 4. TOWARD AN IMPLEMENTATION

Theorem 3.2 is the foundation of our error bounding algorithm. It involves several quantities that need further study before deriving an implementation in §5. We decompose the computation of the bound on $|\widetilde{R} - R|$ into four main steps. We recall that at this point, only $A$ and $\widetilde{R}$ are known.

**Step 1. Invertibility check of $\widetilde{R}$.** For dealing with $\widetilde{R}^{-1}$ in a certified way, which is a non trivial question in fixed precision, we use the verification solution of Oishi and Rump [23]. We compute a purely numerical approximate inverse $V \approx \widetilde{R}^{-1}$ (by numerical triangular inversion). Then we know from [23] that $\widetilde{R}$ is invertible if

$$\|\widetilde{R}V - I\|_\infty < 1. \tag{9}$$

**Step 2. Bounding $G$.** For bounding $G$, we are also inspired by [23] and introduce $W = \widetilde{R}V$ ($\approx I$). We have

$$\begin{aligned} G &= |\widetilde{R}^{-T}A^TA\widetilde{R}^{-1} - I| \\ &= |W^{-T}(V^TA^TAV - W^TW)W^{-1})| \\ &\leq |W^{-T}| \cdot |V^TA^TAV - W^TW| \cdot |W^{-1}|. \end{aligned}$$

In the inequality above, if $\widetilde{R}$ is close to $R$ and $V$ is close to $\widetilde{R}^{-1}$, then both $V^TA^TAV$ and $W^TW$ are close to identity. Hence it is natural to pursue with:

$$G \leq |W^{-T}| \cdot |(V^TA^TAV - I) - (W^TW - I)| \cdot |W^{-1}|$$

which gives

$$G \leq |W^{-T}| \cdot (|(V^TA^TAV - I)| + |(W^TW - I)|) \cdot |W^{-1}|. \quad (10)$$

We will use (10) for computing a certified bound for $G$. The products involving $A$, $V$, and $W = \widetilde{R}V$ will be bounded directly by interval techniques. It remains to bound $|W^{-1}|$. We expect $W$ to be close to $I$, and may use a specific approximation. We have $|W^{-1}| = |(I - (I - W))^{-1}|$ (see [23, Introduction]). Then, when $\widetilde{R}$ is invertible,

$$\begin{aligned} |W^{-1}| &= |I + (I - W) + (I - W)^2 + \dots| \\ &\leq |2I - W| + |(I - W)^2| \cdot |I + (I - W) + \dots|. \end{aligned}$$

Using that the entries of $|I-W|^2 \cdot |I+(I-W)+(I-W)^2+\dots|$ are bounded by the infinity norm, and since $W$ is triangular, it follows that

$$|W^{-1}| \leq |2I - W| + \mathcal{T}\left(\frac{\|I - W\|^2}{1 - \|I - W\|_\infty}\right) \quad (11)$$

where, for $x \in \mathbb{R}$, $\mathcal{T}(x)$ denotes the upper triangular matrix whose entries on the diagonal and above are equal to $x$. Note that the invertibility check (9) ensures that $1 - \|I - W\|_\infty > 0$. The absolute value $|W^{-1}|$ could have been bounded directly using $1/(1 - \|I - W\|_\infty)$, but introducing the infinity norm only in the second order terms leads to a much better bound in our experiments.

The matrix manipulations we have done for obtaining (10) and (11) follow some keys to the design of verification methods. We especially refer to [28, p. 211] where the introduction of small factors is recommended. We have introduced the matrices $V^TA^TAV - I$ and $W^TW - I$ whose absolute bounds are expected to be small when $\widetilde{R} \approx R$ and $W \approx I$. On the other hand, in (11), $|2I - W|$ is expected to be close to $I$, and remaining terms are second order terms (see also the analysis for $\alpha$ in [23, §5]).

**Step 3. Bounding the spectral radius of $G$.** For any consistent matrix norm we have $\rho(A) \leq \|A\|$. With the above bound on $G$, we will simply test whether

$$\|G\|_\infty < 1 \quad (12)$$

for asserting that $\rho(G) < 1$ in Theorem 3.2. This test could certainly be sharpened in future versions of the certificate.

**Step 4. Bounding $|\widetilde{R} - R|$.** Once a bound on $G$ is known it remains to bound $H = \mathrm{triu}(G(I - G)^{-1})$. Similarly to (11) we will use:

$$H \leq \mathrm{triu}(G) + \mathcal{T}\left(\frac{\|G\|_\infty^2}{1 - \|G\|_\infty}\right). \quad (13)$$

Note that using the spectral radius check (12) ensures that $1 - \|G\|_\infty > 0$.

# 5. ERROR BOUNDING ALGORITHM

Let $\mathbb{F}$ be a set of floating point numbers such that the arithmetic operations in $\mathbb{F}$ satisfy the IEEE 754 standard. $A$ and $\widetilde{R}$ are now matrices in $\mathbb{F}^{n\times n}$ (in general, the entries of $R$ are not in $\mathbb{F}$). We carry the four steps of §4 over to $\mathbb{F}$ for bounding the error. The checks (9) and (12), and inequalities (10), (11), and (13), only involve matrix multiplications, additions, subtractions, and divisions by a scalar.

**Certified bounds for matrix expressions.** We denote by $\mathrm{fl}(x)$ the value of an arithmetic expression $x$ computed by arithmetic in $\mathbb{F}$. For instance, for $a, b \in \mathbb{F}$, $\mathrm{fl}(a+b/c)$ denotes the result in $\mathbb{F}$ with the addition and division performed in $\mathbb{F}$. In the text, an arithmetic expression on floating point numbers denotes the exact value in $\mathbb{R}$. For instance $a + b \in \mathbb{R}$ is the result of the addition in $\mathbb{R}$. The abolute value, the max, and the negation are exact operations: for $a, b \in \mathbb{F}$, $\mathrm{fl}(|a|) = |a|$, $\mathrm{fl}(\max\{a, b\}) = \max\{a, b\}$, $\mathrm{fl}(-a) = -a$. We can use the possibility of changing the rounding mode. We essentially follow Rump's approach [28], and Oishi & Rump [23]. We use the statements "round(down)" and "round(up)"[1], all operations after such a statement are rounded downwards or upwards, respectively, until the next call to round. For $a, b \in \mathbb{F}$, a bound $r$ on $|a \ op \ b|$ for $op \in \{+, -, \times, \div\}$ may be computed as follows. The program

$$\begin{array}{ll} \text{round(down);} & \underline{r} = \mathrm{fl}(a \ op \ b) \\ \text{round(up);} & \overline{r} = \mathrm{fl}(a \ op \ b); \quad r = \max\{|\underline{r}|, |\overline{r}|\} \end{array} \quad (14)$$

leads to $\underline{r}$ and $\overline{r}$ such that $\underline{r} \leq a \ op \ b \leq \overline{r}$, and to $r \in \mathbb{F}$ such that $|a \ op \ b| \leq r$. The IEEE standard ensures that $\underline{r}$ and $\overline{r}$ are the best possible bounds in $\mathbb{F}$. This may be extended to the matrix operation $AB - C$ with $A, B, C \in \mathbb{F}^{n\times n}$. If $AB$ is implemented using only additions and multiplications, then the program

$$\begin{array}{ll} \text{round(down);} & \underline{R} = \mathrm{fl}(AB - C) \\ \text{round(up);} & \overline{R} = \mathrm{fl}(AB - C); \quad R = \max\{|\underline{R}|, |\overline{R}|\} \end{array} \quad (15)$$

where the maximum is taken componentwise, provides $\underline{R} \leq AB - C \leq \overline{R}$, and $R \in \mathbb{F}^{n\times n}$ such that $|AB - C| \leq R$. For bounding more general matrix expressions we will use a midpoint-radius matrix representation [28, §10.9]. Assume that $M$ and $N$ are two matrices known to be in intervals $[\underline{M}, \overline{M}]$ and $[\underline{N}, \overline{N}]$, respectively. Then the program [28, Fig. 10.22]

$$\begin{array}{ll} \text{round(up);} & \mathrm{m}_M = \mathrm{fl}((\overline{M} + \underline{M})/2); \quad \mathrm{r}_M = \mathrm{fl}(\mathrm{m}_M - \underline{M}) \\ & \mathrm{m}_N = \mathrm{fl}((\overline{N} + \underline{N})/2); \quad \mathrm{r}_N = \mathrm{fl}(\mathrm{m}_N - \underline{N}) \\ \text{round(down);} & \underline{R} = \mathrm{fl}(\mathrm{m}_M \times \mathrm{m}_N - I) \\ \text{round(up);} & \overline{R} = \mathrm{fl}(\mathrm{m}_M \times \mathrm{m}_N - I) \\ & R = \mathrm{fl}(\max\{|\underline{R}|, |\overline{R}|\} + |\mathrm{m}_M| \times \mathrm{r}_N \\ & \qquad + \mathrm{r}_M \times (|\mathrm{m}_N| + \mathrm{r}_N)) \end{array}$$

computes $R$ such that $|M \times N - I| \leq R$. The two latter programs allow to use fast matrix routines such as the BLAS ones [28, §10.9]). Other matrix operations that we perform are additions, products, divisions by scalars, and infinity norms for matrices with positive entries. With no subtraction certified bounds can be computed with directed rounding using (14). For upper bounds on divisions by a floating point number $1-g$, we first compute an upper bound for $-(g - 1)$. Other approaches for certified matrix computations could be considered. We refer to Rump [28] for a

[1] `fesetround(FE_DOWNWARD)` and `fesetround(FE_UPWARD)` in C language.

general discussion on this topic, and for the efficiency of the approach chosen here. Using above certified techniques, the four computational steps of §4 are implemented directly. We obtain the following operation count.

THEOREM 5.1. *Let $A \in \mathbb{F}^{n\times n}$, and $\widetilde{R} \in \mathbb{F}^{n\times n}$ upper triangular be given. The error bounding algorithm computes a matrix $F \in \mathbb{F}^{n\times n}$ such that $|\widetilde{R} - R| \leq F$, where $R$ is the unknown $QR$ factor of $A$, in $10n^3 + O(n^2)$ floating point operations.*

We omit the proof here (see [39, §6.2]) which consists in elementary dense and triangular matrix manipulations. A $QR$ factorization typically costs $2n^3+O(n^2)$ (GS or Householder approaches). Hence we are able to compute a certified error bound $|\widetilde{R}-R|$ at the cost of only five approximate factorizations. We have implemented the algorithm in C. The error bounding program takes as input two floating point matrices $A$ and $\widetilde{R}$ and always returns a matrix $F$. The entries of $F$ are finite floating numbers if the program is able to certify that $\widetilde{R}$ is invertible, that the spectral radius of $G$ is less than one, and if no overflow is produced. Otherwise, the entries of $F$ may be equal to infinity.

**Computational results.** Our results correspond to the application of Theorem 5.1 with double precision floating point numbers. Here and in §6 the condition numbers and the "true errors" have been computed with high precision using Mpfr [20]. We study the behaviour of the certified error bound by looking at its value and accuracy (with respect to the true error), especially when the dimension and the condition number increase. We mainly focus on the exponent $k$ such that relative error is in $10^{-k}$, $k$ expresses the number of significant decimal digits we certify for $\widetilde{R}$.

For the matrix $A_1$ of Section 3, and $\widetilde{R}$ from Matlab, we compute the bound

$$|\widetilde{R} - R| \leq \begin{bmatrix} 6.7\cdot 10^{-11} & 6.7\cdot 10^{-11} \\ 0 & 5\cdot 10^{-16} \end{bmatrix}.$$

Comparing to (8), we see that the finite precision estimator we propose is only slightly overestimating the best bound that could be obtained by the method.

For next results, $\widetilde{R}$ is computed with the MGS algorithm using double precision as for the estimator. Our tests use ten matrix samples. We first illustrate the *value of the certified bound with respect to the dimension.*

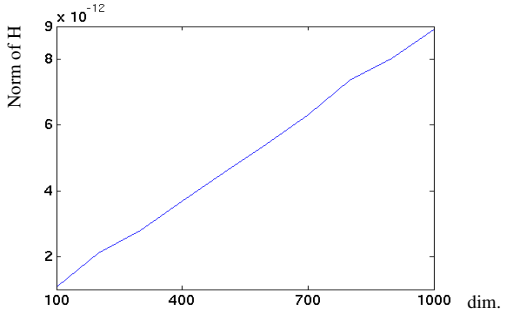


Figure 5.1: Certified $\|H\|_\infty$ for random $A$, $\kappa_2(A) \approx 10^3$.

Figures 5.1 and 5.2 are for random input matrices $A$ (`randsvd` type [12, Ch. 28]). We keep the condition number almost constant when the dimension increase. We draw the infinity norm of $H$ such that $|\widetilde{R} - R| \leq H|\widetilde{R}|$, and the certified maximum relative error on the diagonal, we mean $\max_i |\widetilde{r}_{ii} - r_{ii}|/|\widetilde{r}_{ii}|$. We see that $\|H\|_\infty$ increases linearly with $n$. The loss of accuracy on the diagonal is approximately quadratic in $n$ (we use a logarithmic scale for the $y$ axis on Figure 5.2).

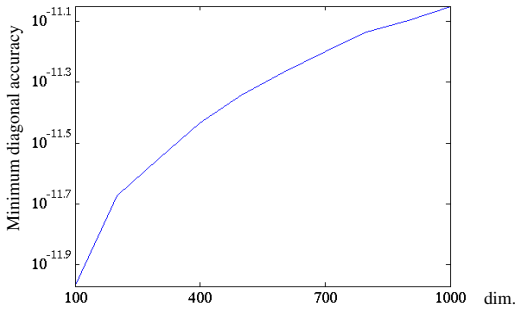


Figure 5.2: Certified maximum relative error on $R$ for random $A$, $\kappa_2(A) \approx 10^3$ ($y$ axis with logarithmic scale).

Such small increase rates—that are typical for numerical algorithm forward errors themselves—demonstrate a first aspect of the effectiveness of our finite precision bounds. The certified general maximum error $\max_{ij} |\widetilde{r}_{ij} - r_{ij}|/|\widetilde{r}_{ij}|$ increases faster. It typically grows from $10^{-7}$ to $10^{-5}$ for the dimensions considered here. We need further investigation especially for a better understanding of the influence of the product $H|\widetilde{R}|$, and of the magnitudes in $R$.

We discuss next the *accuracy of the certified bound with respect to the exact error* (not the quality of the $QR$ algorithm itself). In addition to `randsvd` matrices we also consider random integer matrices with entries of absolute values less than $10^3$. The condition numbers $\kappa_\infty(A)$ are varying from about $10^4$ to $10^6$. On dimension 1500, the maximum exact relative error on $R$ has order $10^{-10}$ to $10^{-9}$. We certify this error by returning an error bound of order $10^{-6}$ to $10^{-5}$. *With respect to the dimension*, we observe that the fast certified bound overestimates the componentwise error by a factor of order of about $10^3$ for $n = 200$ to $10^5$ for $n = 1500$. Restricted to the diagonal entries, the overestimation goes from about $10^2$ to less than $10^4$. This shows that even with condition numbers and dimensions that can be here quite large, we are able to certify at least four or five significant decimal digits for every entries of $R$, and at least 9 digits on the diagonal. On matrices with small condition number (Matlab `gallery('orthog')` [12, Chapter 28]) the quality of the certified bound may be remarkably good and stable. For dimensions between 60 and 500, and $\mathrm{cond}(R^{-1}) \approx 3$ ($\kappa_\infty \leq 200$), we most of the time obtain an overestimation between 15 and 22 (and more than 12 certified significant decimal digits in $\widetilde{R}$).

We may now ask the question of the sensitivity of the *quality of the certified error bound with respect to the condition number of the input matrix.*

| $n$ | 30 | 40 | 50 | 60 | 70 |
|---|---|---|---|---|---|
| $\kappa_\infty(A)$ | $1.1\cdot 10^6$ | $7.8\cdot 10^7$ | $4.8\cdot 10^9$ | $2.8\cdot 10^{11}$ | $1.5\cdot 10^{13}$ |
| $q_{\mathrm{err}}$ | 281 | 161 | 103 | 140 | 152 |
| Dig. | 10 | 9 | 7 | 5 | 4 |

Figure 5.3: Quality of the bound for Kahan matrices.

We first report that the quality maybe be very good even for matrices with high condition number. For Figure 5.3 we use $A = QA_K \in \mathbb{F}^{n\times n}$. The matrices $Q$ are random orthogonal from the Matlab `gallery` function [12, Chapter 28]. The matrices $A_K$ are Kahan upper triangular matrices with $a_{ii} = (\sin\theta)^{i-1}$, $a_{ij} = -(\sin\theta)^{i-1}\cos\theta$ for $j > i$, and $\theta = 1.2$. We give the ratio $q_{\text{err}}$ of the certified relative error bound and the true error (max), and the number of significant decimal digits certified in $\widetilde{R}$.

In general, the quality of the bound may depend on the condition number. Consider for instance the relative error ratio $q_{\text{err}}$ for several small matrices ($n = 10$). For a Chebyshev Vandermonde-like (nearly orthogonal, $\kappa_\infty \approx 13$), $q_{\text{err}} \approx 11$. We have $q_{\text{err}} \approx 14$ for Toeplitz and symmetric positive definite matrices ($\kappa_\infty \approx 700$). On the Pascal matrix ($\kappa_\infty \approx 8 \cdot 10^9$) we get $q_{\text{err}} \approx 25$, and about 1600 for the Hilbert matrix ($\kappa_\infty \approx 3.5 \cdot 10^{13}$). Figure 5.4 is more general. The overestimation of certified error bound seems to increase quite slowly with the condition number.

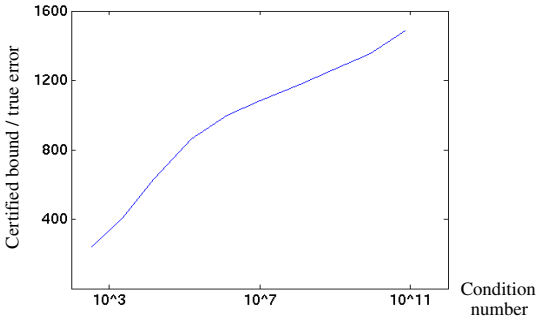


Figure 5.4: Error ratio $q_{\text{err}}$ with respect to $\kappa_\infty(A)$ on `randsvd` matrices of dimension $n = 200$.

We see that the conditions in which we return finite bounds are clearly linked with the numerical properties of $A$. Let us give two examples for the impossibility to certify the spectral radius using (12). We return finite bounds for the error for the Pascal matrix of dimension 14 ($\kappa_\infty \approx 3.8 \cdot 10^{14}$, $\|G\|_\infty \approx 0.06$). For $n = 15$ the algorithm produces infinity bounds. On `randsvd` matrices of dimension 40, the algorithm is effective until $\kappa_\infty \approx 3 \cdot 10^{14}$ with $\|G\|_\infty \approx 0.9$. Note that in double precision, with relative rounding unit $2^{-53}$ (the backward error is larger in general), and for a relative forward error less than 1, the rule of thumb (4) advocates for a condition number less than $10^{16}$.

The certified bound is computed with finite precision, hence inherently, it overestimates the true error. However, for realistic dimensions and condition numbers (with respect to the precision), the overestimation is mastered. It follows that in general, many significant digits are certified in the approximate $QR$ factor $\widetilde{R}$. The latter is a key to the application of the fast bound to the reducedness certificate.

# 6. LLL REDUCEDNESS CERTIFICATE

To $A \in \mathbb{Z}^{n\times n}$ we associate the Euclidean lattice $\mathcal{L}$ generated by the columns $(a_j)$ of $A$. About lattices the reader may refer for instance to [6]. Since the seminal Lenstra-Lenstra-Lovász algorithm [15], the lattice basis reduction problem receives much attention. In particular, floating-point variants that lead to very fast reduction approaches have been invented. See the work of Nguyen and Stehlé [21, 34], of Schnorr [32], and references therein. Most of floating point variants lead to powerful heuristics, especially à la Schnorr-Euchner [33], that are implemented in most of computer algebra and number theory systems. Our aim here is not to study the basis reduction itself. We focus on the reducedness. Indeed, a fast heuristic may not certify that the output basis is reduced (still working very well), and it is worthwhile to study the problem of checking *a posteriori* whether a given basis is reduced or not. The notion of reduction we consider is the LLL reduction [15].

We propose here an algorithm that takes as input an invertible matrix $A \in \mathbb{Z}^{n\times n}$, and tests the LLL reducedness of the basis formed by the columns of $A$. We have seen that this consists in testing the two conditions (1) and (2). Let $R$ be the $QR$ factor of $A$. If the $a_j$ are proper, we mean

$$|r_{i,j}|/r_{i,i} \le \eta,\ 1 \le i < j \le n, \tag{16}$$

and if the Lovász conditions

$$\sqrt{\delta - (r_{i,i+1}/r_{i,i})^2}\ r_{i,i} \le r_{i+1,i+1},\ 1 \le i \le n-1, \tag{17}$$

are satisfied, then the basis $a_1, \ldots, a_n$ of $\mathcal{L}$ is called LLL reduced with parameters $(\delta, \eta)$. The latter holds for $1/4 < \delta \le 1$ and $1/2 \le \eta < \sqrt{\delta}$. The principle of the algorithm is to compute an approximate $\widetilde{R}$ together with error bounds (using the floating point algorithm of §5), then to test (16) and (17). The integer entries of $A$ may not be in $\mathbb{F}$ (we use Gmp [10]), nevertheless, for computing $\widetilde{R}$ we may take $\widetilde{A}$ by direct conversion to $\mathbb{F}$. Since $\widetilde{R}$ will be an approximation anyway, this does not really influence the quality of subsequent computations. Then $\widetilde{R}$ is computed by the MGS algorithm, and we apply Theorem 5.1 for a certified error bound. The only expression where $A$ is involved is (10), where the computation of $AV$ using (15) is needed. The problem of conversion to $\mathbb{F}$ is solved here by rounding upwards and downwards during the conversion integer to floating point. We mean that we introduce a small interval such that $A \in [A_-, A_+]$ with $A_-, A_+ \in \mathbb{F}^{n\times n}$, and we evaluate $A_-V$ and $A_+V$ in (15). Therefore the error bound $F \in \mathbb{F}^{n\times n}$ we compute by Theorem 5.1 is actually such that $|R - \widetilde{R}| \le F$ for $R$ the $QR$ factor of any $A \in [A_-, A_+]$. Once $F$ is known, for fixed $i$ and $j$, we test (16) by resorting to the bounding techniques of §5:

$$\begin{array}{ll} \text{round(down);} & \underline{\eta} = \text{fl}(\eta);\quad t_i = \text{fl}((r_{i,i} - f_{i,i}) \times \underline{\eta}) \\ \text{round(up);} & t_j = \text{fl}(|r_{i,j}| + f_{i,j}) \\ & \text{test } t_j \le t_i? \end{array}$$

with temporary variables $t_i$ and $t_j$. Recall that the diagonal entries of $R$ are positive. Similarly, for a fixed $i$, we test (17) using:

$$\begin{array}{ll} \text{round(up);} & t_i = \text{fl}(r_{i,i} + f_{i,i});\ \overline{\delta} = \text{fl}(\delta); \\ \text{round(down);} & t_{i+1} = \text{fl}(r_{i+1,i+1} - f_{i+1,i+1}) \\ & t = -\left(\text{fl}(((|r_{i,i+1}| - f_{i,i+1})/t_i)^2) - \delta\right); \\ \text{round(up);} & t = \text{fl}(\sqrt{t} \times t_i) \\ & \text{test } t \le t_{i+1}? \end{array} \tag{18}$$

with temporary variables $t$ and $t_i$. In practice, for minimizing the cost induced by the changes of rounding mode, loops are put between the round instructions. In addition to the $10n^3 + O(n^2)$ operations for computing $F$ using Theorem 5.1, the reducedness test requires $2n^3 + O(n^2)$ operations for computing an approximate factor $\widetilde{R}$.

THEOREM 6.1. *Let $A \in \mathbb{Z}^{n\times n}$ invertible and parameters $(\delta, \eta)$ be given. The reducedness certificate certifies in $12n^3 +$*

*$O(n^2)$ floating point operations that the column lattice of $A$ is LLL reduced with parameters $(\delta, \eta)$, or returns "failed".*

Note that the rectangular case $A \in \mathbb{Z}^{m \times n}$ should not differ in a significant way. The ingredients we use (especially Section 3) carry over to the case $m > n$.

The reducedness is certified when the error bound computed for $|\widetilde{R} - R|$ is finite, when no overflow or underflow occur during the test, and when the basis is reduced. The cost of the certificate is roughly the one of six floating point $QR$ factorizations. Therefore in general, the reducedness test should be much faster than the reduction process itself (see the cost bounds in the introduction), and may be appended to any reduction heuristic program.

**Computational results.** As previously we use double precision floating point numbers. Figures 6.1-6.3 show certified bounds that we obtain for errors and quantities involved in the reducedness test. We have manipulated lattices using Magma [16], the LLL reduction implementation is based on the work of Nguyen and Stehlé [21, 34].

The first family of reduced bases we consider are obtained by the reduction of random integer matrices. The bases are reduced for the classical LLL parameters $(\delta, \eta) = (3/4, 1/2)$ in Figure 6.1, and $(\delta, \eta) = (0.99, 0.5001)$ for a stronger reduction in Figure 6.2. Since the numerical quality of the tested bases is good ($\kappa_\infty(A) \leq 10^6$), the reducedness certificate is highly efficient. The certified error is very small, and hence the tests are passed except in exceptional cases. We look at the smallest difference $\text{diff}_{\min} = t_k - t = \min_i\{t_{i+1} - t\}$ whose positiveness has to be certified in (18). The certificate has lots of room since the absolute errors on $t$ and $t_k = \|a_k^*\|_2$ are much smaller.

| $n$ | 40 | 200 | 500 | 1000 |
|---|---|---|---|---|
| $\kappa_\infty(A)$ | $4.7 \cdot 10^2$ | $2.4 \cdot 10^4$ | $1.8 \cdot 10^5$ | $9 \cdot 10^5$ |
| $\text{diff}_{\min}$ | 18 | 10 | 13 | 23 |
| Abs. err. on $\|a_k^*\|_2$ | $7.5 \cdot 10^{-12}$ | $3 \cdot 10^{-10}$ | $1.5 \cdot 10^{-9}$ | $1.2 \cdot 10^{-8}$ |
| $\max_{ij} \mu_{ij}$ | 0.4997 | 0.499994 | 0.49991 | 0.49999 |
| Rel. err. on $\|r_{ij}\|$ | $2.8 \cdot 10^{-11}$ | $8.6 \cdot 10^{-9}$ | $1.5 \cdot 10^{-7}$ | $3 \cdot 10^{-5}$ |

Figure 6.1: On $(3/4, 1/2)$-reduced bases from random integer matrices with entries on $10^3$ bits, $\max|a_{ij}| \leq 1000$.

Exceptional cases will rather occur when testing properness. Indeed, testing reducedness may be an ill-posed problem because of the possible equalities in (16) and (17). An ill-posed case with say $\eta = 1/2$, is for example a reduced basis with $\mu_{ij} = 1/2$ for some $i, j$. Therefore the algorithm will rather be used for certifying that a $(\delta, \eta)$-reduced basis is a $(\delta - \epsilon_1, \eta + \epsilon_2)$-reduced basis for small $\epsilon_1, \epsilon_2$. The latter does not really affect the relevant certified informations provided by the reduction.

| $n$ | 40 | 200 | 500 | 1000 |
|---|---|---|---|---|
| $\text{diff}_{\min}$ | $4.8 \cdot 10^{-2}$ | $7.7 \cdot 10^{-2}$ | $5.3 \cdot 10^{-2}$ | $7.3 \cdot 10^{-2}$ |
| Abs. err. on $\|a_k^*\|_2$ | $9.4 \cdot 10^{-14}$ | $6 \cdot 10^{-12}$ | $4 \cdot 10^{-11}$ | $2 \cdot 10^{-10}$ |

Figure 6.2: On $(0.99, 0.501)$-reduced bases from random integer matrices with entries on 10 bits, $\max|a_{ij}| \leq 10$.

A second type of reduced bases on which we have run the certificate comes from the problem of computing a good floating point coefficient polynomial approximation to a function [3]. Reduced bases with parameters $(3/4, 1/2)$ may have integer entries as large as $10^{80}$. The certificate has always succeeded. With $n = 18$ and with $\kappa_\infty(A) \approx 4 \cdot 10^{12}$, the smallest difference $\text{diff}_{\min} = t - t_k$ has been around $2.4 \cdot 10^{76}$ with certified absolute error $1.95 \cdot 10^{62}$. The maximum of the $\mu_{ij}$ has been certified to be less than 0.493. With $n = 31$ and $\kappa_\infty(A) \approx 8 \cdot 10^{13}$, we have certified an absolute error $3.2 \cdot 10^{53}$ for $\text{diff}_{\min} \approx 1.7 \cdot 10^{67}$. Thanks to a maximum relative error $|\widetilde{R} - R|$ certified to be less than 0.2 (only $6 \cdot 10^{-15}$ on the diagonal) we have also checked that $\max \mu_{ij} \leq 0.4991$.

The first main source of failure of the certificate is the failure of the error bounding algorithm when the precision is too small compared to the numerical quality of the tested basis. We have run the certificate on a third class of reduced bases. These bases are obtained by the reduction of "random" (knapsack type) lattice bases in the sense of [22, §3.4]. Here the non reduced bases have random integers of $10^3$ bits in the knapsack weight row. The reduced bases in input of the certificate are dense with integers as large as $10^{45}$ for $n = 75$, and $10^{20}$ for $n = 300$. We use the parameters $(\delta, \eta) = (3/4, 1/2)$ and $(\delta, \eta) = (0.99, 0.5001)$. The choice $(\delta, \eta) = (0.99, 0.5001)$ produces better reduced bases as shown by $\kappa_\infty$ in Figure 6.3 (for a same non reduced basis). Until dimension 175 the certificate is very likely to succeed since the maximum certified relative error is small. On several tenths of trials, the certificate never failed, with a certified $\max|\mu_{ij}|$ as close to $1/2$ (with $\eta = 1/2$) as 0.4999916.

| $n$ | 75 | 125 | 150 | 175 |
|---|---|---|---|---|
| $(\delta_1, \eta_1)$, $\kappa_\infty(A)$ | $6 \cdot 10^5$ | $2.3 \cdot 10^8$ | $1.3 \cdot 10^{10}$ | $2 \cdot 10^{11}$ |
| $\text{diff}_{\min}$ | $1.3 \cdot 10^{37}$ | $4.2 \cdot 10^{20}$ | $3 \cdot 10^{15}$ | $1.2 \cdot 10^{12}$ |
| Rel. err. on $\|r_{ij}\|$ | $1.3 \cdot 10^{-9}$ | $2.2 \cdot 10^{-6}$ | $2.1 \cdot 10^{-5}$ | $6.3 \cdot 10^{-3}$ |
| $(\delta_2, \eta_2)$, $\kappa_\infty(A)$ | $2.4 \cdot 10^4$ | $4 \cdot 10^5$ | $4 \cdot 10^7$ | $9 \cdot 10^8$ |
| Rel. err. on $\|r_{ij}\|$ | $5.1 \cdot 10^{-10}$ | $3.9 \cdot 10^{-8}$ | $6 \cdot 10^{-7}$ | $9.5 \cdot 10^{-6}$ |

Figure 6.3: On "random" reduced bases, $\max|a_{ij}|$ goes from $10^{45}$ ($n = 50$) down to $10^{25}$ ($n = 175$), $(\delta_1, \eta_1) = (3/4, 1/2)$ and $(\delta_2, \eta_2) = (0.99, 0.5001)$.

Beyond dimension 175 with this type of reduced basis, the certificate starts to fail. On dimension 200 with a conditioning about $10^{12}$ with $(3/4, 1/2)$, the error bound on the relative error approaches 1. The properness with $\eta = 1/2$ may become impossible to check, and ask for a certificate with $\eta = 1/2 + \epsilon$, say $\eta = 0.5001$. Note that the Lovász test (18) seems to fail later thanks to much better error bounds on the diagonal. On dimension 300 for $(3/4, 1/2)$ the quality of the reduced bases is too deteriorated ($\kappa_\infty \approx 10^{19}$), and the error bounding algorithm fails with the impossibility of having a small spectral radius for $G$. Nevertheless, on a typical example for dimension 300 with a $(0.99, 0.5001)$ reduced basis, the error bounding algorithm remains effective ($\kappa_\infty \approx 2.5 \cdot 10^{13}$, $\|H\|_\infty \approx 0.6$). The certificate may not be able to certify the actual reducedness of the basis, for example with $\min_i\{t_i - t\} \approx -4.12 \cdot 10^8$, and a too big absolute error bound $4.42 \cdot 10^8$. By changing the certificate parameters to $(\delta - \epsilon_1, \eta + \epsilon_2) = (0.985, 0.515)$, the certificate succeeds again, and therefore is still able to certify a relevant information on the basis. The limitations of the certificate deserve to be further investigated, especially in relation with those identified in [22] for the reduction itself.

# 7. CONCLUSIONS

Between numerical approximation and computer algebra, we propose a certificate for an (exact) algebraic/geometric property—the LLL reducedness of a lattice basis. This work, based on the fast computation of certified error bounds, inherits from the verification methods approach. Thanks

to the IEEE arithmetic standard the floating point errors do not put a curb on the objective of certification. They may rather be mastered and used for accelerating the programs. The foreground of our study is to understand the compromise between the cost and the quality/effectiveness of bounds and certificates, for instance, may we hope for an $O(n^2)$ effective certificate? Computer arithmetics come in the background, where floating point computation, multi-precision, verification identities, intervals, and exact computation are collaborative tools.

We think that our study raises several directions that deserve further investigations. The error bounding problem and its finite precision implementation should be better understood and improved, diagonal scaling and other approximate $QR$ factorizations should be introduced. The usefulness of taking into account the algorithm used for computing $\widetilde{R}$ should be studied (in a more restrictive verification approach). Could the reducedness be certified without resorting to the $QR$ factorization?

To our knowledge, the minimum precision required for a proven LLL variant is $1.6n+o(n)$ with the $\mathrm{L}^2$ algorithm [21, 22] ($\delta$ close to 1 and $\eta$ close to 1/2). Our experiments show we may certify reducedness for dimensions much higher than this worst-case limit ($n_{\max} \leq 53/1.6 \approx 33$). However, the precision required as a function of the dimension remains to be studied. The certificate is very effective for a use complementary to reduction heuristics, it seems worth studying its extensions to reduction algorithms and reducedness certificates with adaptative precision, and sensitive to the numerical quality of the input basis.

**Acknowledgements.** We thank Damien Stehlé for fruitful discussions around the floating point reduction algorithms and heuristics, and for his help in testing reduced bases.

# 8. REFERENCES


[1] ANSI/IEEE 754-1985. Standard for Binary Floating-Point Arithmetic, 1985.

[2] D. Boneh, and G. Durfee. Cryptanalysis of RSA with private key $d$ less than $N^{0.292}$. *IEEE Inf. Theo.*, 46(4):233–260, 2000.

[3] N. Brisebarre and S. Chevillard. Efficient polynomial $L^\infty$-approximations. In *Proc. 18th Symposium on Computer Arithmetic, Montpellier, France*. IEEE Computer Society Press, 2007.

[4] X.-W. Chang and C.C. Paige. Componentwise perturbation analyses for the QR factorization. *Numer. Math.*, 88:319–345, 2001.

[5] X.-W. Chang, C.C. Paige, and G.W. Stewart. Perturbation analyses for the QR factorization. *SIAM J. Matrix Anal. Appl.*, 18:775–791, 1997.

[6] H. Cohen. *A Course in Computational Number Theory*. Springer-Verlag, 2nd Edition, 1995.

[7] J. Demmel, B. Diament, and G. Malajovich. On the Complexity of Computing Error Bounds. *Found. Comput. Math.*, 1(1):101–125, 2000.

[8] J.J. Dongarra, J. Du Croz, I.S. Duf, and S. Hammarling. A set of Level 3 Basic Linear Algebra Subprograms. *ACM Trans. Math. Software*, 16:1–17, 1990.

[9] R. Freivalds. Fast probabilistic algorithms. In *Proc. 8th Symposium on Mathematical Foundations of Computer Science*, LNCS 74, pages 57–69. Springer Verlag, 1979.

[10] Gnu MP. *The GNU Multiple Precision Arithmetic Library, Edition 4.2.1,* `http://gmplib.org`. 2006.

[11] O. Goldreich, S. Goldwasser, and S. Halevi. Public-key cryptosystems from lattice reduction problems. In *Proc. CRYPTO'97, Santa Barbara, California, USA*, LNCS 1294, pages 112–131. Springer Verlag, 1997.

[12] N.J. Higham. *Accuracy and stability of numerical algorithms*. SIAM, Philadelphia, PA, 2nd Edition, 2002.

[13] J. Hoffstein, J. Pipher, and J.H. Silverman. NTRU: a ring-based public key cryptosystem. In *Proc. Third International Symposium ANTS-III, Portland, Oregon, USA*, LNCS 1423, pages 267–288. Springer Verlag, 1998.

[14] E. Kaltofen and G. Villard. Computing the sign or the value of the determinant of an integer matrix, a complexity survey. *J. Comp. Applied Math*, 162(1):133–146, 2004.

[15] A.K. Lenstra, H.W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:515–534, 1982.

[16] Magma. *Handbook of Magma Functions, Version 2.13*. Computational Algebra Group, U. Sydney, Australia, 2006.

[17] Matlab. *User's Guide, V7.2*. The MathWorks, Inc., 2006.

[18] G. Mayer. Result Verification for Eigenvectors and Eigenvalues. In J. Herzberger, editor, *IMACS-GAMM International Workshop, Oldenburg, Germany, 1993*, Stud. Comput. Math., pages 209–276. Elsevier, 1994.

[19] Michael B. Monagan, Keith O. Geddes, K. Michael Heal, George Labahn, Stefan M. Vorkoetter, James McCarron, and Paul DeMarco. *Maple 10 Programming Guide*. Maplesoft, Waterloo, Ontario, Canada, 2005.

[20] MPFR. *The Multiple Precision Floating-Point Reliable Library, Edition 2.2.1,* `http://www.mpfr.org`. 2006.

[21] P.Q. Nguyen and D. Stehlé. Floating-point LLL revisited. In *Proc. Eurocrypt'05*, LNCS 3494, pages 215–233. Springer Verlag, 2005.

[22] P.Q. Nguyen and D. Stehlé. LLL on the average. In *Proc. ANTS VII*, LNCS 4076, pages 238–256. Springer Verlag, 2006.

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

[24] V.Y. Pan and Y. Yu. Certification of numerical computation of the sign of the determinant of a matrix. *Algorithmica*, 30:708–724, 2001.

[25] S.M. Rump. Verification Methods for Dense and Sparse Systems of Equations. In J. Herzberger, editor, *Topics in Validated Computations – Studies in Computational Mathematics*, pages 63–136. Elsevier, 1994.

[26] S.M. Rump. Computational Error Bounds for Multiple or Nearly Multiple Eigenvalues. *Linear Algebra and its Applications*, 324:209–226, 2001.

[27] S.M. Rump. Algorithms for Computing Validated Results. In J. Grabmeier, E. Kaltofen, and V. Weispfenning, editor, *Computer Algebra Handbook*, pages 110–112. Springer-Verlag, Heidelberg, Germany, 2003.

[28] S.M. Rump. Computer-Assisted Proofs and Self-Validating Methods. In B. Einarsson, editor, *Handbook of Accuracy and Reliability in Scientific Computation*, pages 195–240. SIAM, 2005.

[29] S.M. Rump. Verification of positive definiteness. *BIT Numerical Mathematics*, 46:433–452, 2006.

[30] S.M. Rump and T. Ogita. Super-fast validated solution of linear systems. *J. Comp. App. Math*, 199(2):199–206, 2007.

[31] C.P. Schnorr. A more efficient algorithm for lattice basis reduction. *Journal of Algorithms*, 9(1):47–62, 1988.

[32] C.P. Schnorr. Fast LLL-Type Lattice Reduction. *Information and Computation*, 204:1–25, 2006.

[33] C.P. Schnorr and M. Euchner. Lattice basis reduction: improved practical algorithms and solving subset sum problems. *Mathematics of Programming*, 66:181–199, 1994.

[34] D. Stehlé. *Algorithmique de la réduction de réseaux et application à la recherche de pires cas pour l'arrondi de fonctions mathématiques*. PhD thesis, Université Henri-Poincaré - Nancy 1, Nancy, France, December 2005.

[35] G.W. Stewart. On the perturbation of LU, Cholesky, and QR factorizations. *SIAM J. Math. Anal.*, 14(4):1141–1145, 1993.

[36] A. Storjohann. Faster Algorithms for Integer Lattice Basis Reduction. TR249 ETH-Zurich, Dpt. Comp. Sc., Zurich, Switzerland, 1996.

[37] J.-G. Sun. Perturbation bounds for the Cholesky and $QR$ factorizations. *BIT*, 31:341–352, 1991.

[38] J.-G. Sun. Componentwise perturbation bounds for some matrix decompositions. *BIT*, 32:702–714, 1992.

[39] G. Villard. Certification of the QR Factor R, and of Lattice Basis Reducedness. RR2007-3 LIP, ÉNS Lyon, France. `http://hal.archives-ouvertes.fr/hal-00127059/en`

[40] H. Zha. A componentwise perturbation analysis of the QR decomposition. *SIAM J. Matrix Anal. Appl.*, 14(4):1124–1131, 1993.