Certification of the QR factor R, and of lattice basis reducedness

Gilles Villard

▶ **To cite this version:**

Gilles Villard. Certification of the QR factor R, and of lattice basis reducedness. 2007. hal-00127059

HAL Id: hal-00127059
https://hal.science/hal-00127059v1

Preprint submitted on 29 Jan 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

CERTIFICATION OF THE QR FACTOR R, AND OF LATTICE BASIS REDUCEDNESS

Gilles Villard

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

hal-00127059, version 1 - 29 Jan 2007

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 entries of the R factor of the QR matrix factorization. This algorithm takes into account all possible approximation and rounding errors.

The cost $12n^3 + O(n^2)$ of the certificate is only six times more than the cost of numerical algorithms for computing the QR factorization itself, and 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.

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 [31, Ch. II-3] and [20]. The two main constraints are speed and effectiveness. Indeed, 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 heuristic. Hence our general concern is somehow the compromize between speed and proven accuracy. The certificate will be introduced later below. It relies on error bounds for the R factor of the QR factorization of a matrix that we discuss first.

Bounding errors for the factor R. Let A be an $n \times n$ invertible integer matrix. The QR factorization (see for instance [10, 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).

This material is based on work supported in part by the French National Research Agency, ANR Gecko.

LIP Research Report RR2007-03, École Normale Supérieure de Lyon — January, 2007.

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 Section 6 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 $H \in \mathbb{F}^{n\times n}$ with positive entries such that (see (9) on page 7):

$$|\widetilde{R} - R| \leq H|\widetilde{R}|. \tag{1}$$

Since floating point numbers are rational numbers, when $\widetilde{R}$ and E are known, (1) provides a rigourous mathematical bound for the error with respect to the unkown matrix R.

For understanding the behaviour of the error bounding algorithm better, we recall in Section 3 some existing numerical pertubation analyses for the QR factorization. The necessary background material may be found in Higham's book [10]. Then in Sections 4 and 5, we give the mathematical foundations of our approach. We focus on the componentwise bounds of [34] 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 [25, 26], see also the short discussion in Section 2. As numerical experiments of Section 6.3 will demonstrate, the error bounding algorithm is effective in practice. Its cost 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 themselves calculated using floating point operations, nevertheless, they take into account all possible numerical and rounding errors. The reducedness certificate will 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].

At a given precision, the error bounding algorithm provides relevant bounds for input matrices with appropriate numerical properties. In particular, the dimension and related condition numbers should be considered in relation with the precision (see Section 6.3). However, the power of the verification approach [25, 26] is to be effective on many inputs for which the numerical approach itself is effective—here the numerical QR factorization. For example, we report experiments using 64 bits floating point numbers, and $\widetilde{R}$ computed by the modified Gram-Schmidt orthogonalization (see [10, Alg. 19.12]). On integer matrices of dimension $n = 1500$ with condition number around 10^5, we certify that the relative error on the entries of $\widetilde{R}$ has order as small as 10^{-6} or 10^{-5}, with only 10^{-10} or 10^{-9} on the diagonal. We refer here to the diagonal entries since they play a key role for instance in the LLL Lovász test (see (3)). For large condition numbers (with respect to double precision), say 10^{12}, and $n = 200$, the algorithm may typically certify relative errors in 10^{-1}, and 10^{-4} on the diagonal.

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 [12], where the reduction is defined via the Gram-Schmidt orthogonalization of $a_1, a_2, \ldots, a_n \in \mathbb{Z}^n$. The Gram-Schmidt orthogonalization determines the associated orthogonal basis $a_1^*, a_2^*, \ldots, a_n^* \in \mathbb{Q}^n$ by induction, together with factors μ_{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 Gram-Schmidt orthogonalization satisfies

$$|\mu_{ij}| \leq \eta, \ 1 \leq j < i \leq n. \tag{2}$$

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 δ and η if the vectors are proper, and if they satisfy the Lovász conditions:

$$(\delta - \mu_{i+1,i}^2)\|a_i^*\|_2^2 \leq \|a_{i+1}^*\|_2^2, \ 1 \leq i \leq n-1, \tag{3}$$

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

$$\begin{cases} \|a_i^*\|_2 = r_{ii}, \ 1 \leq i \leq n, \\ \mu_{ij} = r_{ji}/r_{jj}, \ 1 \leq j < i \leq n. \end{cases} \tag{4}$$

We see from (4) that if an approximation $\widetilde{R}$ of R with error bounds on its entries are known, then (depending on the quality of the bounds) it may be possible to check whether (2) and (3) are satisfied. All the above draws the reducedness certificate that we propose in Section 7. We also 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 $\widetilde{R}$ such that $A \approx \widetilde{Q}\widetilde{R}$;

II: Certified computation of $F \in \mathbb{F}^{n \times n}$ such that $|\widetilde{R} - R| \leq F$ (see (1));

III: Certified check of properness (2) and Lovász conditions (3).

Following the principles of verification algorithms [26], Step I is purely approximation, and we propose an implementation of Steps II and III that is independent of the factorization 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 Section 6.1). We rely on the fact that the arithmetic operations $+, -, \times, \div, \sqrt{\ }$ in $\mathbb{F}$ are according to the IEEE 754 standard. We especially use explicit changes of rounding mode for certified bounds.

Verification algorithms are a powerful alternative between numerical and computer algebra algorithms, they somehow illustrate the boundary between the two fields. The reducedness certificate we propose illustrates a cooperation of purely numerical computation with a certified approach based on the IEEE 754 standard, in order to provide a computer algebra answer. Our progress in linear algebra is in the line of previous works on error bounds for linear systems [23, 21, 28], on certifying the sign of the determinant [22, 11], on verifying positive definiteness [27], or on eigenvalues [15, 24]. Our contribution is to establish the effectiveness of componentwise bounds for a whole matrix, propose a corresponding certified algorithm using fast verification techniques, and derive and test with experiments a certificate for the LLL reducedness application.

Absolute value and matrix norms. We already considered above the absolute value of a matrix $A = (a_{ij})$ defined by $(|a_{ij}|)$. We write $|A| \leq |B|$ if $|a_{ij}| \leq |b_{ij}|$. It is possible to check that if $A = BC$ then $|A| \leq |B||C|$. We will use several matrix norms (see [10, Ch. 6]) such as the Frobenius norm $\|\cdot\|_F$ or the 2-norm $\|\cdot\|_2$. We will also especially use 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 $h_{ij} = h$.

Condition numbers. 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 ∞ [10, Th. 6.4]. With the infinity norm we will also use the Bauer-Skeel condition number $\mathrm{cond}(A) = \||A^{-1}||A|\|_\infty \leq \kappa_\infty(A)$ [10, § 7.2].

2 Error bounds computation and verification algorithms

In linear algebra, few things are known about the complexity of computing certified and effective error bounds. The problem is somewhere between the one of computing approximate solutions, and the one of computing multi-precision or exact solutions. A main result in [7] shows that the problem of computing a certified estimation of $\|A^{-1}\|$ (for a consistent matrix norm) is as difficult as testing whether the product of two matrices is zero. Hence if we consider $O(n^3)$ operations for multiplying two matrices of dimension n, a deterministic error bound—based on a condition number bound—would cost $O(n^3)$. The use of randomization may lead to error estimations in $O(n^2)$ operations, we refer to [10, Chap. 15] and references therein, and to the fact that the matrix product could be verified in $O(n^2)$ operations [9]. We did not investigate the randomization possibilities yet.

Verification methods have been developped in [23, 21] for computing certified error bounds for linear system solution. In [21] the error bound (normwise) is computed in twice the time of numerical Gaussian elimination. In the same spirit, a verification approach using $O(n^3)$ floating point operations is proposed in [22] for the sign of the determinant (see [11] for a survey on this topic). Note that computing the sign of the determinant corresponds to knowing the determinant with a relative error less than 1. Our error bounding algorithm for R will also use $O(n^3)$ floating point operations. The verification approach [25, 26] gives an effective alternative to interval arithmetic whose exponential overestimation of the error would not be appropriate for our problem [26, §10.7]. The general strategy for calculating an error bound is first to establish a result whose assertion is a mathematical expression for the bound (see Theorem 4.2), then design an algorithm that verifies the assumptions for the latter assertion, and computes a certified evaluation of the bound (see Section 5).

3 Perturbation analyses and bounds for the QR factorization

A finite precision computation of the QR factorization of A leads to an approximate factor $\widetilde{R}$. The errors in $\widetilde{R}$ with respect to R are called the *forward errors* (absolute or relative). The matrix $\widetilde{R}$ is not the factor of the QR factorization of A, however, it is seen as the QR factor of a perturbed matrix $\widetilde{A} = A + E$, where E is called the *backward error.* The choice of

$\widetilde{A}$ is non unique, and one refers for instance for the smallest error norm. The link between backward and forward error is made using the condition number of the problem, hence for us the condition number for the problem of computing R. The (relative) *condition number* of the problem—under some class of perturbations—measures the relative change in the output for a relative change in the input. In this context, a useful tool for estimating the accuracy of the solution to a problem, is the rule of thumb [10, p. 9]:

$$\text{forward error} \lesssim \text{condition number} \times \text{backward error}. \tag{5}$$

We survey below some more precise instantiations of (5) for the QR factorization. Known results are, in general, approximate inequalities (first order results), but could be extended for giving strict bounds on the forward error. The rule of thumb therefore gives a first possible direction for deriving an error bounding algorithm for $|\widetilde{R} - R|$ (the forward absolute error). However, most of corresponding bounds rely on matrix norms, and may thus overestimate the actual componentwise error in most cases.

We will investigate an alternative direction in Section 4. Rather than on the rule of thumb, our error bounding algorithm will be based on the componentwise bounds of Sun [34]. This will lead to an algorithm that seems to be naturally more effective than a matrix norm approach for our problem. Another advantage of using Sun's results is to remain in the spirit of the verification methods. In particular, we will see that the error bounding algorithm is oblivious of the algorithm that is used for computing the approximate factor $\widetilde{R}$. 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 for using (5). An approximate $\widetilde{Q}$ in not orthogonal in general, the backward error problem is to know for which matrix $\widetilde{A}$ close to A, there exists an orthogonal $\widehat{Q}$ such that $\widetilde{A} = \widehat{Q}\widetilde{R}$? Backward error bounds are known for specific QR algorithms such as Householder or Gram-Schmidt ones (see Theorems 19.4 and 19.13 in [10]), but may not be available in the general case. We will circumvent the need of the backward error in Section 4 using the correspondence between the QR factorization of A, and the Cholesky factorization R^TR of A^TA.

Sensitivity of the QR factorization. The condition number of the problem of computing R (the "rate of change" of R) in the QR factorization may be defined theoretically for given classes of perturbations, but it is non trivial to derive expressions of the condition number 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 $A \approx \widetilde{A} + E = \widehat{Q}\widetilde{R}$ be QR factorizations. As already noticed, for a floating point factorization $A \approx \widetilde{Q}\widetilde{R}$, in general we have $\widehat{Q} \neq \widetilde{Q}$ since $\widetilde{Q}$ is not orthogonal. Let $\widetilde{R} = R + F$. For a sufficiently small backward error E, consider the normwise relative error $\epsilon = \|E\|_F/\|A\|_2 = \|\widetilde{A} - A\|_F/\|A\|_2$. Then Sun's [33, Rem. 3.5] perturbation bounds (see also [32]) give

$$\|\widetilde{R} - R\|_F/\|R\|_2 \leq \sqrt{2}\kappa_2(A)\epsilon + O(\epsilon^2). \tag{6}$$

An improved bound is given by Zha [Theorem 2.1][35] (see also [4, § 5] and [10, §19.9]) under a componentwise model of perturbation that we simplify here. Let $|\widetilde{A} - A| = |E| = \epsilon|A|$, then for sufficiently small ϵ we have:

$$\|\widetilde{R} - R\|_\infty / \|R\|_\infty \leq c_n \text{cond}(R^{-1})\epsilon + O(\epsilon^2) \tag{7}$$

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 $\log \text{cond}(R^{-1})$. This typical behaviour is illustrated by Figure 3.1 where we have computed QR factorizations of random matrices (of `randsvd` type [10, Ch. 28]). The algorithm used is the Modified Gram-Schmidt algorithm [10, Algo. 19.12].

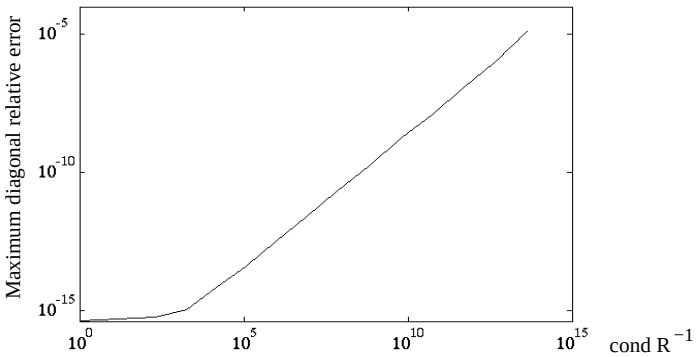

Figure 3.1: Maximum relative diagonal error in R (Modified Gram-Schmidt algorithm) with respect to $\text{cond}(R^{-1})$ for random matrices A ($n = 200$).

Identities (6) and (7) provide first order estimations of the errors. They are essential for an idea of the normwise loss of accuracy. Nevertheless, the loss of accuracy on individual entries (needed for the reducedness certificate) may not be deduced from these identities. Consider for instance the case of Figure 3.1 where the ratios of the r_{ij} may be as large as 10^{11}. The normwise bound of (7), that involves the max row sum $\|R\|_\infty$, cannot provide relevant informations for every $|\tilde{r}_{ij} - r_{ij}|$. Note also that the loss of accuracy would certainly be amplified by the implementation of the error estimation itself in finite precision (Figure 3.1 is a mathematical representation of the error). Normwise bounds much sharper than (6) and (7) may be found, especially in [5, 4], it remains to know how well the corresponding proposed estimations approximate the true condition number [4, §10]. It would also be interesting to investigate how the new techniques of [5, 4] could lead to practical componentwise bounds.

4 Strict componentwise bounds for the R factor

We now present the mathematical view and justification of the error bounding algorithm of Section 6. Given $A \in \mathbb{R}^{n\times n}$ invertible, and an upper triangular matrix $\widetilde{R} \in \mathbb{R}^{n\times n}$, the problem is to bound $|\widetilde{R} - R|$ where R is the unknown QR factor of A. In practice we will have $A, \widetilde{R} \in \mathbb{F}^{n\times n}$.

4.1 QR and Cholesky factorization

The strict componentwise analysis of Sun [34, § 4] for QR uses the matrix $\widetilde{A}$ such that $\widetilde{A} = \widehat{Q}\widetilde{R}$ is a QR factorization. Note that, because of the loss of orthogonality, if $\widetilde{Q}$ is a numerical approximation of Q then $\widetilde{A}$ is not in general the matrix $\widetilde{Q}\widetilde{R}$. Informations on $\widetilde{A}$ may be available by taking into account the algorithm that has produced $\widetilde{R}$. We refer for instance to [4, Eq. (5.8)] and [10, §19.9] where properties of Householder transformations are used for bounding the backward error. This is not sufficient for our problem since we are given only A and $\widetilde{R}$, and since one of our goal is to be oblivious of the method used for $\widetilde{R}$.

For not relying on $\widetilde{A}$, we propose to rather resort to Sun's study of the Cholesky factorization [34, § 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 [10, 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. It may not be a good idea to use the Cholesky factorization for computing R numerically. The condition number of the problem may indeed increase too much, especially $\kappa_2(A^T A) = (\kappa_2(A))^2$. For avoiding this drawback, our point is to implement the reduceness certificate of Section 7 using QR for computing $\widetilde{R}$, and to use the Cholesky point of view only for computing the error bound.

4.2 The bound on $|\widetilde{R} - R|$

For a matrix $A \in \mathbb{R}^{n\times n}$, the spectral radius $\rho(A)$ is the maximum of the eigenvalue modules. 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 $i \leq j$, and $t_{ij} = 0$ otherwise. The following Theorem is [34, Th. 2.1].

Theorem 4.1. *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*

$$G = |\widetilde{R}^{-T} E \widetilde{R}^{-1}|. \tag{8}$$

Then if $\rho(G) < 1$ we have

$$|\widetilde{R} - R| \leq \mathrm{triu}(G(I - G)^{-1})|\widetilde{R}|. \tag{9}$$

Inequality (9) is what we announced with (1). Let us apply Theorem 4.1 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 from (8):

$$\begin{aligned} G &= |\widetilde{R}^{-T} E \widetilde{R}^{-1}| = |\widetilde{R}^{-T}(\widetilde{B} - B)\widetilde{R}^{-1}| = |\widetilde{R}^{-T}(\widetilde{A}^T\widetilde{A} - A^T A)\widetilde{R}^{-1}| \\ &= |\widetilde{R}^{-T}\widetilde{A}^T\widetilde{A}\widetilde{R}^{-1} - \widetilde{R}^{-T}A^T A\widetilde{R}^{-1}| = |\widehat{Q}^T\widehat{Q} - \widetilde{R}^{-T}A^T A\widetilde{R}^{-1}| = |\widetilde{R}^{-T}A^T A\widetilde{R}^{-1} - I|. \end{aligned}$$

Going back to the R factor of the QR factorization we then have the following corollary to Theorem 4.1

Theorem 4.2. *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*

$$G = |\widetilde{R}^{-T} A^T A \widetilde{R}^{-1} - I|. \tag{10}$$

Then if

$$\rho(G) < 1, \tag{11}$$

we have

$$|\widetilde{R} - R| \leq \mathrm{triu}(G(I-G)^{-1})|\widetilde{R}|. \tag{12}$$

Proof. Since $\widetilde{R}$ is invertible, $\widetilde{B} = \widetilde{R}^T\widetilde{R}$ is positive definite, the same holds for $B = A^TA$. By construction R and $\widetilde{R}$ are the Cholesky factors of B and $\widetilde{B}$. It suffices to apply Theorem 4.1 for concluding. □

Few things are known about the (mathematical) quality of Bound (12) over $\mathbb{R}$. Furthermore, both additional method and arithmetic errors will be introduced for the finite precision evaluation of the bound. Additional method errors will be introduced especially for calculating certified bounds for $\widetilde{R}^{-1}$ and $\mathrm{triu}(G(I-G)^{-1})$ (see Section 5). Additional arithmetic errors will be introduced by the finite precision itself. All together we produce an error bounding algorithm that is not fully analyzed, the experiments of Section 6.3 will however give a precise idea of its practical behaviour and effectiveness. For illustrating Bound (12) over $\mathbb{R}$, let us consider some examples that show that Theorem 4.2 leads to accurate bounds. The calculations have been done in Maple [16], either exactly or with high precision, then rounded for the presentation. Let $H = \mathrm{triu}(G(I-G)^{-1})$ such that (12) is $|\widetilde{R} - R| \leq H|\widetilde{R}|$.

On the matrices used for Figure 3.1 (`randsvd`, $n = 200$), with $\widetilde{R}$ computed using 64 bits floating point numbers via the Modified Gram-Schmidt algorithm, we typically get the following. For A with $\mathrm{cond}(R^{-1}) \approx 10^5$, the infinity norm of the error matrix is $\|H\|_\infty \approx 2 \times 10^{-9}$. This leads to the knowledge that $\widetilde{R}$ approximates R with (relative) accuracy $\approx 10^{-10}$. The accuracy of $\widetilde{R}$ is about 10^{-13} for the diagonal entries, and the diagonal error estimation is only in a factor of 2 from the true diagonal error. If $\mathrm{cond}(R^{-1}) \approx 4 \times 10^{13}$ then $\|H\|_\infty \approx 3 \times 10^{-3}$, and R is known with accuracy about 10^{-2} (2×10^{-5} on the diagonal). The ratio between the estimation and the true error is less than 4 on the diagonal. Again, we will certainly loose accuracy with our finite precision implementation, but keep a very satisfying overall behaviour. 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 $\mathrm{cond}(R^{-1}) \approx 2 \times 10^{10}$. We compute the matrix $\widetilde{R}$ in Matlab [14], and obtain over $\mathbb{R}$ the error bound:

$$|\widetilde{R} - R| \approx \begin{bmatrix} 9.7 \times 10^{-17} & -1.3 \times 10^{-16} \\ 0 & 3.7 \times 10^{-17} \end{bmatrix} \leq \begin{bmatrix} 3.5 \times 10^{-12} & 3.5 \times 10^{-12} \\ 0 & 7.4 \times 10^{-17} \end{bmatrix}. \tag{13}$$

The matrix R is known with (relative) accuracy about 2.5×10^{-12} on the first row, and 5.25×10^{-7} for r_{22}. On the first row the error is overestimated by a factor about 3.6×10^4. Notwithstanding the fact that the accuracy of the bound produced by Theorem 4.1 is

penalized by the particular form of the matrix, the estimation of the accuracy of $\widetilde{R}$ remains very good. Now let A be the random 3×3 integer matrix

$$A_2 = \begin{bmatrix} -60 & 28 & 51 \\ -24 & -35 & -89 \\ 37 & 51 & -23 \end{bmatrix}.$$

We look at Bound (12) when perturbing only the second row of the exact R and get:

$$|\widetilde{R} - R| = \left| \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.0071 & -0.0052 \\ 0 & 0 & 0 \end{bmatrix} \right| \leq \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.014204 & 0.023087 \\ 0 & 0 & 2.9 \times 10^{-6} \end{bmatrix}. \tag{14}$$

The estimator computes the errors very well on the first and the second row. We think that the dummy error estimated for r_{33} is a repercussion of the perturbation of row two. In next section we review the different quantities that are involved in Theorem 4.2 with the aim of looking at first implementation aspects.

5 Toward an implementation

Theorem 4.2 is the foundation of our error bounding algorithm. It involves several quantities that need further study before deriving an implementation in Section 6. We decompose the computation of the bound on $|\widetilde{R} - R|$ into four principal tasks. We need to: 1) check that $\widetilde{R}$ is invertible; 2) compute a bound on G; 3) check that $\rho(G) < 1$; and 4) bound $H = \text{triu}(G(I - G)^{-1})$. We recall that at this point, only A and $\widetilde{R}$ are known.

5.1 Invertibility check of $\widetilde{R}$

For dealing with $\widetilde{R}^{-1}$ in a certified way, which is clearly a non trivial question in finite precision, we use the verification solution of Oishi and Rump [21]. We compute a purely numerical approximate inverse $V \approx \widetilde{R}^{-1}$ (by numerical triangular inversion). Then we know from [21] that, if

$$\|\widetilde{R}V - I\|_\infty < 1, \tag{15}$$

then $\widetilde{R}$ is invertible.

5.2 Bounding G

For bounding G, and dealing with the unknown inverse of $\widetilde{R}$, we are also inspired by [21], and introduce $W = \widetilde{R}V$ ($\approx I$). We have

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

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

$$\begin{aligned} G &\leq |W^{-T}| \cdot |V^T A^T A V - I + I - W^T W| \cdot |W^{-1}| \\ &\leq |W^{-T}| \cdot |(V^T A^T A V - I) - (W^T W - I)| \cdot |W^{-1}| \end{aligned}$$

which gives

$$G \leq |W^{-T}| \cdot (|(V^T A^T A V - I)| + |(W^T W - I)|) \cdot |W^{-1}|. \tag{16}$$

We will use (16) for computing a certified bound for G. The products involving A, $\widetilde{R}$, 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 [21, Intro.]). Then, when $\widetilde{R}$ is invertible,

$$\begin{aligned} |W^{-1}| &= |I + (I - W) + (I - W)^2 + \dots| \\ &= |2I - W + (I - W)^2(I + (I - W) + (I - W)^2 + \dots)| \\ &\leq |2I - W| + |(I - W)^2| \cdot |I + (I - W) + (I - W)^2 + \dots| \\ &\leq |2I - W| + \mathcal{M}(\|I - W\|_\infty^2 / (1 - \|I - W\|_\infty)) \end{aligned}$$

where $\mathcal{M}(x)$ for $x \in \mathbb{R}$ denotes the matrix whose all entries are equal to x. Here we have used the fact that the entries of $|I - W|^2 \cdot |I + (I - W) + (I - W)^2 + \dots|$ are bounded by the infinity norm. Since W is triangular, it follows that

$$|W^{-1}| \leq |2I - W| + \frac{\|I - W\|^2}{1 - \|I - W\|_\infty} \cdot \mathrm{triu}(1_n \cdot 1_n^T) \tag{17}$$

where 1_n is the column vector with all entries equal to 1. Note that the invertibility check (15) 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 (16) and (17) follow some keys to the design of verification methods. We especially refer to [26, p. 211] where the introduction of small factors is recommended. We have introduced the matrices $V^T A^T A V - I$ and $W^T W - I$ whose absolute bounds are expected to be small when $\widetilde{R} \approx R$ and $W \approx I$. On the other hand, in (17), $|2I - W|$ is expected to be close to I, and remaining terms are second order terms (see also the analysis for α in [21, §5]).

5.3 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

$$\|G\|_\infty < 1 \tag{18}$$

for asserting that $\rho(G) < 1$ in Theorem 4.2. This test corresponds to the Gershgörin disks. It could certainly be sharpened in future versions of the certificate, see for instance the Cassini ovals in [3], or the iterative estimation in [27].

5.4 Bounding $|\widetilde{R} - R|$

Once a bound on G is known it remains to bound $H = \mathrm{triu}(G(I-G)^{-1})$. We have

$$G(I-G)^{-1} = G + G^2 + G^3 + \ldots = G + G^2(I + G + G^2 + \ldots)$$

and

$$\mathrm{triu}(G(I-G)^{-1}) \leq \mathrm{triu}(G) + \mathrm{triu}\left(\frac{\|G\|_\infty^2}{1-\|G\|_\infty} \cdot 1_n \cdot 1_n^T\right). \tag{19}$$

Since G is expected to be small, $H = \mathrm{triu}(G(I-G)^{-1})$ is expected to be close to $\mathrm{triu}(G)$. Note that using the spectral radius check (18) ensures that $1 - \|G\|_\infty > 0$.

6 Error bounding algorithm for the QR factor R

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 $\mathbb{F}^{n\times n}$. Since (finite) floating point numbers are rational numbers, A and $\widetilde{R}$ can be seen as rational matrices. Let $R \in \mathbb{R}^{n\times n}$ be the unknown QR factor of A (in general, the entries of R are not in $\mathbb{F}$). We carry the approach of Section 5 over to the floating point case for computing a floating point matrix H such that $|\widetilde{R} - R| \leq H|\widetilde{R}|$. The error matrix H provided by Theorem 4.2 can be computed modulo the two checks (15) and (18), and using the inequalities (16), (17), and (19). These checks and inequalities only involve matrix multiplications, additions, subtractions, and divisions by a scalar. After explaining the basic techniques we use for computing certified bounds in floating point arithmetic, we present the error bounding algorithm and demonstrate its effectiveness on various examples.

6.1 Certified bounds for floating point matrix expressions

We denote by $\mathrm{fl}(x)$ the value of an arithmetic expression x computed by floating point arithmetic in $\mathbb{F}$. For instance, for $a, b \in \mathbb{F}$, $\mathrm{fl}(a + b \times c)$ denotes the result in $\mathbb{F}$ with the addition and the mutiplication performed in floating point arithmetic. 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 $a, b \in \mathbb{F}$, $\mathrm{fl}(|a|) = |a|$, $\mathrm{fl}(\max\{a, b\}) = \max\{a, b\}$, $\mathrm{fl}(-a) = -a$.

Thanks to the IEEE 754 standard, we can use the possibility of changing the rounding mode for computing certified bounds. We essentially follow Rump's approach for implementing verified matrix operations [26], and Oishi and Rump [21]. We use the statements "setround(down)" and "setround(up)"*. All operations after a statement "setround(down)" or "setround(up)" are rounded downwards or upwards, respectively, until the next call to setround. For two floating point numbers a and b, a bound r on $|a \; op \; b|$ for $op \in \{+, -, \times, \div\}$ may be computed as follows. The program

$$\begin{array}{ll} \text{setround(down);} & \underline{r} = \mathrm{fl}(a \; op \; b) \\ \text{setround(up);} & \overline{r} = \mathrm{fl}(a \; op \; b); \quad r = \max\{|\underline{r}|, |\overline{r}|\} \end{array} \tag{20}$$

*`fesetround(FE_DOWNWARD)` and `fesetround(FE_UPWARD)` in C language.

leads to $\underline{r}$ and $\overline{r}$ such that $\underline{r} \le a \; op \; b \le \overline{r}$, and to $r \in \mathbb{F}$ such that $|a \; op \; b| \le r$, for any a and b, and any op. 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 $A \times B - C$ with $A, B, C \in \mathbb{F}^{n \times n}$. If $A \times B$ is implemented using only additions and multiplications, then the program

$$\begin{array}{ll} \text{setround(down);} & \underline{R} = \text{fl}(A \times B - C) \\ \text{setround(up);} & \overline{R} = \text{fl}(A \times B - C); \quad R = \max\{|\underline{R}|, |\overline{R}|\} \end{array} \tag{21}$$

where the maximum is taken componentwise, provides $\underline{R} \le A \times B - C \le \overline{R}$, and $R \in \mathbb{F}^{n \times n}$ such that $|A \times B - C| \le R$. For bounding more general matrix expressions we will use a midpoint-radius matrix representation (we refer to [26, §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. The intervals are for instance obtained by a computation of type (21). Then the program [26, Fig. 10.22]:

$$\begin{array}{ll} \text{setround(up);} & \text{m}_M = \text{fl}((\overline{M} - \underline{M})/2); \quad \text{r}_M = \text{fl}(\text{m}_M - \underline{M}) \\ & \text{m}_N = \text{fl}((\overline{N} - \underline{N})/2); \quad \text{r}_N = \text{fl}(\text{m}_N - \underline{N}) \\ \text{setround(down);} & \underline{R} = \text{fl}(\text{m}_M \times \text{m}_N - I) \\ \text{setround(up);} & \overline{R} = \text{fl}(\text{m}_M \times \text{m}_N - I) \\ & R = \text{fl}\left(\max\{|\underline{R}|, |\overline{R}|\} + |\text{m}_M| \times \text{r}_N + \text{r}_M \times (|\text{m}_N| + \text{r}_N)\right) \end{array} \tag{22}$$

computes R such that $|M \times N - I| \le R$. Both (21) and (22) allow to use fast matrix routines such as the BLAS ones (see the general discussion in [26, §10.9]) The number of operations in $\mathbb{F}$ needed is 2 and 4 matrix products, respectively.

Other matrix operations that we will perform are additions, products, and divisions by scalars for matrices with positive entries (absolute values essentially). We also compute infinity norms. With no subtraction involved, certified bounds can be computed using directed rounding. From (20), upper bounds for these computations are obtained by evaluating the floating point expressions after a "setround(up)" statement. For upper bounds on divisions by a floating point number $1 - g$, we first compute upper bounds for $-(g-1)$ and $1/(g-1)$.

Other approaches for certified matrix computations could be considered. We refer to Rump [26] for a general discussion on this topic, and for the efficiency of the approach chosen here.

6.2 Computing an error bound

For A and $\widetilde{R}$ in $\mathbb{F}^{n \times n}$, $\widetilde{R}$ upper triangular, we follow Section 5 for computing a floating point matrix H such that $|\widetilde{R} - R| \le H|\widetilde{R}|$. All operations are done in the given floating point number set $\mathbb{F}$. For simplifying the presentation we often forget the costs in $O(n^2)$.

The first step is the computation of $V \approx \widetilde{R}^{-1}$. Such a triangular matrix inversion is done in $n^3/3$ operations [10, Ch. 14]. We then compute $\underline{W}$ and $\overline{W}$ for $W = \widetilde{R}V$ by two triangular matrix products, this is done in $2n^3/3$ operations. This dominates the cost for checking that $\widetilde{R}$ is invertible by bounding $|W - I|$ using (21), and by the infinity norm test (15). Re-using $\underline{W}$ and $\overline{W}$, a bound on $|W^{-1}|$ is then computed using (17) in $O(n^2)$ operations. The latter uses (21) for $|W - 2I|$, and computes a bound with positive matrices using directed upwards

rounding. We now look at bounding G using (16). Since G is symmetric we restrict ourselves to counting the operations for calculating the upper triangular part. With $W \in [\underline{W}, \overline{W}]$ one can bound $|W^T W - I|$ using (22) in four matrix products. Since W is upper triangular, and W^T is lower triangular, the bound is obtained in $4n^3/3$ operations. We then use (21) and (22) for computing an interval for AV in $2n^3$ operations (two dense $\times$ triangular matrix products), and for bounding $|V^T A^T A V^T - I|$ in $4n^3$ operations (four dense products resulting in a symmetric matrix). A bound on G is deduced by operations on matrices with positive entries in $4n^3/3$ operations. The latter is essentially two dense $\times$ triangular matrix products with a symmetric result. Once a bound on G is known, testing its spectral radius by (18) costs $O(n^2)$ operations. G has positive entries, a bound on the error matrix H can then be computed by directed towards rounding using (19) also in $O(n^2)$ operations.

We summarize this analysis, and take into account the final matrix product $H|\widetilde{R}|$ in the following result.

Theorem 6.1. *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.*

A QR factorization typically costs $2n^3 + O(n^2)$ (Gram-Schmidt or Householder approaches) or $3n^3 + O(n^2)$ (using Givens rotations). 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 language. The error bounding program takes in input two floating point matrices A and $\widetilde{R}$ and always returns a matrix F. The entries of F are finite (positive) 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.

6.3 Computational results

The results we present here correspond to the application of Theorem6.1 with 64 bits floating point numbers. In this section and in Section 7 the condition numbers and the "true errors" have been computed with high precision using Mpfr [18]. For several types of matrices, we study the behaviour of the certified error bound by looking at its value and its 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 the entries of $\widetilde{R}$. Let us first come back on the examples of Section 4. On the matrix A_1, and $\widetilde{R}$ from Matlab, we compute the bound

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

Comparing to (13), we see that the finite precision estimator we propose is only slightly overestimating the best bound that could be obtained by the method. On the matrix A_2,

and the corresponding perturbation of the exact R we get:

$$|\widetilde{R} - R| \leq \begin{bmatrix} 8.8 \times 10^{-6} & 9.52 \times 10^{-6} & 1.96 \times 10^{-6} \\ 0 & 0.014207 & 0.023098 \\ 0 & 0 & 1.16 \times 10^{-5} \end{bmatrix}.$$

The "large" perturbation of the second row is detected very accurately. For next results, $\widetilde{R}$ is computed with the Modified Gram-Schmidt algorithm using 64 bits numbers as for the estimator. Our tests use ten matrix samples.

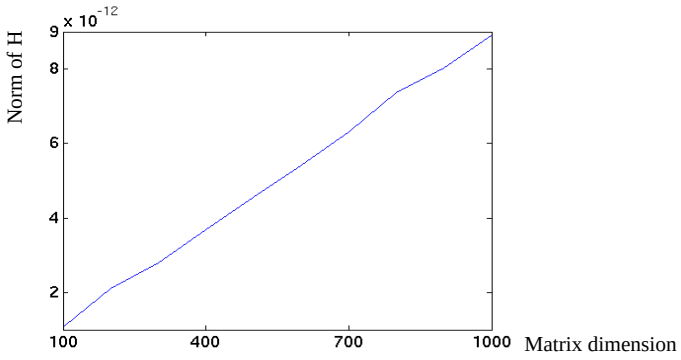

Figure 6.1: Certified $\|H\|_\infty$ for random matrices A with $\kappa_2(A) \approx 10^3$.

We first illustrate the *value of the certified bound with respect to the dimension*. Figures 6.1 and 6.2 are for random input matrices A (of `randsvd` type [10, Ch. 28]).

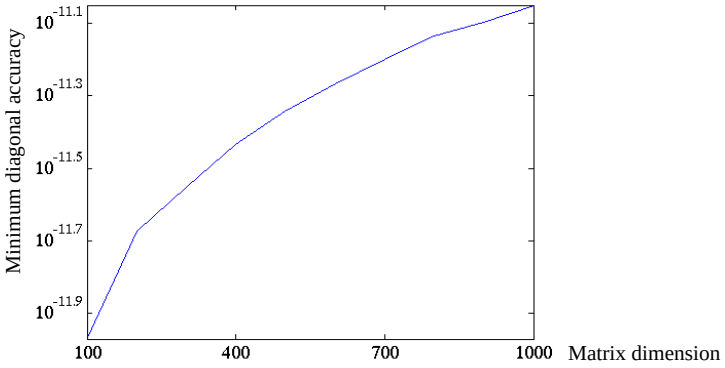

Figure 6.2: Certified maximum relative error on R for random matrices A such that $\kappa_2(A) \approx 10^3$ (y axe with logarithmic scale).

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 of $\widetilde{R}$, 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 axe on Figure 6.2). Such small increase rates—that are typical of 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 for a better understanding of the latter behaviour, especially of the influence of the product $H|\widetilde{R}|$, and of the magnitudes in R. Note also that for the two latter figures, $\mathrm{cond}(R^{-1})$ is sligthly growing, and the growth of the estimation depends on the true error itself.

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 above `randsvd` matrices we also consider random integer matrices with entries of absolute values less than 1000. On these two types of matrices we obtain similar results. The condition numbers $\kappa_\infty(A)$ are varying from about 10^4 to 10^6. On random integer matrices of dimension 1500, the maximum exact relative error on R has order 10^{-10} to 10^{-9}. We are able to 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 about 10^3 for $n = 200$ to about 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 (where the error itself is much smaller in general). On matrices with small condition number (generated using Matlab `gallery('orthog')` [10, Chapter 28]) the quality of the certified bound may be remarkably small and stable with respect to the dimension. 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*. We first report that the quality maybe be very good even for matrices with high condition number. For Figure 6.4 we use $A = QA_K \in \mathbb{F}^{n\times n}$. The matrices Q are random orthogonal from the Matlab `gallery` function [10, 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 $\theta = 1.2$.

Dimension	10	20	30	40	50	60	70
$\kappa_\infty(A)$	10^2	1.3×10^4	1.1×10^6	7.8×10^7	4.8×10^9	2.8×10^{11}	1.5×10^{13}
Bound/error	45	106	281	161	103	140	152
Certified digits in $\widetilde{R}$	14	12	10	9	7	5	4

Figure 6.4: Ratio of the certified relative error bound and the true error (max) on Kahan matrices, and number of significant decimal digits certified in $\widetilde{R}$.

However, in general, the quality of the bound may depend on the condition number. Consider for instance the ratio of the certified relative error bound and the true error (max) for small matrices ($n = 10$). For a Chebyshev Vandermonde-like (nearly orthogonal, $\kappa_\infty \approx 13$), the ratio is about 11. We have a ratio about 14 for Toeplitz and symmetric positive definite matrices ($\kappa_\infty \approx 700$). On the Pascal matrix ($\kappa_\infty \approx 8\times10^9$) we get a ratio about 25, and

about 1600 for the Hilbert matrix ($\kappa_\infty \approx 3.5 \times 10^{13}$). Figure 6.5 is more general. The overestimation of certified error bound seem to increase quite slowly with the condition number.

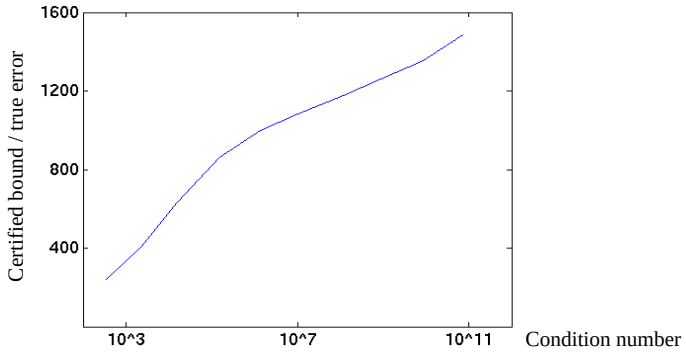

Figure 6.5: Ratio of the certified relative error bound and the true error (max) with respect to $\kappa_\infty(A)$ on `randsvd` matrices of dimension $n = 200$.

We see that the limits of our algorithm, we mean the conditions in which it is returning 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 (18). We return finite bounds for the error on every entries of $\widetilde{R}$ for the Pascal matrix of dimension 14 ($\kappa_\infty \approx 3.8 \times 10^{14}$, $\|G\|_\infty \approx 0.06$). For $n = 15$ the algorithm produces infinity bounds. On random `randsvd` matrices of dimension 40, the algorithm is effective until $\kappa_\infty \approx 3 \times 10^{14}$ 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 (5) 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.

7 A certificate for LLL reducedness

To an $n \times n$ integer matrix A we associate the Euclidean lattice $\mathcal{L}$ generated by the columns (a_j) of A. About lattices the reader may refer for instance to Cohen's book [6]. Since the seminal Lenstra-Lenstra-Lovász algorithm [12]—whose range of application is exceptional—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 [19, 31], of Schnorr [29], and references therein. Most of floating point variants lead to powerful heuristics, especially à la Schnorr-Euchner [30], that are implemented (often with improvements) 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 [12].

We propose here an algorithm that takes in 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. In the Introduction we have seen that this consists in testing the two conditions (2) and (3). Let R be the QR factor of A. If the a_j are proper, we mean

$$|r_{i,j}|/r_{i,i} \leq \eta,\ 1 \leq i < j \leq n, \tag{23}$$

and if the Lovász conditions

$$\sqrt{\delta - (r_{i,i+1}/r_{i,i})^2}\ r_{i,i} \leq r_{i+1,i+1},\ 1 \leq i \leq n-1, \tag{24}$$

are satisfied, then the basis $a_1, a_2, \ldots, a_n$ of $\mathcal{L}$ is called LLL reduced with parameters (δ, η). The latter satisfy $1/4 < \delta \leq 1$ and $1/2 \leq \eta < \sqrt{\delta}$.

The principle of the algorithm is to compute an approximate $\widetilde{R}$ together with error bounds (using the floating point algorithm of Section 6), then to test (23) and (24).

The entries of A are integers of arbitrary size (our implementation relies on Gmp [17]). Therefore the entries of A may not be represented exactly by elements in $\mathbb{F}$. Nevertheless, for the computation of an approximate $\widetilde{R}$ we may take $\widetilde{A}$ by direct conversion to $\mathbb{F}$. Since the error is very small and $\widetilde{R}$ will be an approximation anyway, this does not really influence the quality of subsequent computations. Then $\widetilde{R}$ is computed by the Modified Gram-Schmidt algorithm. Once $\widetilde{R}$ is known we apply Theorem 6.1 for computing a certified error bound. The only expression that has to be bounded with A involved is in (16), where the computation of AV using the program (21) 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 $A_-, A_+ \in \mathbb{F}^{n\times n}$ (see the certified techniques in Section 6.1), and we evaluate A_-V and A_+V in (21). Therefore the error bound $F \in \mathbb{F}^{n\times n}$ we compute by Theorem 6.1 is actually such that $|R - \widetilde{R}| \leq F$ for R the QR factor of any $A \in [A_-, A_+]$.

Once F is known, for fixed i and j, we test (23) by resorting to the bounding techniques of Section 6.1:

$$\begin{array}{ll} \text{setround(down);} & \underline{\eta} = \text{fl}(\eta);\quad t_i = \text{fl}((r_{i,i} - f_{i,i}) \times \underline{\eta}) \\ \text{setround(up);} & t_j = \text{fl}(|r_{i,j}| + f_{i,j}) \\ & \text{test } t_j \leq t_i? \end{array} \tag{25}$$

with temporary variables t_i and t_j. Recall that the diagonal entries of R are positive. Similarly, for a fixed i, we test (24) using:

$$\begin{array}{ll} \text{setround(up);} & t_i = \text{fl}(r_{i,i} + f_{i,i});\ \overline{\delta} = \text{fl}(\delta); \\ \text{setround(down);} & t_{i+1} = \text{fl}(r_{i+1,i+1} - f_{i+1,i+1}) \\ & t = \text{fl}\left(((|r_{i,i+1}| - f_{i,i+1})/t_i)^2\right) - \delta;\ t = -t; \\ \text{setround(up);} & t = \text{fl}(\sqrt{t} \times t_i) \\ & \text{test } t \leq t_{i+1}? \end{array} \tag{26}$$

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 setround instructions. In addition to the $10n^3 + O(n^2)$ operations for computing F using Theorem 6.1, the reducedness test essentially requires $2n^3 + O(n^2)$ operations for computing an approximate factor $\widetilde{R}$. This gives the following.

Theorem 7.1. *Let $A \in \mathbb{Z}^{n\times n}$ invertible and parameters (δ, η) 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 (δ, η), or returns "failed".*

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, and may be appended to any reduction heuristic program.

Let us now report some experiments. As previously in the paper all codes are run using 64 bits floating point numbers. The effectiveness of the certificate essentially relies on the effectiveness of the error bounding algorithm. We have manipulated lattices using Magma [13], the LLL reduction implementation is based on the work of Nguyen and Stehlé [19, 31]. The first family of reduced bases—matrices A—we consider are obtained by the reduction of $n \times n$ random integer matrices. The bases are reduced for the classical LLL parameters $(\delta, \eta) = (3/4, 1/2)$ in Figure 7.1, and $(\delta, \eta) = (0.99, 0.5001)$ for a stronger reduction in Figure 7.2.

Dimension	40	200	500	1000
$\kappa_\infty(A)$	4.7×10^2	2.4×10^4	1.8×10^5	9×10^5
$t_k - t = \min_i\{t_{i+1} - t\}$ in (26)	18	10	13	23
Certified absolute error on $\|a_k^*\|_2$	7.5×10^{-12}	3×10^{-10}	1.5×10^{-9}	1.2×10^{-8}
Certified $\max_{ij} \mu_{ij}$	0.4997	0.499994	0.49991	0.49999
Max. certified relative error on $\|r_{ij}\|$	2.8×10^{-11}	8.6×10^{-9}	1.5×10^{-7}	3×10^{-5}

Figure 7.1: Reducedness certificate output on $(3/4, 1/2)$-reduced bases from random integer matrices with entries on 10^3 bits, $\max|a_{ij}| \leq 1000$.

Since the numerical quality of the tested bases is good ($\kappa_\infty(A) \leq 10^6$), the reducedness certificate is highly efficient. We mean that the certified error is very small, and hence the tests are passed except in exceptional cases. Figures 7.1 and 7.2 for instance look at the smallest difference $t_k - t$ whose positiveness has to be certified in (26). The certificate has lots of room since the absolute errors on t and $t_k = \|a_k^*\|_2$ are much smaller. Exceptional cases will rather occur when testing properness. Indeed, testing reducedness may be an ill-posed problem because of the possible equalities in (23) and (24). 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 (δ, η)-reduced basis is a $(\delta - \epsilon_1, \eta + \epsilon_2)$-reduced basis for small ϵ_1, ϵ_2. The latter does really affect the relevant certified informations provided by the reduction.

Dimension	40	200	500	1000
$t_k - t = \min_i\{t_{i+1} - t\}$ in (26)	4.8×10^{-2}	7.7×10^{-2}	5.3×10^{-2}	7.3×10^{-2}
Certified absolute error on $\|a_k^*\|_2$	9.4×10^{-14}	6×10^{-12}	4×10^{-11}	2×10^{-10}

Figure 7.2: Reducedness certificate output 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 [2]. We have considered reduced bases with parameters $(3/4, 1/2)$. These bases may have integer entries as large as 10^{80}. The certificate has always succeeded. On a 18×18 example, with $\kappa_\infty(A) \approx 4 \times 10^{12}$, the smallest difference $t - t_k$ has been around 2.4×10^{76} with certified absolute error 1.95×10^{62}. The maximum of the μ_{ij} has been certified to be less than 0.493. On an example with $n = 31$ and $\kappa_\infty(A) \approx 8 \times 10^{13}$, we have certified an absolute error 3.2×10^{53} for $t - t_k \approx 1.7 \times 10^{67}$. On the latter example we have also checked that $\max \mu_{ij} \leq 0.4991$, thanks to a maximum relative error $|\widetilde{R} - R|$ certified to be less than 0.2 (only 6×10^{-15} on the diagonal).

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 [20, §3.4]. In the experiments reported 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 (matrices A) 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 κ_∞ in Figure 7.3 (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.

Dimension	75	100	125	150	175
$(\delta, \eta) = (3/4, 1/2)$, $\kappa_\infty(A)$	6×10^5	5.2×10^6	2.3×10^8	1.3×10^{10}	2×10^{11}
$t_k - t = \min_i\{t_{i+1} - t\}$ in (26)	1.3×10^{37}	8.5×10^{26}	4.2×10^{20}	3×10^{15}	1.2×10^{12}
Max. certified relative error on $\|r_{ij}\|$	1.3×10^{-9}	3.4×10^{-8}	2.2×10^{-6}	2.1×10^{-5}	6.3×10^{-3}
$(\delta, \eta) = (0.99, 0.5001)$, $\kappa_\infty(A)$	2.4×10^4	4.6×10^5	4×10^5	4×10^7	9×10^8
Max. certified relative error on $\|r_{ij}\|$	5.1×10^{-10}	2.5×10^{-9}	3.9×10^{-8}	6×10^{-7}	9.5×10^{-6}

Figure 7.3: Reducedness certificate output on "random" reduced bases from knapsack problems, $\max |a_{ij}|$ goes from 10^{45} ($n = 50$) down to 10^{25} ($n = 175$).

Beyond dimension 175 with this type of reduced basis, the certificate starts to fail more often. 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 (26) seems to fail later thanks to much better error bounds on the diagonal in general. 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 in (5.3). Nevertheless, on a typical example of dimension 300 with a $(0.99, 0.5001)$ reduced basis, the error bounding algorithm remains effective ($\kappa_\infty \approx 2.5 \times 10^{13}$, $\|H\|_\infty \approx 0.6$). The certificate may not be able to certify the actual reducedness of the basis, for example with $\min_i\{t_i - t\} \approx -4.12 \times 10^8$, and a too big absolute error bound 4.42×10^8. 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 numerical limitations of the certificate are close to those identified in [20, Heuristic 4] for the reduction process itself. Indeed, on the knapsack bases, it is claimed in [20] that a precision $n/4 + o(n)$ should suffice when using the floating point reduction of [19]. This means that $n \approx 200$ is a barrier with a 53 bits precision (64 bits numbers). The eventuality of a link between both limitations deserves to be further investigated.

8 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. In particular, 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. In error bound computation and property certification, the foreground of our study is to understand the compromize between the cost and the quality/effectiveness of bounds and certificates. In our case for instance, may we hope for an $O(n^2)$ effective certificate? Various computer arithmetics come in the background, where floating point computation, multi-precision, verification identities, midpoint-radius intervals, and exact computation are collaborative tools.

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

To our knowledge, the minimum precision required for a proven LLL variants is $1.6n+o(n)$ with the L^2 algorithm of [19, 20] (for δ close to 1 and η close to $1/2$). Our experiments show we may certify reducedness for dimensions much higher than this worst-case limit ($n_{\max} \leq 53/1.6 \approx 33$). The certificate is therefore very effective for a use complementary to reduction heuristics in dimension greater than $n_{\max}$ with double precision. Noticing the fact that the certificate is sensitive to the numerical properties of the input basis, it is worth studying its extensions to reduction algorithms and reducedness certificates with adaptative precision, and sensitive to the numerical quality.

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

References

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

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

[3] R.A. Brualdi and S. Mellendorf. Regions in the complex plane containing the eigenvalues of a matrix. *Am. Math. Mon.*, 101(10):975–985, 1994.

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

[11] 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.

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

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

[14] Matlab. *User's Guide, Version 7.2*. The MathWorks, Inc., 2006.

[15] 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.

[16] 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 ON, Canada, 2005.

[17] Gnu MP. *The GNU Multiple Precision Arithmetic Library, Edition 4.2.1,* `http://www.swox.com/gmp`. 2006.

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

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

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

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

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

[23] 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.

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

[25] 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.

[26] 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.

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

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

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

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

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

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

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

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

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