published in BIT Numerical Mathematics, 46:433–452, 2006

VERIFICATION OF POSITIVE DEFINITENESS *

S.M. RUMP

Institute for Reliable Computing, Hamburg University of Technology,
Schwarzenbergstr. 95, 21071 Hamburg, Germany,
and Waseda University, Faculty of Science and Engineering,
2-4-12 Okubo, Shinjuku-ku, Tokyo 169-0072, Japan.
email: rump@tu-harburg.de

Abstract. We present a computational, simple and fast sufficient criterion to verify positive definiteness of a symmetric or Hermitian matrix. The criterion uses only standard floating-point operations in rounding to nearest, it is rigorous, it takes into account all possible computational and rounding errors, and is also valid in the presence of underflow. It is based on a floating-point Cholesky decomposition and improves a known result. Using the criterion an efficient algorithm to compute rigorous error bounds for the solution of linear systems with symmetric positive definite matrix follows. A computational criterion to verify that a given symmetric or Hermitian matrix is *not* positive definite is given as well. Computational examples demonstrate the effectiveness of our criteria.

Keywords and phrases. Positive definite, verification, self-validating methods, Cholesky decomposition, rigorous error bounds, INTLAB, semidefinite programming

AMS subject classification (2000). 65G20, 15A18

1. Introduction and notation. The aim of this paper is to derive a fast and rigorous criterion to prove positive semidefiniteness of a symmetric or Hermitian matrix. The method is applicable to large, sparse matrices. It includes the case where X is an interval matrix, in which case positive semidefiniteness is proved for all *symmetric (Hermitian)* $\widetilde{X} \in X$. A computable criterion to prove that a symmetric or Hermitian matrix is *not* positive definite is given as well. The well known book on deterministic global optimization by Floudas devotes a chapter to the computation of lower bounds for the smallest eigenvalue of all (symmetric) matrices $\widetilde{A}$ within an interval matrix [5, Chapter 12.4]. Our method is based on a single floating-point Cholesky decomposition.

The problem of verifying positive semidefiniteness of a singular positive semidefinite matrix is ill-posed: An arbitrarily small change in the input data can change the answer from yes to no. Therefore we verify *positive definiteness* rather than positive semidefiniteness. This is a principle of so-called self-validating methods (see [10] or Volume 324 of Linear Algebra and its Applications (LAA), which is devoted to self-validating methods).

There are various applications of verifying positive definiteness, for example in semidefinite programming problems [5]. Denote by $M_n(\mathbb{IK})$, $\mathbb{IK} \in \{\mathbb{IR}, \mathbb{C}\}$ the set of $n \times n$ matrices over $\mathbb{IK}$. A standard formulation of a semidefinite programming problem is

$$\text{minimize } \langle C, X\rangle \quad \text{for} \quad \langle A_i, X\rangle = b_i,\ 1 \le i \le m \quad \text{and} \quad X \succeq 0,$$

where $\langle C, X\rangle := \text{trace}(C^T X)$, and $X \succeq 0$ means that the (symmetric or Hermitian) matrix X is positive semidefinite. To check feasibility of some $X^* = X \in M_n(\mathbb{IK})$, we have to verify positive semidefiniteness of X, i.e. $x^* X x \ge 0$ for all $x \in \mathbb{IK}^n$.

Another application of our criterion is the computation of verified error bounds for large, sparse linear systems. The apparently only known effective method for that [10] relies on the verification of positive definiteness of a certain matrix.

Our method is based on standard IEEE 754 floating-point arithmetic with rounding to nearest. The main computational effort is one floating-point Cholesky decomposition. A major advantage of the method is that any library routine can be used. We extend and improve a result in [12] (see also [3, 6]). Moreover, we allow

*Received 02.01.2005 Revised 27.01.2005 Communicated by Per Christian Hansen

underflow so that our criterion is rigorous under all circumstances: provided the hardware and software work to their specifications, our criterion is like a mathematical proof.

We use standard notation for rounding error analysis [6]. For example, $\mathrm{fl}(\cdot)$ is the result of the expression within the parenthesis computed in rounding to nearest. Denote by $\mathbb{F}$ ($\mathbb{F}+i\mathbb{F}$) the set of real (complex) floating-point numbers with relative rounding error unit `eps` and underflow unit `eta`. In case of IEEE 754 double precision, `eps` $= 2^{-53}$ and `eta` $= 2^{-1074}$. Then

$$\begin{array}{lll}
a,b\in\mathbb{F}: & \mathrm{fl}(a\circ b)=a\circ b(1+\varepsilon_1) & \text{for } \circ\in\{+,-\},\ |\varepsilon_1|\le \texttt{eps}\\
a,b\in\mathbb{F}: & \mathrm{fl}(a\circ b)=a\circ b(1+\varepsilon_2)+\eta_2 & \text{for } \circ\in\{\cdot,/\},\ |\varepsilon_2|\le \texttt{eps},\ |\eta_2|\le \texttt{eta},\\
 & & \qquad\varepsilon_2\eta_2=0\\
0\le a\in\mathbb{F}: & \mathrm{fl}(a^{1/2})=a^{1/2}(1+\varepsilon_3) & |\varepsilon_3|\le \texttt{eps}\\
x,y\in\mathbb{F}+i\mathbb{F}: & \mathrm{fl}(x\pm y)=(x\pm y)(1+\varepsilon_4) & |\varepsilon_4|\le \texttt{eps}\\
x,y\in\mathbb{F}+i\mathbb{F}: & \mathrm{fl}(x\cdot y)=xy(1+\varepsilon_5)+\eta_5, & |\varepsilon_5|\le\sqrt{2}\gamma_2,\ |\eta_5|\le 2\sqrt{2}\texttt{eta}\\
x\in\mathbb{F}+i\mathbb{F}: & \mathrm{fl}(x^*x)=x^*x\cdot(1+\varepsilon_6)+\eta_6, & |\varepsilon_6|\le\gamma_2,\ |\eta_6|\le 2\texttt{eta}\\
x\in\mathbb{F}+i\mathbb{F},\ b\in\mathbb{F}: & \mathrm{fl}(x/b)=x/b\cdot(1+\varepsilon_7)+\eta_7, & |\varepsilon_7|\le \texttt{eps},\ |\eta_7|\le\sqrt{2}\texttt{eta}
\end{array} \tag{1.1}$$

where, as usual,

$$\gamma_k := \frac{k\texttt{eps}}{1-k\texttt{eps}} \quad \text{for} \quad k\ge 0. \tag{1.2}$$

Here x^* denotes the complex conjugate of x. Most of the properties are proved in [6, (2.4) and Lemma 3.5], the others follow easily. In a recent paper [1] it is shown that $\sqrt{2}\gamma_2$ for complex multiplication can be replaced by $\sqrt{5}$`eps` which is essentially sharp. Note that no underflow correction is necessary for addition and subtraction. We will use well known properties of γ_k such as $m\gamma_k\le\gamma_{mk}$. All our estimations will hold no matter what the order of evaluation.

We add a special remark to today's computers and architectures. When computing in double precision on a PC, intermediate results may be stored in extended precision depending on the setting of the control word. For example, the imaginary part of x^*x for complex $x=a+ib\in\mathbb{F}+i\mathbb{F}$ might be computed by the statements `Im=a*b; Im=Im-b*a`. If the first result `Im=a*b` is stored in double, but the intermediate result `b*a` accumulated in extended precision, then the imaginary part may be nonzero. To avoid such phenomena we assume throughout the paper that all floating-point results are computed and stored in one working precision, for example double precision.

2. Error estimation. Suppose $A^T=A\in M_n(\mathbb{F})$ or $A^*=A\in M_n(\mathbb{F}+i\mathbb{F})$. Up to different order of execution every variant of the Cholesky decomposition $R^*R=A$ follows the scheme [6, Algorithm 10.2]:

$$\begin{array}{l}
\text{for } j=1:n\\
\quad \text{for } i=1:j-1\\
\quad\quad r_{ij}=(a_{ij}-\sum_{k=1}^{i-1}r_{ki}^*r_{kj})/r_{ii}\\
\quad \text{end}\\
\quad r_{jj}=(a_{jj}-\sum_{k=1}^{j-1}r_{kj}^*r_{kj})^{1/2}\\
\text{end}
\end{array} \tag{2.1}$$

as implied by solving $R^*R=A$ for r_{ij}. Note that R is upper triangular. We say the decomposition "runs to completion" if all square roots are real. For the analysis of floating-point Cholesky decomposition we extend and improve the analysis in [3] to complex matrices and to include underflow. To obtain "nice" constants, we first extend the standard result [6, Lemma 8.4] to complex data and improve it for real data.

LEMMA 2.1. *For floating-point quantities a_i, b_i and c, where b_k is real, let*

$$\widetilde{y}=\mathrm{fl}((c-\sum_{i=1}^{k-1}a_ib_i)/b_k). \tag{2.2}$$

Assume $\gamma_{k+1}\le 1$. Then, no matter what the order of evaluation and allowing underflow,

$$|c-\sum_{i=1}^{k-1}a_ib_i-b_k\widetilde{y}| \ < \ \gamma_p\left(\sum_{i=1}^{k-1}|a_ib_i|+|b_k\widetilde{y}|\right) \ + \ 3\texttt{eta}\,(2k+|b_k|),$$

where $p = k$ for real data and $p = k + 1$ for complex data.

Remark. Note that we will prove (2.5) which implies

$$|c - \sum_{i=1}^{k-1} a_i b_i - b_k \widetilde{y}| \leq \gamma_{k-1} \sum_{i=1}^{k-1} |a_i b_i| + \gamma_k |b_k \widetilde{y}| \tag{2.3}$$

for real a_i, b_i if no underflow occurs. This is slightly better than the famous Lemma 8.4 in [6] and is needed to obtain "nice" constants in Theorem 2.3.

Proof. We use the standard scheme in [6, Lemma 8.4]. We first analyze the numerator in (2.2) and denote

$$\widetilde{s} = \mathrm{fl}(c - \sum_{i=1}^{k-1} a_i b_i).$$

The sum consists of k terms, the 0-th term $t_0 := c$ and the i-th term $t_i := -\mathrm{fl}(a_i b_i) = -(a_i b_i(1 + \varepsilon_i) + \eta_i)$ for $1 \leq i \leq k-1$, where $|\varepsilon_i| \leq \mathtt{eps}\,, |\eta_i| \leq \mathtt{eta}$ for real a_i, b_i, and $|\varepsilon_i| \leq \gamma_3, |\eta_i| \leq 2\sqrt{2}\mathtt{eta}$ for complex a_i, b_i. Let $\pi : \{0, \ldots, k-1\} \to \{0, \ldots, k-1\}$ be the permutation of terms such that $\widetilde{s}$ is actually computed by

$$\begin{array}{l} \widetilde{s} = t_{\pi(0)} \\ \text{for } i = 1 : k-1 \\ \qquad \widetilde{s} = \mathrm{fl}(\widetilde{s} + t_{\pi(i)}) \\ \text{end} \end{array}$$

Then

$$\widetilde{s} = \sum_{i=0}^{k-1} \Phi_{\pi(i)} t_{\pi(i)} \qquad \text{with} \quad \Phi_{\pi(0)} = \Phi_{\pi(1)} = 1 + \delta_{\pi(1)} \text{ and } \Phi_{\pi(i)} = \prod_{\nu=1}^{i} (1 + \delta_{\pi(\nu)}) \text{ for } i > 1,$$

where $|\delta_\nu| \leq \mathtt{eps}$ for real and complex a_i, b_i. Note that each $\Phi_{\pi(i)}$ consists of at least one and of at most $k-1$ factors $1 + \delta_{\pi(\nu)}$. Let $t_0 = c$ be the m-th term $t_{\pi(m)}$ in the computation of $\widetilde{s}$, then $0 \leq m \leq k-1$ and

$$\begin{aligned} \frac{\widetilde{s}}{\Phi_{\pi(m)}} &= c + \textstyle\sum_{\substack{i=0 \\ i \neq m}}^{k-1} \dfrac{\Phi_{\pi(i)}}{\Phi_{\pi(m)}} t_{\pi(i)} \\ &= c \;+\; \textstyle\sum_{i=0}^{m-1} \prod_{\nu=\max(2,i+1)}^{m} (1 + \delta_{\pi(\nu)})^{-1} t_{\pi(i)} \;+\; \sum_{i=m+1}^{k-1} \prod_{\nu=\max(2,m+1)}^{i} (1 + \delta_{\pi(\nu)}) t_{\pi(i)} \; . \end{aligned}$$

This implies

$$\widetilde{s}(1 + \Theta_{k-1}) \;=\; c + \sum_{i=1}^{k-1} (1 + \Theta_{k-2}^{(i)}) t_i \tag{2.4}$$

for real and for complex data, where $|\Theta_{k-1}| \leq \gamma_{k-1}$ and $|\Theta_{k-2}^{(i)}| \leq \gamma_{k-2}$. Furthermore,

$$\widetilde{y} \;=\; \mathrm{fl}(\widetilde{s}/b_k) \;=\; \widetilde{s}/b_k \cdot (1 + \varepsilon_k) + \eta_k \; ,$$

where $|\varepsilon_k| \leq \mathtt{eps}\,, |\eta_k| \leq \mathtt{eta}$ for real a_i, b_i, and, because b_k is real, $|\varepsilon_k| \leq \mathtt{eps}\,, |\eta_k| \leq \sqrt{2}\mathtt{eta}$ for complex a_i, b_i. Hence

$$\frac{\widetilde{y} b_k (1 + \Theta_{k-1})}{1 + \varepsilon_k} = c - \sum_{i=1}^{k-1} (1 + \Theta_{k-2}^{(i)})(a_i b_i (1 + \varepsilon_i) + \eta_i) + \frac{b_k \eta_k (1 + \Theta_{k-1})}{1 + \varepsilon_k} \; . \tag{2.5}$$

This implies the result for real a_i, b_i and for complex a_i, b_i . ∎

Lemma 2.2. *For $a_i \in \mathbb{IF} + i\mathbb{IF}, c \in \mathbb{IF}$ assume $\gamma_{k+1} < 1$ and let*

$$\widetilde{s} = \mathrm{fl}(c - \sum_{i=1}^{k-1} a_i^* a_i) \geq 0 \quad \textit{and} \quad \widetilde{y} = \mathrm{fl}(\widetilde{s}^{1/2}).$$

Then, also in the presence of underflow,

$$|c - \sum_{i=1}^{k-1} a_i^* a_i - \widetilde{y}^2| < \gamma_{k+1} \left(\sum_{i=1}^{k-1} a_i^* a_i + \widetilde{y}^2 \right) + 4k\texttt{eta}$$

and

$$\widetilde{y}^2 + \sum_{i=1}^{k-1} a_i^* a_i \le (1 - \gamma_{k+1})^{-1} c.$$

PROOF. Note that $\mathrm{fl}(a_i^* a_i)$ is real because real floating-point multiplication is commutative, and

$$\mathrm{fl}(a_i^* a_i) = \mathrm{fl}((\mathcal{R}e\, a_i)^2 + (\mathcal{I}m\, a_i)^2) = a_i^* a_i (1 + \varepsilon_i) + \eta_i \ge 0 \tag{2.6}$$

with $|\varepsilon_i| \le \gamma_2$, $|\eta_i| \le 2\texttt{eta}$ for real and for complex a_i. We proceed similarly as before with $a_i b_i = a_i^* a_i$, so that

$$\widetilde{y} = \mathrm{fl}(\widetilde{s}^{1/2}) = \widetilde{s}^{1/2}(1 + \varepsilon), \quad |\varepsilon| \le \texttt{eps}$$

implies

$$\widetilde{y}^2 = \widetilde{s}(1 + \Theta_2) \qquad \text{with } |\Theta_2| \le \gamma_2$$

for real and complex a_i and $\widetilde{s}$ as in (2.4). Now (2.6) implies $\eta_i \ge 0$, with or without underflow. Inserting in (2.4) yields

$$\frac{\widetilde{y}^2(1 + \Theta_{k-1})}{1 + \Theta_2} = c - \sum_{i=1}^{k-1} (1 + \Theta_{k-2}^{(i)})(a_i^* a_i (1 + \varepsilon_i) + \eta_i) \ ,$$

or

$$\widetilde{y}^2(1 + \Theta_{k+1}) = c - (1 + \Theta_k) \sum_{i=1}^{k-1} a_i^* a_i - \eta \quad \text{with} \quad 0 \le \eta \le 2(k-1)(1 + \gamma_k)\texttt{eta}$$

and $|\Theta_{k+1}| \le \gamma_{k+1}$, $|\Theta_k| \le \gamma_k$. This implies the first inequality. Moreover $\eta \ge 0$ yields

$$\widetilde{y}^2 + \sum_{i=1}^{k-1} a_i^* a_i \le c + \gamma_{k+1}(\widetilde{y}^2 + \sum_{i=1}^{k-1} a_i^* a_i),$$

and the result follows. ■

Now let real $A^T = A \in M_n(\mathbb{F})$ or complex $A^* = A \in M_n(\mathbb{F} + i\mathbb{F})$ be given, and assume the Cholesky decomposition (2.1) *executed in floating-point* runs to completion. This implies $a_{jj} \ge 0$ and $\widetilde{r}_{jj} \ge 0$. Note that we do not assume A to be positive semidefinite and that underflow may occur. Then we can derive the following improved lower bound for the smallest eigenvalue of A.

THEOREM 2.3. *Let $A^T = A \in M_n(\mathbb{F})$ or $A^* = A \in M_n(\mathbb{F} + i\mathbb{F})$ be given. Denote the symbolic Cholesky factor of A by $\widehat{R}$. For $1 \le i, j \le n$ define*

$$s(i,j) := |\{k \in \mathbb{N} : 1 \le k < \min(i,j) \quad \textit{and} \quad \widehat{r}_{ki}\widehat{r}_{kj} \ne 0\}|$$

and denote

$$\alpha_{ij} := \begin{cases} \gamma_{s(i,j)+2} & \textit{if} \quad s(i,j) \ne 0 \\ 0 & \textit{otherwise.} \end{cases}$$

Suppose $\alpha_{jj} < 1$ for all j. With

$$d_j := ((1 - \alpha_{jj})^{-1} a_{jj})^{1/2} \qquad \textit{and} \qquad M := 3(2n + \max a_{\nu\nu})$$

define

$$0 \le \Delta(A) \in M_n(\mathbb{R}) \qquad \textit{by} \qquad \Delta(A)_{ij} := \alpha_{ij} d_i d_j + M\texttt{eta} \ .$$

Then, if the floating-point Cholesky decomposition of A runs to completion, the smallest eigenvalue $\lambda_{\min}(A)$ of A satisfies

$$\lambda_{\min}(A) > -\|\Delta(A)\|_2.$$

Remark 1. Note that $s(i,j)$ is an upper bound on the number of nontrivial multiplications in (2.1) to compute r_{ij}. An obvious upper bound is $s(i,j) \le \min(i,j) - 1$.

Remark 2. The matrix $\Delta(A)$ and therefore the lower bound for the smallest eigenvalue of A depends only on the diagonal elements and the sparsity pattern of A. If A is positive definite, then the nonnegative matrix $\Delta(A)$ has the same sparsity pattern as the Cholesky factor of A (ignoring zeros by cancelation).

Remark 3. The matrix $\Delta(A)$ is nonnegative and symmetric, so $\|\Delta(A)\| = \varrho(\Delta(A))$, that is the spectral radius and Perron root of $\Delta(A)$ coincide.

Remark 4. If floating-point Cholesky runs to completion, then $a_{jj} \ge 0$ and d_j is real.

Proof. Suppose the floating-point Cholesky factorization of A runs to completion, denote the computed Cholesky factor by $\widetilde{R}$ and its j-th column by $\widetilde{r}_j$. Then $\widetilde{r}_{jj} \ge 0$, and Lemma 2.2 and (2.1) imply

$$\|\widetilde{r}_j\|_2^2 \le (1-\alpha_{jj})^{-1} a_{jj} = {d_j}^2 \ . \tag{2.7}$$

Define $\delta A := A - \widetilde{R}^*\widetilde{R}$. Then (2.1), the symmetry of δA and Lemma 2.1 imply

$$|\delta a_{ij}| < \alpha_{ij} |\widetilde{r}_i^*| \, |\widetilde{r}_j| + 3\mathtt{eta}\,(2n + \max \widetilde{r}_{\nu\nu}) \qquad \text{for} \qquad i \ne j \tag{2.8}$$

for real A and for complex A, and by Lemma 2.2 this is also true for $i = j$. So (2.7), (2.8) and (2.1) yield

$$|\delta a_{ij}| < \alpha_{ij} \|\widetilde{r}_i\|_2 \|\widetilde{r}_j\|_2 + 3\mathtt{eta}\,(2n + \max a_{\nu\nu}) \le \alpha_{ij} d_i d_j + M\mathtt{eta} = \Delta(A)_{ij} \quad \text{for all} \quad i,j. \tag{2.9}$$

The matrix $\widetilde{R}^*\widetilde{R}$ is positive semidefinite and the eigenvalues of A and $A - \delta A$ differ at most by $\|\delta A\|_2 \le \| \, |\delta A| \, \|_2 \le \|\Delta(A)\|_2$ by applying Perron-Frobenius Theory to the nonnegative matrices $|\delta A| \le \Delta A$. The theorem is proved. ∎

With this result we can establish the following rigorous test on positive definiteness. The test can be executed in pure floating-point; to simplify matter we first use floating-point subtraction with rounding downwards.

Corollary 2.4. *With the notations of Theorem 2.3 assume $\|\Delta(A)\|_2 \le c \in \mathbb{F}$. Let $\widetilde{A} \in \mathbb{F}^{n\times n}$ be given with $\widetilde{a}_{ij} = a_{ij}$ for $i \ne j$ and $\widetilde{a}_{ii} \le a_{ii} - c$ for all i. If the floating-point Cholesky decomposition applied to $\widetilde{A}$ runs to completion, then A is positive definite.*

Proof. The floating-point Cholesky decomposition of $\widetilde{A}$ runs to completion, so $0 \le \widetilde{a}_{ii} \le a_{ii}$ for all i. The symmetric matrix $\Delta(A)$ in Theorem 2.3 depends only on the nonzero pattern and the diagonal of A, so $0 \le \Delta(\widetilde{A}) \le \Delta(A)$ and Perron-Frobenius Theory yield

$$\|\Delta(\widetilde{A})\|_2 = \varrho(\Delta(\widetilde{A})) \le \varrho(\Delta(A)) = \|\Delta(A)\|_2 \le c.$$

By assumption, $\widetilde{A} = A - cI - D$ with diagonal $D \ge 0$. Hence Theorem 2.3 proves

$$\lambda_{\min}(A) = c + \lambda_{\min}(A - cI) = c + \lambda_{\min}(\widetilde{A} + D) \ge c + \lambda_{\min}(\widetilde{A}) > c - \|\Delta(\widetilde{A})\|_2 \ge 0.$$

because $\widetilde{A}$ is symmetric. ∎

One way to obtain suitable $\widetilde{A}$ is directed rounding, which is available in INTLAB [11], the Matlab toolbox for verified computations. With little effort we can avoid this as shown by the following lemma. The proof uses the fact that $\mathrm{fl}(a \pm b) = a \pm b$ for $|a \pm b| \le \mathtt{eps}^{-1}\mathtt{eta}$, and (1.1) otherwise. Note that $\frac{1}{2}\mathtt{eps}^{-1}\mathtt{eta}$ is the smallest positive normalized floating-point number.

Lemma 2.5. *Let $a, b \in \mathbb{F}$ and $c = \mathrm{fl}(a \circ b)$ for $\circ \in \{+,-\}$. Define $\varphi := \mathtt{eps}\,(1 + 2\mathtt{eps}\,) \in \mathbb{F}$. Then*

$$\mathrm{fl}(c - \varphi|c|) \le a \circ b \le \mathrm{fl}(c + \varphi|c|). \tag{2.10}$$

Hence, we can define $\widetilde{A}$ by

$$\widetilde{a}_{ij} := \begin{cases} \mathrm{fl}(d - \varphi|d|) \quad \text{with} \quad d := \mathrm{fl}(a_{ii} - c) & \text{if } i = j \\ a_{ij} & \text{otherwise,} \end{cases} \tag{2.11}$$

where $\varphi := \mathtt{eps}\,(1 + 2\mathtt{eps}\,) \in \mathbb{F}$.

We can use the results to prove that all symmetric (Hermitian) matrices within a real (complex) interval matrix are positive definite. For $A^* = A \in \mathbb{K}^{n\times n}$ and $0 \le R \in \mathbb{R}^{n\times n}$, $R^T = R$, $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$, denote

$$\langle A, R\rangle := \{\widetilde{X} \in \mathbb{K}^{n\times n} :\ \widetilde{X}^* = \widetilde{X},\ R^T = R, \quad \text{and} \quad |\widetilde{X} - A| \le R\} \tag{2.12}$$

where absolute value and comparison are to be understood entrywise. Then

$$\lambda_\nu(\widetilde{X}) \ge \lambda_\nu(A) - \|R\| \qquad \text{for all} \quad \widetilde{X} \in \langle A, R\rangle \text{ and all } \nu.$$

The radius matrix, however, is real symmetric and nonnegative, so $\|R\| = \varrho(R)$ is the Perron root of R. This can easily estimated by the well-known Lemma by Collatz [2], [7, Theorem 8.1.26].

Lemma 2.6. *Let $0 \le A \in M_n(\mathbb{R})$ and $x \in \mathbb{R}^n$ with $x_i > 0$ for $1 \le i \le n$. Then*

$$\min_i \frac{(Ax)_i}{x_i} \le \varrho(A) \le \max_i \frac{(Ax)_i}{x_i}.$$

For irreducible $A \ge 0$, the iteration $y = Ax^{(k)}$, $x^{(k+1)} = y/\|y\|$ produces for arbitrary starting vector $x^{(0)} > 0$ a strictly decreasing sequence $r^{(k)} := \max y_\nu/x_\nu^{(k)}$ converging to the Perron root [13]. Usually few iterations suffice to produce a good approximation. The quality, i.e. lower and upper bounds for $\varrho(A)$, and thus a stopping criterion follow by Lemma 2.6.

Corollary 2.7. *Let $A^T = A \in M_n(\mathbb{F})$ or $A^* = A \in M_n(\mathbb{F} + i\mathbb{F})$ and $0 \le R \in M_n(\mathbb{F})$, $R^T = R$ be given. With the notation of Theorem 2.3 assume $\|\Delta(A)\|_2 \le c \in \mathbb{F}$ and $\|R\|_2 \le r \in \mathbb{F}$. Let $\widetilde{A} \in \mathbb{F}^{n\times n}$ be given with $\widetilde{a}_{ij} = a_{ij}$ for $i \ne j$ and $\widetilde{a}_{ii} \le a_{ii} - c - r$ for all i. If the floating-point Cholesky decomposition applied to $\widetilde{A}$ runs to completion, then every matrix $\widetilde{X} \in \langle A, R\rangle$ as defined by (2.12) is symmetric positive definite.*

Note that Corollary 2.7 is a sufficient criterion for positive definiteness of all symmetric (Hermitian) matrices within $\langle A, R\rangle$. Also note that establishing a necessary and sufficient criterion is an NP-hard problem [9].

We can also use the previous results to verify that a symmetric (Hermitian) matrix is *not* positive semidefinite, i.e. has a negative eigenvalue. For this we need a "converse" of Theorem 2.3 improving a result in [3], see also [6, Theorem 10.7].

Theorem 2.8. *Let $A^T = A \in M_n(\mathbb{F})$ or $A^* = A \in M_n(\mathbb{F} + i\mathbb{F})$ be given. Assume that floating-point Cholesky decomposition of A ends prematurely. Then, with the notation of Theorem 2.3,*

$$\lambda_{\min}(A) < \|\Delta(A)\|_2.$$

Proof. Without loss of generality we may assume $a_{ii} \ge 0$ for all i, otherwise $\lambda_{\min}(A) < 0$. Suppose that floating-point Cholesky decomposition finishes successfully stages $1 \ldots k-1$ and computes

$$\widetilde{s} = \mathrm{fl}(a_{kk} - \sum_{\nu=1}^{k-1} \widetilde{r}_{\nu k}^* \widetilde{r}_{\nu k}) < 0$$

in stage k. In contrast to (2.1) we define $\widetilde{r}_{kk} := 0$ and denote by $A_k, \widetilde{R}_k$ the upper left $k \times k$-matrix in $A, \widetilde{R}$, respectively. We proceed as in the proof of Theorem 2.3 and note that for $1 \le i, j \le k$ the estimation of $|\delta a_{ij}|$ in (2.9) does not depend on $\widetilde{r}_{kk}$ except for the indices $i = j = k$. Therefore

$$|a_{ij} - (\widetilde{R}_k^* \widetilde{R}_k)_{ij}\ | < \Delta(A)_{ij} \tag{2.13}$$

at least for all $1 \le i, j \le k$ except $i = j = k$. Suppose $a_{kk} \ge (\widetilde{R}_k^* \widetilde{R}_k)_{kk} + \Delta(A)_{kk}$. Setting $m := s(k, k)$ and using (2.4) implies

$$\widetilde{s}(1 + \Theta_{m-1}) = a_{kk} - \sum_{\nu=1}^{k-1} (1 + \Theta_{m-2}^{(\nu)})(\widetilde{r}_{\nu k}^* \widetilde{r}_{\nu k}(1 + \varepsilon_\nu) + \eta_\nu)$$

with $|\Theta_{m-1}| \le \gamma_{m-1}$, $|\Theta_{m-2}^{(\nu)}| \le \gamma_{m-2}$, $|\varepsilon_\nu| \le \gamma_2$ and $0 \le \eta_\nu \le 2\texttt{eta}$. Hence

$$\begin{aligned} 0 > \widetilde{s}(1+\Theta_{m-1}) &\ge a_{kk} - (1+\gamma_m)(\widetilde{R}_k^*\widetilde{R}_k)_{kk} - 2k(1+\gamma_m)\texttt{eta} \\ &\ge \Delta(A)_{kk} - \gamma_m(\widetilde{R}_k^*\widetilde{R}_k)_{kk} - M\texttt{eta} \\ &\ge \alpha_{kk}(1-\alpha_{kk})^{-1}a_{kk} + M\texttt{eta} - \gamma_m(\widetilde{R}_k^*\widetilde{R}_k)_{kk} - M\texttt{eta} \\ &\ge 0 \end{aligned}$$

using $\alpha_{kk}(1-\alpha_{kk})^{-1} \ge \gamma_m$. This contradiction shows $a_{kk} < (\widetilde{R}_k^*\widetilde{R}_k)_{kk} + \Delta(A)_{kk}$.

If $a_{kk} \ge (\widetilde{R}_k^*\widetilde{R}_k)_{kk}$, then (2.13) is also true for $i = j = k$. By construction, $\widetilde{R}_k^*\widetilde{R}_k$ is singular and of course positive semidefinite, so interlacing implies

$$\lambda_{\min}(A) \le \lambda_{\min}(A_k) < \lambda_{\min}(\widetilde{R}_k^*\widetilde{R}_k) + \|\Delta(A)\|_2 = \|\Delta(A)\|_2 \ . \tag{2.14}$$

If $a_{kk} < (\widetilde{R}_k^*\widetilde{R}_k)_{kk}$, then define $\widetilde{A}_k \in M_k$ to be the matrix A_k with $\widetilde{a}_{kk} := (\widetilde{R}_k^*\widetilde{R}_k)_{kk}$. Then $\lambda_\nu(A_k) \le \lambda_\nu(\widetilde{A}_k)$ for all ν and

$$|\ \widetilde{a}_{ij} - (\widetilde{R}_k^*\widetilde{R}_k)_{ij}\ | < \Delta(A)_{ij} \qquad \text{for all} \quad 1 \le i,j \le k \ .$$

So we can proceed an in (2.14) and the lemma is proved. ∎

Corollary 2.9. *With the notations of Theorem 2.3 assume that $c \in \mathbb{F}$ is given with $\|\Delta(\widetilde{A})\|_2 \le c$, where $\widetilde{A} \in \mathbb{F}^{n\times n}$ satisfies $\widetilde{a}_{ij} = a_{ij}$ for $i \ne j$ and $\widetilde{a}_{ii} \ge a_{ii} + c$ for all i. If the floating-point Cholesky decomposition applied to $\widetilde{A}$ ends prematurely, then A is not positive semidefinite, i.e. has at least one negative eigenvalue.*

Remark. Note that the constant c depends on $\widetilde{A}$, but the computation of $\widetilde{A}$ also involves c, so determination of a suitable c is not obvious; an iteration for c starting with $\|\Delta(A)\|_2$ may be applied, but is costly. Another possibility is given in Corollary 3.1.

Proof of Corollary 2.9. By assumption, $\widetilde{A} = A + cI + D$ with diagonal $D \ge 0$. Hence Theorem 2.8 implies

$$\lambda_{\min}(A) = \lambda_{\min}(A+cI) - c = \lambda_{\min}(\widetilde{A} - D) - c \le \lambda_{\min}(\widetilde{A}) - c < \|\Delta(\widetilde{A})\|_2 - c \le 0$$

using that $\widetilde{A}$ is symmetric. ∎

If directed rounding is available, we can define $\widetilde{A} = \mathrm{fl}_\Delta(A + cI)$. Otherwise we can avoid directed rounding by using Lemma 2.5 and defining $\widetilde{A} \in \mathbb{F}^{n\times n}$ by

$$\widetilde{a}_{ij} := \begin{cases} \mathrm{fl}(d + \varphi|d|) \quad \text{with} \quad d := \mathrm{fl}(a_{ii} + c) & \text{if } i = j \\ a_{ij} & \text{otherwise,} \end{cases} \tag{2.15}$$

where again $\varphi := \texttt{eps}\,(1 + 2\texttt{eps}\,) \in \mathbb{F}$.

3. Practical application. For the practical application of Corollary 2.4 we need an upper bound of $\|\Delta(A)\|_2$. Note that the matrix $\Delta(A)$ is symmetric and nonnegative, so the spectral radius $\varrho(A)$ and the spectral norm $\|\Delta(A)\|_2$ coincide. Moreover, Perron-Frobenius Theory tells that $\|\Delta(A)\|_2 = \varrho(\Delta(A)) \le \varrho(B)$ for any matrix B with $\Delta(A) \le B$ (componentwise), and an upper bound follows by Lemma 2.6. Therefore we will construct such matrices B which are easy to compute.

The practical success of Corollary 2.4 depends on the quality of this bound $\varrho(B)$ of $\|\Delta(A)\|_2$. By the previous considerations any upper bound on $s(i,j)$, the number of nontrivial products necessary to compute $\widetilde{r}_{ij}$, determines a valid upper bound on $\|\Delta(A)\|_2$. The more we invest in bounds on $s(i,j)$, the better the bound on $\|\Delta(A)\|_2$ and the better the criterion. The simplest bound is

$$s(i,j) \le n-1 \quad \text{and therefore} \quad \alpha_{ij} \le \gamma_{n+1} \quad \text{for all} \quad i,j. \tag{3.1}$$

Using (3.1), Theorem 2.3 implies the simple bound

I) $$\|\Delta(A)\|_2 \le \|\gamma_{n+1}dd^T\| + nM\texttt{eta} = \gamma_{n+1}d^Td + nM\texttt{eta} = \gamma_{n+1}(1-\gamma_{n+1})^{-1}\mathrm{tr}(A) + nM\texttt{eta}\,,$$ ∎

where $d = (d_1, \ldots, d_n) \in \mathbb{R}^n$ and $\mathrm{tr}(A)$ denotes the trace of A. This is basically the bound in [3] and [6, Lemma 10.5]. A similar result already occurs in [12, Theorem 3.5][1], where it is formulated neglecting higher order terms. However, for sparse matrices this is in general fairly weak. Better bounds are obtained using the fact that nonzero elements of R must be inside the envelope of A. For a matrix A with nonzero diagonal define

$$t_j := j - \min\{i : \ a_{ij} \neq 0\}. \tag{3.2}$$

This is the number of nonzero elements above the diagonal in the j-th column of A. It is $0 \leq t_j \leq n-1$ for all j, and Cholesky decomposition (2.1) implies

$$s(i,j) \leq \min(t_i, t_j) \qquad \text{for all} \quad i,j.$$

Defining

$$\delta_i := ((1-\beta_i)^{-1}\beta_i a_{ii})^{1/2} \qquad \text{with} \quad \beta_i := \gamma_{t_i+2} \tag{3.3}$$

implies $\alpha_{ij} d_i d_j \leq \delta_i \delta_j$, and $\delta := (\delta_1, \ldots, \delta_n) \in \mathbb{R}^n$ and using Theorem 2.3 yields

II) $$\|\Delta(A)\|_2 \leq \delta^T \delta + nM\mathtt{eta}\,.$$

Both bounds I) and II) have the advantage that they are very easy to compute, without an extra (symbolic) factorization of A. Bound II) can be further improved. For the Cholesky factor R of A define

$$t'_j := |\{i < j : \ r_{ij} \neq 0\}|,$$

the number of nonzero elements in the j-th column of R. Since R is within the envelope of A and by the definition of $s(i,j)$, we have

$$s(i,j) \leq \min(t'_i, t'_j) \leq \min(t_i, t_j),$$

and replacing β_i by $\beta'_i := \gamma_{t'_i+2}$ in the definition (3.3) of δ_i implies

III) $$\|\Delta(A)\|_2 \leq \delta'^T \delta' + nM\eta \quad \text{ for } \delta'_i := ((1-\beta'_i)^{-1}\beta'_i a_{ii})^{1/2},\ \beta'_i := \gamma_{t'_i+2}$$

where $\delta' := (\delta'_1, \ldots, \delta'_n$.

All bounds I), II) and III) require only $\mathcal{O}(n)$ operations, whereas the computation of $\|\Delta(A)\|_2$ using its original definition in Theorem 2.3 requires to form the (sparse) matrix $\Delta(A)$ and some iterations using Collatz's Lemma [2, 13] to compute an upper bound for $\varrho(\Delta(A))$.

Bounds I) and II) can be directly computed from the original data, whereas bound III) requires information from the (symbolic) Cholesky decomposition. Usually the latter is performed anyway in order to minimize fill-in, so this represents no extra effort. The Matlab routine `symbfact` does not give the required information, unfortunately. It only yields the number of elements per *rows* of the Cholesky factor R, not as required of the columns.

We tested the methods on various matrices out of the Harwell-Boeing matrix market [4]. To improve the quality of the bounds one may

a) reorder the matrix and/or
b) scale the matrix.

Suitable reordering would be simple column ordering, (reverse) Cuthill-McKee, or minimum degree. According to van der Sluis' result [6, Corollary 7.6],

$$A \to DAD \qquad \text{with} \quad (DAD)_{ii} \approx 1. \tag{3.4}$$

seems to be a reasonable choice for scaling. We choose proper powers of 2 near $a_{ii}^{-1/2}$ to avoid rounding errors. Various tests with many test matrices did not show a panacea, but suggested that it seems a good choice to scale according to (3.4) if the variation of a_{ii} is larger than n, and to apply minimum degree reordering. Based on that we propose the following algorithm to verify positive definiteness of a symmetric or Hermitian matrix. The criterion is completely rigorous although we use only rounding to nearest.

[1] thanks to P. Batra for pointing to this reference.

```
function res = isspd(A)
%ISSPD        logical function: Matrix A is positive definite
%
%Given real symmetric or Hermitian complex matrix A,
%
%   res   1  Matrix A is proved to positive definite
%         0  positive definiteness could not be verified
%

% constants
n = dim(A); Eps = 2^(-53); Eta = 2^(-1074);

% diagonal check
if any( diag(A)<=0 )
    res = 0; return
end

% scaling
d = 2.^(-ceil(0.5*log2(diag(A))));
maxdiagA = max(d); mindiagA = min(d);
if ( maxdiagA/mindiagA>sqrt(n) ) & ~( ( maxdiagA>1e100 ) |
        ( mindiagA<1e-100 ) )        % apply van der Sluis scaling
   D = spdiags( d ,0,n,n );          % D_ii are powers of 2
   A = D*A*D;                        % 0.25 <= abs(A_ii) < 1
   maxdiagA = 1;
end

% Minimum degree sorting
p = symamd(A); A = A(p,p);

[i,j] = find(A); index = find(diff(j));
t = [0 ; (2:n)'-i(index+1)];        % max #elts left of (or above) diag(A)
if any ( t>3e15 )                   % dimension check, make sure alpha<1
   res = 0; return
end

% hull of A
if n<67108861                       % alpha_k/(1-alpha_k) < (k+1)*Eps
   alpha = (t+3)*Eps;               % exact
else
   alpha = (t+2)*Eps;  alpha = ( alpha./(1-alpha) )/(1-2*Eps);
   alpha = ( alpha./(1-alpha) )/(1-3*Eps)
end
d = sqrt(alpha.*diag(A))/(1-3*Eps);

% Upper bound for norm(dA) and shift
c = ( (3*n)*(2*n+maxdiagA)/(1-3*Eps) ) * Eta;
c = ( (d'*d)/(1-(n+2)*Eps) + c )/(1-2*Eps);   % bound II)

% floating-point Cholesky
A = A - c*speye(n);  A = A-diag(diag(A)*(Eps*(1+2*Eps)))
[R,p] = chol(A);                    % floating-point Cholesky
res = ( p==0 );                     % p=0  <=>  successful completion
```

ALGORITHM 1. *Verification of positive definiteness*

Algorithm 1 is executable Matlab code [8]. The algorithm is also added to INTLAB [11], the Matlab toolbox for verified computations, where directed roundings are available and are used. Here also interval input $\mathbf{A}$ is allowed, in which case positive definiteness of *all* symmetric (Hermitian) $\widetilde{X} \in \mathbf{A}$ is verified by Corollary 2.7.

In Algorithm 1 we use only rounding to nearest but nevertheless the result is completely rigorous. To see this observe that we need upper bounds for `alpha`, `d` and `c`. Note that `alpha` is a vector. All operands in the computation of these constants are positive and cannot underflow, so we can use

$$0 < \mathrm{fl}(a \circ b) \cdot (1 - \texttt{eps}\,) \le a \circ b \le \mathrm{fl}(a \circ b)/(1 - \texttt{eps}\,) \qquad \text{for} \quad a, b \in \mathbb{F},\ 0 \le a \circ b \tag{3.5}$$

which follows by (1.1). For small enough `n` the computation of `alpha` $=$ `(t + 3) * Eps` is exact. For larger `n` the dimension check ensures that the final value of `alpha` is less than 1, otherwise the criterion cannot hold. For the estimation of rounding errors observe for $0 \le a, b, \mathrm{fl}(a \circ b) \in \mathbb{F}$

$$\begin{array}{rcl} a \circ b & \le & (1 - \texttt{eps}\,)^2 (a \circ b)/(1 - 2\texttt{eps}\,) \le (1 - \texttt{eps}\,)\mathrm{fl}(a \circ b)/(1 - 2\texttt{eps}\,) \\ & \le & \mathrm{fl}((a \circ b)/(1 - 2\texttt{eps}\,)) \end{array} \tag{3.6}$$

using $1 - 2\texttt{eps} \in \mathbb{F}$. We introduce intermediate variables in the computation of `alpha`:

```
c1 = (t+2)*Eps; c2 = 1-alpha;  c3 = (c1./c2)/(1-2*Eps);
c4 = 1-c3;  alpha = (c3./c4)/(1-3*Eps);
```

In the notation of Theorem 2.3 denote $k := s(j, j)$ for fixed $j \in \{1, \cdots, n\}$. Now k corresponds to `t`, the computations of `c1` and `c2` are exact, and (3.6) gives $\gamma_{k+2} \le$ `c3`. Again using (3.6) yields

$$\begin{array}{rcl} \texttt{c3}/(1 - \texttt{c3}) & \le & (1 - \texttt{eps}\,)^3 \texttt{c3}/(1 - \texttt{c3})/(1 - 3\texttt{eps}\,) \le (1 - \texttt{eps}\,)^2 \texttt{c3}/\mathrm{fl}(1 - \texttt{c3})/(1 - 3\texttt{eps}\,) \\ & \le & (1 - \texttt{eps}\,)\mathrm{fl}(\texttt{c3}/(1 - \texttt{c3}))/(1 - 3\texttt{eps}\,) \le \mathrm{fl}((\texttt{c3}/(1 - \texttt{c3}))/(1 - 3\texttt{eps}\,)) = \texttt{alpha} \end{array}$$

and implies $\alpha_{jj}/(1-\alpha_{jj}) \le$ `alpha`. The estimation of `d` and `c` follows along the same lines using the standard estimation $|\sum_{i=1}^n d_i^2 - \mathrm{fl}(\sum_{i=1}^n d_i^2)| \le \gamma_n \sum_{i=1}^n d_i^2$ [6, (3.5)].

With the simplified criterion II) we can also computationally verify that a matrix is *not* positive semidefinite, i.e. has at least one negative eigenvalue. Following (3.3) define $\beta_i' = \beta_i(1 - \beta_i)^{-1}$, then II) reads

$$\|\Delta(A)\|_2 \le \sum_{i=1}^n \delta_i^2 + nM\texttt{eta} = \sum_{i=1}^n \beta_i' a_{ii} + nM\texttt{eta}\,. \tag{3.7}$$

For a positive constant $c \in \mathbb{F}$ to be set define $\widetilde{A} = \mathrm{fl}_\Delta(A + cI)$, that is a shift with rounding upwards. Then $\widetilde{a}_{ii} = (a_{ii} + c)(1 + \varepsilon_i)$ with $0 \le \varepsilon_i \le \texttt{eps}$ for all i. Define $\beta_i'' := \beta_i'(1 + \texttt{eps}\,)$, assume $\sum \beta_i'' < 1$ and let $c \in \mathbb{F}$ be such that

$$c \ge (1 - \sum_{i=1}^n \beta_i'')^{-1} (\sum_{i=1}^n \beta_i'' a_{ii} + nM\texttt{eta}\,). \tag{3.8}$$

Then a little computation using (3.7) shows

$$\|\Delta(\widetilde{A})\|_2 \le \sum_{i=1}^n \beta_i'(a_{ii} + c)(1 + \texttt{eps}\,) + nM\texttt{eta} \le c.$$

Now suppose floating-point Cholesky decomposition of $\widetilde{A}$ ends prematurely. Then $\widetilde{A} = A + cI + D$ with diagonal $D \ge 0$ and Theorem 2.8 implies

$$\lambda_{\min}(A) = \lambda_{\min}(\widetilde{A} - D) - c \le \lambda_{\min}(\widetilde{A}) - c < \|\Delta(\widetilde{A})\|_2 - c \le 0.$$

Corollary 3.1. *Let symmetric $A \in M_n(\mathbb{F})$ or Hermitian $A \in M_n(\mathbb{F} + i\mathbb{F})$ be given. With t_j as in (3.2) define*

$$\beta_i := \gamma_{t_i+2}, \quad \beta_i' := \beta_i(1 - \beta_i)^{-1} \quad \textit{and} \quad \beta_i'' := \beta_i'(1 + \texttt{eps}\,)$$

for $i \in \{1, \ldots, n\}$, assume $\sum_{i=1}^n \beta_i'' < 1$, and let $c \in \mathbb{F}$ with (3.8) be given. Let $\widetilde{A} := \mathrm{fl}_\Delta(A + cI)$ be the floating-point computation of $A + cI$ with rounding upwards. If floating-point Cholesky decomposition of $\widetilde{A}$ ends prematurely, then the matrix A has at least one negative eigenvalue. This statement is also true in the presence of underflow during Cholesky decomposition.

4. Computational results. For all following matrices, we first perform diagonal scaling as in Algorithm 1 and reordering by `symamd`, the improved minimum degree reordering algorithm in Matlab [8]. Furthermore, all matrices are normed to $\|A\|_1 \approx 1$ by a suitable power of 2 to have comparable results for different matrices.[2]

matrix	n	b	av	nnz(A)	$\|\Delta(A)\|_2$	ϱ_1	ϱ_2	ϱ_3
494bus	494	491	3.4	1666	$5.22\cdot10^{-15}$	2858.69	79.65	21.59
685bus	685	681	4.7	3249	$1.54\cdot10^{-14}$	1814.62	102.31	16.56
1138bus	1138	1137	3.6	4054	$9.06\cdot10^{-15}$	8381.51	187.94	28.60
nos1	237	156	4.3	1017	$6.78\cdot10^{-16}$	3936.69	78.18	61.55
nos2	957	636	4.3	4137	$6.78\cdot10^{-16}$	63985.74	318.93	249.80
nos3	960	952	16.5	15844	$2.90\cdot10^{-13}$	163.75	21.90	5.52
nos6	675	670	4.8	3255	$4.01\cdot10^{-14}$	622.80	57.73	9.39
nos7	729	719	6.3	4617	$6.10\cdot10^{-13}$	61.91	9.51	2.34
bcsstk08	1074	1057	12.1	12960	$5.49\cdot10^{-13}$	120.55	15.96	3.26
bcsstk09	1083	1042	17.0	18437	$9.56\cdot10^{-13}$	74.60	12.69	3.56
bcsstk10	1086	653	20.3	22070	$8.44\cdot10^{-14}$	792.63	111.00	16.37
bcsstk11	1473	1413	23.2	34241	$3.65\cdot10^{-13}$	356.67	42.19	8.49
bcsstk12	1473	1413	23.2	34241	$3.65\cdot10^{-13}$	356.67	42.19	8.49
bcsstk13	2003	1992	41.9	83883	$7.93\cdot10^{-12}$	30.91	6.76	2.27
bcsstk14	1806	1712	35.1	63454	$1.35\cdot10^{-12}$	137.58	17.34	4.55
bcsstk15	3948	3878	29.8	117816	$1.12\cdot10^{-11}$	82.83	13.31	3.59
bcsstk16	4884	4808	59.5	290378	$5.69\cdot10^{-12}$	231.25	39.41	7.39
bcsstk17	10974	10315	39.1	428650	$6.62\cdot10^{-11}$	109.69	11.89	1.00
bcsstk18	11948	11028	12.5	149090	$4.10\cdot10^{-11}$	224.99	9.83	1.00
bcsstk19	817	816	8.4	6853	$1.42\cdot10^{-14}$	2781.22	627.38	34.66
bcsstk20	485	454	6.5	3135	$6.29\cdot10^{-15}$	2368.19	84.41	30.13
bcsstk21	3600	1781	7.4	26600	$8.04\cdot10^{-13}$	1046.54	45.31	7.65
bcsstk22	138	109	5.0	696	$8.08\cdot10^{-15}$	134.36	15.37	6.02
bcsstk23	3134	3113	14.4	45178	$1.07\cdot10^{-11}$	53.99	10.14	2.46
bcsstk24	3562	3548	44.9	159910	$3.49\cdot10^{-12}$	223.30	22.62	4.98
bcsstk25	15439	15363	16.3	252241	$8.89\cdot10^{-11}$	163.21	12.85	1.00
bcsstk26	1922	1900	15.8	30336	$3.76\cdot10^{-13}$	575.28	42.14	6.83
bcsstk27	1224	1199	45.9	56126	$2.99\cdot10^{-13}$	276.13	80.28	11.10
bcsstk28	4410	4327	49.7	219024	$2.95\cdot10^{-12}$	388.25	45.13	7.13
bcsstk29	13992	13667	44.3	619488	$9.24\cdot10^{-11}$	117.21	7.48	1.00
bcsstk30	28924	28741	70.7	2043492	$4.20\cdot10^{-10}$	221.17	15.23	1.00
bcsstk31	35588	35437	33.2	1181416	$5.47\cdot10^{-10}$	256.89	10.55	1.00
bcsstk32	44609	44495	45.2	2014701	$5.84\cdot10^{-10}$	378.32	10.88	1.00
bcsstm10	1086	641	20.3	22092	$9.57\cdot10^{-14}$	768.55	113.67	16.41
bcsstm12	1473	981	13.3	19659	$1.54\cdot10^{-13}$	823.77	57.04	11.47
bcsstm27	1224	1199	45.9	56126	$3.46\cdot10^{-13}$	268.95	75.43	10.70
s1rmq4m1	5489	5347	47.8	262411	$9.40\cdot10^{-12}$	190.72	20.15	5.29
s1rmt3m1	5489	5486	39.7	217651	$5.25\cdot10^{-12}$	339.25	36.31	6.11
s2rmq4m1	5489	5396	48.0	263351	$1.05\cdot10^{-11}$	188.12	22.68	5.09
s2rmt3m1	5489	5346	39.7	217681	$6.40\cdot10^{-12}$	302.85	25.90	5.44
s3dkq4m2	90449	90160	49.0	4427725	$1.50\cdot10^{-9}$	290.89	9.55	1.00
s3dkt3m2	90449	89808	40.8	3686223	$1.37\cdot10^{-9}$	404.02	13.95	1.00
s3rmq4m1	5489	5480	47.9	262943	$9.53\cdot10^{-12}$	209.34	26.94	5.85

[2]All tests are performed on a Pentium M, 1.2GHz Laptop, Matlab Version 7.1 [8] and INTLAB Version 5.2 [11].

matrix	n	b	av	nnz(A)	$\|\Delta(A)\|_2$	ϱ_1	ϱ_2	ϱ_3
s3rmt3m1	5489	5487	39.7	217669	$5.83{\cdot}10^{-12}$	364.31	33.16	6.41
s3rmt3m3	5357	5325	38.7	207123	$4.62{\cdot}10^{-12}$	404.36	38.67	6.35
e40r0000	17281	17231	32.0	553216	$5.23{\cdot}10^{-11}$	251.34	11.15	1.00
fidapm11	22294	22057	27.7	617874	$4.04{\cdot}10^{-10}$	55.69	7.98	1.00

TABLE 4.1. $\|\Delta(A)\|_2$ *and bounds* I), II) *and* III)

The first table displays for various matrices out of [4].

name	the name of the Harwell-Boeing test matrix
n	the dimension
b	the bandwidth after reordering
av	the average number of nonzero elements per row
nnz	the total number of nonzero elements
$\|\Delta(A)\|_2$	for $\Delta(A)$ as given in Theorem 2.3
ϱ_1	the bound I) divided by $\|\Delta(A)\|_2$
ϱ_2	the bound II) divided by $\|\Delta(A)\|_2$
ϱ_3	the bound III) divided by $\|\Delta(A)\|_2$

The ratios ϱ_ν display the overestimation of criterion ν) compared to $\varrho(\Delta(A)) = \|\Delta(A)\|_2$. This applies to all data except for the matrices of dimension greater than 10000. These matrices where too big to perform the *symbolic* Cholesky factorization on our PC^2 to apply Corollary 2.4 to compute $\|\Delta(A)\|_2$. Bounds I), II) and III) could be computed without problem for all matrices. In the cases $n > 10000$, bound III) is displayed in column $\|\Delta(A)\|_2$ and the ratios refer to that bound.

The table shows that bound I) is sometimes a significant overestimation of $\|\Delta(A)\|_2$. The bounds II) and III) are frequently not too different. Note that bound II) is obtained with significantly less computational cost and memory. The following table shows the minimum, median, average and maximum of ϱ_1, ϱ_2 and ϱ_3 over all our test matrices (not only the ones displayed in Table 4.1).

	minimum	median	average	maximum
ϱ_1	2.4	241	1750	64000
ϱ_2	1.2	22	49	627
ϱ_3	1.2	5.4	13	250

TABLE 4.2. *Quality of bounds* I), II) *and* III)

Again, bound II) seems to be a reasonable compromise between quality and computational effort.

We may ask how close we can get to the smallest eigenvalue of a symmetric or Hermitian matrix by Algorithm 1 and its counterpart based on Corollary 3.1. For $s := \|A\|_1$ the matrix $A - sI$ has surely only nonpositive eigenvalues, and $A + sI$ is positive semidefinite. We bisect the interval $[-s, s]$ to find a narrow interval $[s_1, s_2]$ such that Algorithm 1 verifies positive definiteness of $A - s_1 I$, and its counterpart based on Corollary 3.1 verifies existence of at least one negative eigenvalue of $A - s_2 I$. It follows

$$s_1 < \lambda_{\min}(A) < s_2.$$

For the following Table 4.3 we first norm A by a suitable power of 2 to $\|A\|_1 \approx 1$, and display the name of the matrix, dimension n and average number av of elements per row, $\lambda_{\min}(A) \approx \frac{1}{2}(s_1 + s_2)$ and

$$acc := \frac{s_2 - s_1}{|s_1 + s_2|}.$$

For example we could calculate for the matrix "494bus" bounds for the smallest eigenvalue $\lambda_{\min}(A)$ coinciding to about 8 decimal figures. For some matrices (like "bcsstk17") the smallest eigenvalue is enclosed to almost maximum accuracy. Note that all matrices are scaled to $\|A\|_1 \approx 1$.

Since the matrices are normed to $\|A\|_1 \approx 1$, the reciprocal of $\lambda_{\min}(A)$ approximates the condition number of A. If Algorithm 1 verifies positive definiteness of $A - \underline{\lambda} I$ for $0 < \underline{\lambda} < \lambda_{\min}(A)$, then $\underline{\lambda}$ is a lower bound for

the smallest singular value of A and proves A to be nonsingular. For an approximate solution $\widetilde{x}$ of a linear system $Ax = b$ it follows

$$\|A^{-1}b - \widetilde{x}\|_2 \leq \underline{\lambda}^{-1}\|b - A\widetilde{x}\|_2, \tag{4.1}$$

so for the positive definite matrices in Table 4.3 bounds for the solution of a corresponding linear system can be computed by (4.1). For details, see [10].

matrix	n	b	av	$\lambda_{\min}(A)$	acc
494bus	494	491	3.4	$1.895504\cdot10^{-7}$	$4.58\cdot10^{-8}$
685bus	685	681	4.7	$1.888678\cdot10^{-6}$	$2.43\cdot10^{-8}$
1138bus	1138	1137	3.6	$5.366303\cdot10^{-8}$	$1.47\cdot10^{-6}$
nos1	237	156	4.3	$2.872064\cdot10^{-8}$	$1.80\cdot10^{-10}$
nos2	957	636	4.3	$1.122035\cdot10^{-10}$	$1.13\cdot10^{-8}$
nos3	960	952	16.5	$1.785976\cdot10^{-5}$	$2.29\cdot10^{-7}$
nos6	675	670	4.8	$1.192111\cdot10^{-7}$	$4.06\cdot10^{-14}$
nos7	729	719	6.3	$2.476047\cdot10^{-10}$	$9.85\cdot10^{-4}$
bcsstk08	1074	1057	12.1	$2.143796\cdot10^{-8}$	$6.41\cdot10^{-9}$
bcsstk09	1083	1042	17.0	$5.291573\cdot10^{-5}$	$8.75\cdot10^{-8}$
bcsstk10	1086	653	20.3	$1.271827\cdot10^{-6}$	$1.17\cdot10^{-7}$
bcsstk11	1473	1413	23.2	$2.760485\cdot10^{-9}$	$5.69\cdot10^{-5}$
bcsstk12	1473	1413	23.2	$2.760485\cdot10^{-9}$	$5.69\cdot10^{-5}$
bcsstk13	2003	1992	41.9	$3.232490\cdot10^{-11}$	$4.72\cdot10^{-8}$
bcsstk14	1806	1712	35.1	$5.820766\cdot10^{-11}$	$1.67\cdot10^{-16}$
bcsstk15	3948	3878	29.8	$1.164153\cdot10^{-10}$	$1.11\cdot10^{-16}$
bcsstk16	4884	4808	59.5	$1.164153\cdot10^{-10}$	$1.11\cdot10^{-16}$
bcsstk17	10974	10315	39.1	$2.910383\cdot10^{-11}$	$2.78\cdot10^{-16}$
bcsstk18	11948	11028	12.5	$1.806456\cdot10^{-12}$	$3.36\cdot10^{-8}$
bcsstk19	817	816	8.4	$5.095952\cdot10^{-12}$	$6.29\cdot10^{-4}$
bcsstk20	485	454	6.5	$1.798913\cdot10^{-13}$	$8.78\cdot10^{-6}$
bcsstk21	3600	1781	7.4	$2.687480\cdot10^{-8}$	$1.56\cdot10^{-6}$
bcsstk22	138	109	5.0	$6.298865\cdot10^{-6}$	$1.03\cdot10^{-9}$
bcsstk23	3134	3113	14.4	$2.368783\cdot10^{-13}$	$1.54\cdot10^{-3}$
bcsstk24	3562	3548	44.9	$2.237716\cdot10^{-12}$	$2.64\cdot10^{-4}$
bcsstk25	15439	15363	16.3	$1.067092\cdot10^{-13}$	$1.28\cdot10^{-2}$
bcsstk26	1922	1900	15.8	$3.470026\cdot10^{-9}$	$7.17\cdot10^{-7}$
bcsstk27	1224	1199	45.9	$1.711424\cdot10^{-5}$	$3.41\cdot10^{-9}$
bcsstk28	4410	4327	49.7	$7.582879\cdot10^{-10}$	$8.21\cdot10^{-4}$
bcsstk29	13992	13667	44.3	$-1.420591\cdot10^{-1}$	$2.04\cdot10^{-9}$
bcsstk30	28924	28741	70.7	$-1.037908\cdot10^{-1}$	$8.64\cdot10^{-9}$
bcsstk31	35588	35437	33.2	$-7.967106\cdot10^{-2}$	$7.28\cdot10^{-9}$
bcsstk32	44609	44495	45.2	$-1.260251\cdot10^{-1}$	$9.35\cdot10^{-9}$
bcsstm10	1086	641	20.3	$-3.144121\cdot10^{-2}$	$1.07\cdot10^{-10}$
bcsstm12	1473	981	13.3	$1.324195\cdot10^{-6}$	$1.69\cdot10^{-8}$
bcsstm27	1224	1199	45.9	$-7.273679\cdot10^{-4}$	$1.89\cdot10^{-10}$
s1rmq4m1	5489	5347	47.8	$3.621038\cdot10^{-7}$	$6.09\cdot10^{-5}$
s1rmt3m1	5489	5486	39.7	$1.810878\cdot10^{-7}$	$1.37\cdot10^{-4}$

matrix	n	b	av	$\lambda_{\min}(A)$	acc
s2rmq4m1	5489	5396	48.0	$2.956101\cdot10^{-9}$	$2.51\cdot10^{-3}$
s2rmt3m1	5489	5346	39.7	$2.956061\cdot10^{-9}$	$1.68\cdot10^{-3}$
s3dkq4m2	90449	90160	49.0	$-1.701524\cdot10^{-10}$	1.59
s3dkt3m2	90449	89808	40.8	$-1.998232\cdot10^{-10}$	1.34
s3rmq4m1	5489	5480	47.9	$4.855818\cdot10^{-11}$	$1.43\cdot10^{-1}$
s3rmt3m1	5489	5487	39.7	$2.393005\cdot10^{-11}$	$2.43\cdot10^{-1}$
s3rmt3m3	5357	5325	38.7	$2.457253\cdot10^{-11}$	$1.58\cdot10^{-1}$
e40r0000	17281	17231	32.0	$-9.763761\cdot10^{-6}$	$2.90\cdot10^{-9}$
fidapm11	22294	22057	27.7	$-1.374397\cdot10^{-1}$	$7.10\cdot10^{-9}$
af23560	23560	23275	19.6	$-3.041469\cdot10^{-1}$	$5.39\cdot10^{-9}$

TABLE 4.3. *Accuracy of determination of* $\lambda_{\min}(A)$

Table 4.3 shows that the smallest eigenvalue of all test matrices except the two large ones "s3dkq4m2" and "s3dkt3m2" was calculated to at least 1 decimal place, with a median of more than 7 decimal digits accuracy.

Finally we display the computing time of Algorithm 1 for the test matrices of dimension $n \geq 5000$. As before, all computations are on the Pentium M, 1.2 GHz Laptop. We display the computing time in seconds for Algorithm 1 for the matrix $A + \|A\|_1 \cdot I$ to make sure that Cholesky decomposition does not end prematurely. In the last column we display the ratio of the computing time for the whole verification and for one simple Cholesky decomposition. Given that the Cholesky decomposition is performed anyway, the overhead for verification is less than 1% in all cases.

matrix	n	b	av	time(isspd)	time(isspd)/time(chol)
bcsstk17	10974	521	39.1	1.09	1.0024
bcsstk18	11948	1243	12.5	0.77	1.0016
bcsstk25	15439	292	16.3	1.62	1.0014
bcsstk29	13992	1157	44.3	2.15	1.0016
bcsstk30	28924	16947	70.7	5.14	1.0017
bcsstk31	35588	1668	33.2	11.30	1.0007
bcsstk32	44609	43030	45.2	6.71	1.0017
s1rmq4m1	5489	191	47.8	0.99	1.0017
s1rmt3m1	5489	191	39.7	0.51	1.0024
s2rmq4m1	5489	191	48.0	1.02	1.0023
s2rmt3m1	5489	191	39.7	0.50	1.0025
s3dkq4m2	90449	614	49.0	101.09	1.0084
s3dkt3m2	90449	614	40.8	58.88	1.0005
s3rmq4m1	5489	191	47.9	1.08	1.0016
s3rmt3m1	5489	191	39.7	0.49	1.0027
s3rmt3m3	5357	5302	38.7	0.38	1.0030
e40r0000	17281	451	32.3	0.63	1.0044
fidapm11	22294	6505	28.0	22.32	1.0004
af23560	23560	304	19.6	25.33	1.0003

TABLE 4.4. *Computing time in seconds*

Note that computing times suffer from interpretation overhead. For the big matrices "s3dkq4m" and "s3dkt3m2" we see mainly time for swapping. Because of their size a symbolic factorization by the Matlab routine `symbfact` was not possible so that we could not compute $\Delta(A)$.

5. Summary. We presented an algorithm for the verification of positive definiteness of a symmetric or Hermitian matrix. The verification is rigorous, including all possible effects of rounding errors or underflow. It also allows verification of positive definiteness of all symmetric (Hermitian) matrices within an interval matrix, and the existence of negative eigenvalues can be verified as well. The verification needs only one floating-point Cholesky decomposition, so time and memory requirements are reasonable as long as the decomposition is efficient.

Acknowledgement. The author wishes to thank Per Christian Hansen for very helpful and constructive comments.

REFERENCES

[1] R.P. Brent, C. Percival, and P. Ziemmermann. Error Bounds on complex floating-point multiplication. *Math. Comp.*, 2006.

[2] L. Collatz. Einschließungssatz für die charakteristischen Zahlen von Matrizen. *Math. Z.*, 48:221–226, 1942.

[3] J.B. Demmel. On floating point errors in Cholesky. LAPACK Working Note 14 CS-89-87, Department of Computer Science, University of Tennessee, Knoxville, TN, USA, 1989.

[4] I.S. Duff, R.G. Grimes, and J.G. Lewis. User's guide for Harwell- Boeing sparse matrix test problems collection. Technical Report RAL-92-086, Computing and Information Systems Department, Rutherford Appleton Laboratory, Didcot, UK, 1992.

[5] C.A. Floudas. *Deterministic Global Optimization - Theory, Methods and Applications*, volume 37 of *Nonconvex Optimization and Its Applications*. Kluwer Academic Publishers, Dordrecht, Boston, London, 2000.

[6] N.J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM Publications, Philadelphia, 2nd edition, 2002.

[7] R.A. Horn and Ch. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[8] MATLAB User's Guide, Version 7. The MathWorks Inc., 2004.

[9] J. Rohn. Checking Robust Stability of Symmetric Interval Matrices Is NP-Hard. *Commentat. Math. Univ. Carol. 35*, pages 795–797, 1994.

[10] 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, Amsterdam, 1994.

[11] S.M. Rump. INTLAB - INTerval LABoratory. In Tibor Csendes, editor, *Developments in Reliable Computing*, pages 77–104. Kluwer Academic Publishers, Dordrecht, 1999.

[12] H. Rutishauser. *Vorlesungen über numerische Mathematik. Band 1: Gleichungssysteme, Interpolation und Approximation. Band 2: Differentialgleichungen und Eigenwertprobleme.*, volume 50/57 of *Mathematische Reihe*. Birkhäuser Verlag, Basel - Stuttgart, 1976. English: *Lectures on Numerical Analysis*, Birkhäuser, 1990.

[13] R.S. Varga. *Matrix Iterative Analysis*. Prentice-Hall, Englewood Cliffs, N.J., 1962.