Interval Computations
No 1(3), 1992

INCLUSION OF THE SOLUTION FOR LARGE LINEAR SYSTEMS WITH M-MATRIX

Siegfried M. Rump

In this paper algorithms are described for computing guaranteed error bounds for the solution of a linear system with M-matrix. The matrix as well as the right hand side may be afflicted with tolerances in which case bounds for the set of all solutions for input data within the tolerances are computed. Given a (floating-point) LDM^T-, LDL^T- or Cholesky-decomposition, resp., the additional computational effort to obtain the bounds is $O(n \cdot p)$ for dimension n and bandwidth p. The method works very well for ill-conditioned and for large linear systems with M-matrix; compared to IGA (Interval Gaussian Elimination) we gain a factor 4 in speed. The algorithms were tested up to 1 000 000 unknowns.

ЛОКАЛИЗАЦИЯ РЕШЕНИЯ ДЛЯ БОЛЬШИХ ЛИНЕЙНЫХ СИСТЕМ С M-МАТРИЦАМИ

Зигфрид М. Румп

В статье описаны алгоритмы для вычисления гарантированных границ ошибок решений линейных систем с M-матрицами. Матрица и правая часть системы могут быть заданы с допусками; в этом случае границы для множества всех решений вычисляются для входных данных с заданными допусками. Если заданы разложения (с плавающей точкой) LDM^T, LDL^T или разложение Холецкого, то для нахождения границ требуются дополнительные вычисления объемом $O(n \cdot p)$ где n – размерность, а p – ширина границ. Метод очень хорошо работает для плохо обусловленных и больших линейных систем с M-матрицей. По сравнению с IGA (интервальный метод исключения Гаусса) получено ускорение в 4 раза. Алгоритм был проверен на системах содержащих до 1 000 000 неизвестных.

1. Introduction

Let $\mathbb{F} \subseteq \mathbb{R}$ be a set of floating-point numbers. Operations between floating-point numbers are denoted by $fl(a*b)$ for $a, b \in \mathbb{F}$, $* \in \{+, -, \cdot, \backslash\}$. For a machine unit $\epsilon \in \mathbb{R}$ and $E = [-\epsilon, \epsilon] := \{ x \in \mathbb{R} \mid -\epsilon \leq x \leq \epsilon \}$ our general assumptions are

$$\begin{aligned} &fl(a*b) \in a*b \cdot (1+E) \text{ and } fl(a*b) \in a*b/(1+E), \\ &a*b \in fl(a*b) \cdot (1+E) \text{ and } a*b \in fl(a*b)/(1+E) \end{aligned} \tag{1.1}$$

where $* \in \{+, -, \cdot, /\}$. This will be true as long as no overflow and underflow occurs during computation. Furthermore, we define by induction

$$\sum_{i=1}^{k} {}_{fl} c_i := fl\left(\sum_{i=1}^{k-1} {}_{fl} c_i + c_k \right) \text{ for } c_i \in \mathbb{F}.$$

$\sum_{fl}$ denotes the floating-point sum. Finally, the basic floating-point operations $* \in \{+, -, \cdot, /\}$ are assumed to be sign-preserving, i.e. $a*b \geq 0 \Rightarrow fl(a*b) \geq 0$ as well as $a*b \leq 0 \Rightarrow fl(a*b) \leq 0$. We want to stress that these assumptions are satisfied on most computers, especially on all satisfying the IEEE 754 or 854 floating-point standard.

The set of vectors, matrices over $\mathbb{R}, \mathbb{F}$ is denoted by $\mathbb{R}^n, \mathbb{R}^{n\times n}, \mathbb{F}^n, \mathbb{F}^{n\times n}$, respectively. In this paper only vectors, matrices with n, n^2 elements occur, respectively. For matrices we use the componentwise order relations and componentwise absolute value $|A| \in \mathbb{R}^{n\times n}$ with $(|A|)_{ij} = |A_{ij}|$.

2. Some estimations

In the following we analyze the LDM^T-algorithm executed in floating-point arithmetic, namely we estimate the residual $\widetilde{L}\widetilde{D}\widetilde{M}^T - A$ for the computed floating-point matrices $\widetilde{L}$, $\widetilde{D}$ and $\widetilde{M}$ of an M-matrix A.

For a real matrix $A \in \mathbb{R}^{n\times n}$ the real LDM^T-decomposition is given for example by the following algorithm.

for $k = 1 \ldots n$ **do**

  for $j = 1 \ldots k-1$ **do** $r_{jk} := d_{jj} \cdot M_{kj}$

  $$d_{kk} := A_{kk} - \sum_{j=1}^{k-1} L_{kj} \cdot r_{jk}$$

$$\textbf{for } i = k+1 \ldots n \textbf{ do } \; L_{ik} := \{A_{ik} - \sum_{j=1}^{k-1} L_{ij} \cdot r_{jk}\}/d_{kk}$$

$$\textbf{for } j = 1 \ldots k-1 \textbf{ do } \; s_{jk} := L_{kj} \cdot d_{jj}$$

$$\textbf{for } i = k+1 \ldots n \textbf{ do } \; M_{ik} := \{A_{ki} - \sum_{j=1}^{k-1} M_{ij} \cdot s_{jk}\}/d_{kk}$$

For our purposes we formulate the algorithm for a floating-point matrix $A \in \mathbb{F}^{n \times n}$ with lower, upper bandwidth p, q, resp., i.e. $A_{ij} = 0$ for $i - j > p$ and for $j - i > q$, $p, q \geq 0$.

for $k = 1 \ldots n$ **do**

$\mu := \max(1, k-p, k-q)$

for $j = \mu \ldots k-1$ **do** $\widetilde{r}_{jk} := fl(\widetilde{d}_{jj} \cdot \widetilde{m}_{kj})$

$\widetilde{\rho}_k := \sum_{j=\mu}^{k-1}{}_{fl} fl(\widetilde{l}_{kj} \cdot \widetilde{r}_{jk})$; $\widetilde{d}_{kk} := fl(A_{kk} - \widetilde{\rho}_k)$

for $i = k+1 \ldots \min(k+p, n)$ **do**

$\nu := \max(1, k-q, i-p)$; $\widetilde{\sigma}_{ik} := \sum_{j=\nu}^{k-1}{}_{fl} fl(\widetilde{l}_{ij} \cdot \widetilde{r}_{jk})$

$\widetilde{l}_{ik} := fl(fl(A_{ik} - \widetilde{\sigma}_{ik})/\widetilde{d}_{kk})$

for $j = \mu \ldots k-1$ **do** $\widetilde{s}_{jk} := fl(\widetilde{l}_{kj} \cdot \widetilde{d}_{jj})$

for $i = k+1 \ldots \min(k+q, n)$ **do**

$\xi := \max(1, k-p, i-q)$; $\widetilde{\tau}_{ik} := \sum_{j=\xi}^{k-1}{}_{fl} fl(\widetilde{m}_{ij} \cdot \widetilde{s}_{jk})$

$\widetilde{m}_{ik} := fl(fl(A_{ki} - \widetilde{\tau}_{ik})/\widetilde{d}_{kk})$

Algorithm 1: Floating-point LDM^T without pivoting

If all operations were executed exactly and no diagonal element $\widetilde{d}_{kk}$ becomes 0 then $A = \widetilde{L}\widetilde{D}\widetilde{M}^T$. Due to rounding errors approximate equality holds and we are going to estimate the error $A - \widetilde{L}\widetilde{D}\widetilde{M}^T$. Of course, the algorithm could calculate $\widetilde{L}, \widetilde{D}, \widetilde{M}$ in the same memory as A; however, the following estimations become clearer using separate variables in each step.

We start with some simple estimations.

Lemma 1. *Let* $c_i \in \mathbb{F}, i = 1 \dots k$ *and define* $S_k := \sum_{i=1}^{k} c_i$, $G_k := \sum_{i=1}^{k} fl c_i$. *Then* $S_k \in G_k + \left(\sum_{i=2}^{k} |G_i|\right) \cdot E$. *If the* c_i *satisfy* $c_i \geq 0$ *or* $c_i \leq 0$ *for* $i = 1 \dots k$, *then* $S_k \in G_k \cdot (1 + (k-1) \cdot E)$.

Proof. By induction follows

$$S_{k+1} = S_k + c_{k+1} \in G_k + c_{k+1} + \left(\sum_{i=2}^{k} |G_i|\right) \cdot E$$
$$\in G_{k+1} \cdot (1 + E) + \left(\sum_{i=2}^{k} |G_i|\right) \cdot E.$$

For $c_i \geq 0$ we have $G_i \leq G_{i+1}$ and $0 \leq G_i \leq G_k$ for $i = 2 \dots k$ implying $\sum_{i=2}^{k} G_i \leq (k-1) \cdot G_k$. For $c_i \leq 0$ use $E = -E$. ∎

Remark. Here and in the following we use the following convention. If in an expression a set E occurs more than once then the sets E are treated independently in the sense of power set operations. For example

$$G_{k+1} \cdot (1 + E) + \left(\sum_{i=2}^{k} |G_i|\right) \cdot E$$
$$:= \left\{ G_{k+1} \cdot (1 + e_1) + \left(\sum_{i=2}^{k} |G_i|\right) \cdot e_2 \;\middle|\; e_1, e_2 \in E \right\}.$$

In this way the notation seems to be more convenient to us than absolute values.

Lemma 2. *Let* $\epsilon < 0.01$. *Then for* $c \in \mathbb{R}$ *holds*.

$$(1 + cE) \cdot (1 + E)^2 \subseteq 1 + (c + 2.01(1 + cE)) \cdot E \text{ and}$$
$$(1 + cE) \cdot (1 + E)^3 \subseteq 1 + (c + 3.04(1 + cE)) \cdot E.$$

Proof. Obvious. ∎

We define the matrix (α) with components α_{ij} by

$$\begin{aligned} &\alpha_{11} := 1;\ \alpha_{kk} := \min(k-1, p, q) && \text{for } 2 \le k \le n \\ &\alpha_{i1} := 0;\ \alpha_{ik} := \min(k-1, q, k-i+p) && \text{for } k < i \le n \\ &\alpha_{1i} := 0;\ \alpha_{ki} := \min(k-1, p, k-i+q) && \text{for } k < i \le n. \end{aligned} \tag{2.1}$$

Lemma 3. *With $\beta := \min(p, q)$ holds $\alpha_{ij} \le \beta$ for $1 \le i,\ j \le n$.*

Proof. For $2 \le k \le \beta$ is $\alpha_{kk} = k - 1 \le \beta - 1$, for $\beta < k \le n$ holds $\alpha_{kk} = \beta$. Furthermore for $k < i$ is $\alpha_{ik} \le \min(k-1, q, p-1) \le \alpha_{kk}$ and $\alpha_{ki} \le \min(k-1, p, q-1) \le \alpha_{kk}$. ∎

Let $\Delta := A - \widetilde{A}\widetilde{D}\widetilde{M}^T$. Note that the operations in the definition of Δ are real operations whereas $\widetilde{L}$, $\widetilde{D}$, $\widetilde{M}$ have been computed using floating-point arithmetic. Following algorithm 1 we first note that $\widetilde{r}_{jk}$, $\widetilde{\rho}_k$, $\widetilde{d}_{kk}$, $\widetilde{\sigma}_{ik}$, $\widetilde{l}_{ik}$, $\widetilde{s}_{jk}$, $\widetilde{\tau}_{ik}$ and $\widetilde{m}_{ik}$ is computed once and not altered during the algorithm. It is

$$\widetilde{d}_{jj} \cdot \widetilde{m}_{kj} \in \widetilde{r}_{jk} \cdot (1+E) \quad \text{and} \quad \widetilde{l}_{kj} \cdot \widetilde{d}_{jj} \in \widetilde{s}_{jk} \cdot (1+E)\ .$$

Now we assume A to be an M-Matrix (see [Neu90]) and $\widetilde{d}_{kk} \ge 0$ for $k = 1, \ldots, n$ (which is true for exact computation). Checking algorithm 1 then implies that all $\widetilde{m}_{kj}$, $\widetilde{r}_{jk}$, $\widetilde{s}_{jk}$ are not positive.

Therefore using

$$\Delta_{kk} = A_{kk} - \sum_{j=\mu}^{k-1} \widetilde{l}_{kj} \cdot \widetilde{d}_{jj} \cdot \widetilde{m}_{kj} - \widetilde{d}_{kk}$$

for $1 \le k \le n$ and using $\mu = \max(1, k-p, k-q)$ as defined in algorithm

1 we obtain

$$\begin{aligned}\Delta_{kk} \in & A_{kk} - \sum_{j=\mu}^{k-1} \tilde{l}_{kj} \cdot \tilde{r}_{jk}(1+E) - \tilde{d}_{kk} \\ \subseteq & A_{kk} - \sum_{j=\mu}^{k-1} fl(\tilde{l}_{kj} \cdot \tilde{r}_{jk}) \cdot (1+E)^2 - \tilde{d}_{kk} \\ \subseteq & A_{kk} - \tilde{\rho}_k \cdot (1 + (\alpha_{kk}-1) \cdot E) \cdot (1+E)^2 - \tilde{d}_{kk}\end{aligned}$$

using Lemma 1 and $k-1-\mu = \min(k-1, p, q) - 1$. Therefore

$$\begin{aligned}\Delta_{kk} \in & A_{kk} - \tilde{\rho}_k - \tilde{d}_{kk} + (\alpha_{kk}-1)(1+E)^2 \cdot E \cdot \tilde{\rho}_k \\ \subseteq & \tilde{d}_{kk} \cdot (1+E) - \tilde{d}_{kk} + (\alpha_{kk}-1)(1+E)^2 \cdot E \cdot \tilde{\rho}_k \\ = & \{\tilde{d}_{kk} + (\alpha_{kk}-1)(1+E)^2 \cdot \tilde{\rho}_k\} \cdot E \ .\end{aligned} \tag{2.2}$$

Hence $\tilde{d}_{kk} \leq A_{kk}$ and $\tilde{\rho}_k \leq A_{kk}$ implying

$$\Delta_{kk} \in \{1+(\alpha_{kk}-1)(1+E)^2\} \cdot A_{kk} \cdot E \subseteq \{1+1.03(\alpha_{kk}-1)\} \cdot A_{kk} \cdot E \tag{2.3}$$

provided $\epsilon \leq 0.01$. Using lemma 3 this implies

$$\begin{aligned}&\Delta_{kk} \in 1.03 \cdot \beta \cdot A_{kk} \cdot E && \text{for } 3 \leq k \leq n \text{ and} \\ &\Delta_{kk} \in A_{kk} \cdot E && \text{for } k = 2 \text{ and} \\ &\Delta_{kk} = 0 && \text{for } k = 1 \ .\end{aligned} \tag{2.4}$$

For $i > k$ we have

$$\begin{aligned}\Delta_{ik} = & A_{ik} - \sum_{j=\nu}^{k-1} \tilde{l}_{ij} \cdot \tilde{d}_{ij} \cdot \tilde{m}_{kj} - \tilde{l}_{ik} \cdot \tilde{d}_{kk} \\ \in & A_{ik} - \sum_{j=\nu}^{k-1} \tilde{l}_{ij} \cdot \tilde{r}_{jk} \cdot (1+E) - \tilde{l}_{ik} \cdot \tilde{d}_{kk} \\ \subseteq & A_{ik} - \sum_{j=\nu}^{k-1} fl(\tilde{l}_{ij} \cdot \tilde{r}_{jk})(1+E)^2 - \tilde{l}_{ik} \cdot \tilde{d}_{kk}.\end{aligned}$$

Using $A_{ik} < 0$ because A is an M-matrix yields together with lemma 1 and $k-1-\nu+1 = \min(k-1, q, k-i+p) = \alpha_{ik}$

$$\begin{aligned}\Delta_{ik} &\in fl(A_{ik} - \widetilde{\sigma}_{ik}) \cdot (1+\alpha_{ik}E)(1+E)^2 - \widetilde{l}_{ik}\widetilde{d}_{kk} \\ &\subseteq \widetilde{d}_{kk} \cdot \widetilde{l}_{ik} \cdot (1+\alpha_{ik}E)(1+E)^3 - \widetilde{l}_{ik}\widetilde{d}_{kk} \\ &\subseteq \widetilde{l}_{ik} \cdot \widetilde{d}_{kk} \cdot \{\alpha_{ik} + 3.04(1+\alpha_{ik}E)\} \cdot E.\end{aligned} \tag{2.5}$$

Using lemma 3 this implies

$$\Delta_{ik} \in \widetilde{l}_{ik} \cdot \widetilde{d}_{kk} \cdot 3.08\beta \cdot E. \tag{2.6}$$

We can conclude in a very similar way for $i < k$

$$\Delta_{ki} \in \widetilde{d}_{kk} \cdot \widetilde{m}_{ik} \cdot \{\alpha_{ki} + 3.04(1+\alpha_{ki}E)\} \cdot E \text{ and} \tag{2.7}$$

$$\Delta_{ki} \in \widetilde{d}_{kk} \cdot \widetilde{m}_{ik} \cdot 3.08\beta \cdot E. \tag{2.8}$$

We can use these estimations to derive rigorous bounds for $\Delta = A - \widetilde{L}\widetilde{D}\widetilde{M}^T$.

Theorem 4. *Let $A \in \mathbb{F}^{n\times n}$ be an M-matrix of lower, upper bandwidth p,q, resp., $\beta = \min(p,q)$ and $\widetilde{L}$, $\widetilde{D}$, $\widetilde{M}$ the computed matrices using algorithm 1, assume $\widetilde{D} \geq 0$, and $\epsilon \leq 0.01$. Define $B \in \mathbb{R}^{n\times n}$ by*

$$\begin{aligned}B_{kk} &:= 1.03 \cdot \beta \cdot A_{kk} \\ B_{ik} &:= 3.08 \cdot \beta \cdot \widetilde{l}_{ik} \cdot \widetilde{d}_{kk} \quad \text{for} \quad i > k \\ B_{ki} &:= 3.08 \cdot \beta \cdot \widetilde{d}_{kk} \cdot \widetilde{m}_{ki} \text{ for} \quad i < k.\end{aligned} \tag{2.9}$$

Then $|\Delta| = |A - \widetilde{L}\widetilde{D}\widetilde{M}^T| \leq |B| \cdot \epsilon$.

Using estimations (2.3), (2.5) and (2.7) yields better values for B. Note that rigorous bounds for B can be computed in floating-point either by using upward rounding directed or by multiplying floating-point results by $(1+\epsilon)^2$.

Computing B is very cheap, it needs only $n \cdot (p+q+1)$ additional operations. For large matrices the storage requirement is crucial. In a practical implementation of algorithm 1 the LDM^T decomposition would

overwrite the matrix A. Therefore B needs one additional vector of length n to store the diagonal of A.

If the matrix A is symmetric the LDL^T-algorithm can be applied. This is the very same as algorithm 1 except that in line 3 $\widetilde{m}_{kj}$ is to be replaced by $\widetilde{l}_{kj}$ and the computation of M (lines 8 ... 11) can be omitted. In exact computation it follows $A = LDL^T$. The analysis of the LDL^T-algorithm is the same.

Theorem 5. *Let $A \in \mathbb{F}^{n\times n}$ be a symmetric M-matrix of bandwidth p and $\widetilde{L}$, $\widetilde{D}$ the computed matrices using the floating-point LDL^T-algorithm, assume $\widetilde{D} \geq 0$, and $\epsilon \leq 0.01$. Define $B \in \mathbb{R}^{n\times n}$ by*

$$B_{kk} := 1.03 \cdot p \cdot A_{kk}$$
$$B_{ik} := 3.08 \cdot p \cdot \widetilde{l}_{ik} \cdot \widetilde{d}_{kk} \quad \text{for } i \neq k$$

Then $|\Delta| = |A - \widetilde{L}\widetilde{D}\widetilde{L}^T| \leq |B| \cdot \epsilon$.

3. Cholesky decomposition

Let $A \in \mathbb{F}^{n\times n}$ be a symmetric and positive definite matrix of bandwidth p. Then as in the case of M-matrices error bounds can be computed based on the floating-point Cholesky decomposition. The sums occuring in the algorithm do not necessarily consist of summands of equal sign and the error estimates using lemma 1 would become poor. Therefore, we use a scalar product with one final rounding as proposed by Kulisch [Ku76]. This means that for a_i, $b_i \in \mathbb{F}$, $1 \leq i \leq n$

$$fl\left(\sum_{i=1}^{n} a_i \cdot b_i\right) \in \left(\sum_{i=1}^{n} a_i \cdot b_i\right) \cdot (1 + E)$$

where $E = [-\epsilon, +\epsilon]$.

for $k = 1 \ldots n$ do

$$\mu = \max(1, k - p)$$

$$\widetilde{S}_{kk} = fl\left(A_{kk} - \sum_{j=\mu}^{k-1} \widetilde{g}_{kj}^2\right); \quad \widetilde{g}_{kk} = \sqrt{fl\,\widetilde{S}_{kk}}$$

for $i = k+1 \ldots \min(n, k+p)$ do

$$\nu = \max(1, i-p);\ \xi = \min(n, k-1)$$

$$\widetilde{g}_{ik} = fl\left(fl\left(A_{ik} - \sum_{j=\nu}^{\xi} \widetilde{g}_{ij} \cdot \widetilde{g}_{kj} \right) \Big/ \widetilde{g}_{kk} \right)$$

Algorithm 2. Floating-point Cholesky decomposition using the precise scalar product

Executing algorithm 2 without rounding errors yields $A = G \cdot G^T$. Next we are going to estimate $\Delta = A - \widetilde{G} \cdot \widetilde{G}^T$. The analysis is similar to the one described in [Kie87] but adapted to our purposes. We assume $\widetilde{S}_{kk} > 0$ for $1 \le k \le n$. First

$$\sqrt{\widetilde{S}_{kk}} \in \widetilde{g}_{kk} \cdot (1+E) \text{ implying } \widetilde{S}_{kk} \in \widetilde{g}_{kk}^2 \cdot (1+E)^2.$$

Hence

$$\begin{aligned} \Delta_{kk} =& A_{kk} - \sum_{j=\mu}^{k-1} \widetilde{g}_{kj}^2 - \widetilde{g}_{kk}^2 \in \widetilde{S}_{kk} \cdot (1+E) - \widetilde{g}_{kk}^2 \\ \subseteq& \widetilde{g}_{kk}^2 \cdot (3 + 3E + E^2) \cdot E \ . \end{aligned} \tag{3.1}$$

Again we want to stress that different E in (3.1) are treated independently. That means $(3+3E+E^2)\cdot E = \{(3+3e_1+e_2\cdot e_3)\cdot e_4 \mid e_i \in E \text{ for } 1 \le i \le 4\}$.

Furthermore

$$\begin{aligned} \Delta_{ik} =& A_{ik} - \sum_{j=\nu}^{\xi} \widetilde{g}_{ij} \cdot \widetilde{g}_{kj} - \widetilde{g}_{ik} \cdot \widetilde{g}_{kk} \\ \in& fl\left(A_{ik} - \sum_{j=\nu}^{\xi} \widetilde{g}_{ij} \cdot \widetilde{g}_{kj} \right) \cdot (1+E) - \widetilde{g}_{ik} \cdot \widetilde{g}_{kk} \\ =& \left\{ fl\left(A_{ik} - \sum_{j=\nu}^{\xi} \widetilde{g}_{ij} \cdot \widetilde{g}_{kj} \right) \cdot (1+E)/\widetilde{g}_{kk} - \widetilde{g}_{ik} \right\} \cdot \widetilde{g}_{kk} \\ \subseteq& \{\widetilde{g}_{ik} \cdot (1+E)^2 - \widetilde{g}_{ik}\} \cdot \widetilde{g}_{kk} = \widetilde{g}_{ik} \cdot \widetilde{g}_{kk} \cdot (2+E) \cdot E \ . \end{aligned} \tag{3.2}$$

Lemma 6. *Let $A \in \mathbb{F}^{n\times n}$ be symmetric with bandwidth p and assume $\widetilde{S}_{kk} > 0$ during execution of algorithm 2. Then for $\epsilon \leq 0.01$ and using $B \in \mathbb{R}^{n\times n}$ with*

$$\begin{aligned} B_{kk} &:= 3.04 \cdot \tilde{g}_{kk}^2 \qquad 1 \leq k \leq n \\ B_{ik} &:= 2.01 \cdot \widetilde{g}_{ik} \cdot \widetilde{g}_{kk} \; i \neq k \end{aligned} \tag{3.3}$$

the computed matrix $\widetilde{G}$ satisfies

$$|\Delta| = |A - \widetilde{G}\widetilde{G}^T| \leq \epsilon \cdot |B| \; .$$

Using

$$C := 3.04 \cdot |\widetilde{G}| \cdot \mathrm{diag}(\widetilde{G}) \tag{3.4}$$

we have

$$|\Delta| \leq \epsilon \cdot C \; .$$

The estimation matrix B and slightly weaker C bear the advantage that all components compute directly from $\widetilde{G}$; no additional memory for the diagonal of the matrix A is necessary. Furthermore, the bounds are very simple and sharp.

4. Verified bounds

Let $A \in \mathbb{R}^{n\times n}$ be an M-matrix, $b \in \mathbb{R}^n$ and $\widetilde{x} \in \mathbb{R}^n$ be given. There is no prerequisite on the accuracy of $\widetilde{x}$. If $\widetilde{x}$ is not too bad the residual $A\widetilde{x} - b$ is small.

In practical applications frequently uncertain data occur. Those can frequently be packed into the right hand side. We therefore assume bounds $\underline{b}, \overline{b}$ be given for the right hand side b. In interval notation it is $[b] = [\underline{b}, \overline{b}] := \{ b \in \mathbb{R}^n \mid \underline{b} \leq b \leq \overline{b} \}$ with componentwise $\leq$. We ask for bounds for the solution complex

$$\sum(A, [b]) := \{ x \mid Ax = b, b \in [b] \} \; .$$

For practical applications it turns out that it is superior to ask for bounds for $\sum(A, [b] - A\widetilde{x})$ and use $\sum(A, [b]) = \widetilde{x} + \sum(A, [b] - A\widetilde{x})$. Since in practical applications $[b] - A\widetilde{x}$ is almost symmetric to the origin we use $|[b] - A\widetilde{x}| := y$ with $y_i = \max |[b]_i - (A\widetilde{x})_i|$.

Definition 7. For a nonsingular lower triangular matrix $L \in \mathbb{R}^{n\times n}$ and right hand side $b \in \mathbb{R}^n$, $b \geq 0$ we define $y := L \setminus b$ by

$$y_i := (b_i + \sum_{j=1}^{i-1} |L_{ij}| \cdot b_j) \ / \ |L_{ii}| \ . \tag{4.1}$$

Furthermore $y := L \nwarrow b$ is defined by (4.1) using upward directed rounding.

This form of backward substitution satisfies $L \backslash b = \langle L \rangle^{-1} \cdot b$ where $\langle L \rangle$ is the comparison matrix (see [Neu90]). $L \nwarrow b$ yields the upper bound of the result of performing interval backward substitution for L and b. Using $b \geq 0$ implies $L^{-1} \cdot b \in \pm L \backslash b \subseteq \pm L \nwarrow b$. For upper triangular and diagonal matrices $\backslash$ and $\nwarrow$ are defined similarly.

Definition 7 implies

$$M^{-T} \cdot D^{-1} \cdot L^{-1} \cdot ([b] - A\tilde{x}) \subseteq \pm z \quad \text{for}$$

$$z := M^T \backslash (D \backslash (L \backslash |[b] - A\tilde{x}|)) = \langle M^T \rangle^{-1} \cdot D^{-1} \cdot \langle L \rangle^{-1} \cdot |[b] - A\tilde{x}| \ .$$

Define

$$y := M^T \backslash (D \backslash (L \backslash (|\Delta| \cdot z))) = \langle M^T \rangle^{-1} \cdot D^{-1} \cdot \langle L \rangle^{-1} \cdot |\Delta| \cdot z \ .$$

Then $y \in \mathbb{R}^n, y \geq 0$. Assume $z > y$ and let $\delta \in \mathbb{R}$ be given such that

$$\delta > y_i/(z_i - y_i) \quad \text{for } 1 \leq i \leq n \ .$$

Using $R := \langle M^T \rangle^{-1} \cdot D^{-1} \cdot \langle L \rangle^{-1}$ this implies

$$y + \delta y = R \cdot |\Delta| \cdot (z + \delta z) < \delta \cdot z \ .$$

The quantities z, y, R and δ are nonnegative. Hence for all $C \in \mathbb{R}^{n \times n}, x \in \mathbb{R}^n$ with $0 \leq |C| \leq R \cdot |\Delta|$ and $0 \leq |x| \leq z + \delta z$ holds

$$-\delta \cdot z < C \cdot x < \delta \cdot z \ .$$

Therefore for all $\tilde{z} \in \mathbb{R}^n, 0 \leq |\tilde{z}| \leq z$ and $X := \tilde{z} \pm \delta \cdot z \in \mathbb{PR}^n$

$$C \cdot X \subseteq \text{int}(X - \tilde{z}) \quad \text{or} \quad \tilde{z} + C \cdot X \subseteq \text{int}(X) \ . \tag{4.2}$$

The operations in (4.2) are the power set operations. The assumptions are valid for $C := M^{-T} \cdot D^{-1} \cdot L^{-1} \cdot (LDM^T - A)$ and every $\tilde{z} = M^{-T} D^{-1} L^{-1} \cdot (b - A\tilde{x})$, $b \in [b]$. X is nonempty, compact and $f(x) := \tilde{z} + C \cdot x : \mathbb{R}^n \to \mathbb{R}^n$ is continuous such that (4.2) and Brouwers Fixed Point Theorem implies the existence of a fixed point $\hat{x}$ of f within X. Using theorem 11 in [Ru86] implies the nonsingularity of A and

$$\widehat{x} = (I - C)^{-1} \cdot \widetilde{z} = (M^{-T} D^{-1} L^{-1} A)^{-1} \cdot \widetilde{z} = A^{-1}(b - A\widetilde{x}) \in X \ .$$

Therefore, observing $|X| \leq (1+\delta) \cdot z$ we have the following theorem.

Theorem 8. *Let $A \in \mathbb{F}^{n\times n}$ be an M-matrix of lower, upper bandwidth p,q, resp., $\beta = \min(p,q)$ and $\widetilde{L}$, $\widetilde{D}$, $\widetilde{M}$ the computed matrices using algorithm 1 with $\widetilde{d}_{kk} > 0$ and $\epsilon \leq 0.01$. Define $B \in \mathbb{R}^{n\times n}$ by (2.9) and let $[b] \in \mathbb{PR}^n$, $\widetilde{x} \in \mathbb{R}^n$, $|[b] - A\widetilde{x}| = \max\{|b - A\widetilde{x}| \mid b \in [b]\}$,*

$$z := M^T \nwarrow (D \nwarrow (L \nwarrow |[b] - A\widetilde{x}|)) \in \mathbb{R}^n \quad \textit{and}$$

$$y := M^T \nwarrow (D \nwarrow (L \nwarrow (\epsilon \cdot |B| \cdot z))) \in \mathbb{R}^n$$

using the backward substitution $\nwarrow$ as defined in Definition 7. Assume $z > y$ and let some $\delta \in \mathbb{R}$ be given with $\delta > y_i/(z_i - y_i)$ for $1 \leq i \leq n$. Then A is invertible and for every $b \in [b]$ the solution $A^{-1} \cdot b$ of the linear system $Ax = b$ satisfies

$$A^{-1} \cdot b \in \widetilde{x} \pm (1+\delta) \cdot z \ . \tag{4.3}$$

The quantities used in Theorem 8 are all rigorously computable. One backward substitution of the form of Definition 7 for a triangular matrix of bandwidth β requires less than $n \cdot \beta$ operations. Hence the total additional effort to obtain guaranteed bounds is

$2n \cdot (p+q)$	operations for computing B
$n \cdot (p+q+1)$	operations for computing $\lvert[b] - A\widetilde{x}\rvert$
$2n \cdot (\beta+1)$	operations for computing z
$2n \cdot (\beta+1)$	operations for computing y
$2n$	operations for computing δ
n	operations for computing $\widetilde{x} \pm (1+\delta) \cdot z$

counting one addition and one multiplication as one operation.

Corollary 9. *The computational effort for the bounds (4.3) in Theorem 8 is less than $n \cdot \{3(p+q) + 4\beta + 8\}$ operations.*

This compares to $n \cdot \{(p+q) + 2\beta + 3\}$ for computing $\widetilde{x} + M^T \backslash (D \backslash (L \backslash (b - A\widetilde{x})))$. When accepting slightly weaker bounds the computational effort can be further diminished in obvious ways. It should be stressed that corollary 9 estimates the total computational effort after computing $\widetilde{L}$, $\widetilde{D}$ and $\widetilde{M}$. The cost reduces for symmetric matrices in obvious ways.

Following the proof of Theorem 8 it is immediately clear that $\Delta = A - \widetilde{L}\widetilde{D}\widetilde{M}^T$ can be replaced by $\Delta = A - \widetilde{G}\widetilde{G}^T$ together with Lemma 6 for estimating Δ.

Theorem 10. *Let $A \in \mathbb{F}^{n\times n}$ be a symmetric positive definite matrix of lower bandwidth p and $\widetilde{G}$ be the computed Cholesky decomposition using Algorithm 2 with $\widetilde{g}_{kk} > 0$ and $\epsilon \le 0.01$. Define $C \in \mathbb{R}^{n\times n}$ by (3.4) and let $[b] \in \mathbb{PR}^n$, $\widetilde{x} \in \mathbb{R}^n$, $|[b] - A\widetilde{x}| = \max\{|b - A\widetilde{x}| \mid b \in [b]\}$,*

$$z = \widetilde{G}^T \nwarrow (\widetilde{G} \nwarrow (|[b] - A\widetilde{x}|)) \in \mathbb{R}^n \text{ and}$$

$$y = \widetilde{G}^T \nwarrow (\widetilde{G} \nwarrow (\epsilon \cdot C \cdot z)) \in \mathbb{R}^n$$

using the backward substitution $\nwarrow$ as defined in Definition 7. Assume $z > y$ and let some $\delta \in \mathbb{R}$ be given with $\delta > y_i/(z_i - y_i)$ for $1 \le i \le n$. Then A is invertible and for every $b \in [b]$ the solution $A^{-1}b$ of the linear system $Ax = b$ satisfies

$$A^{-1} \cdot b \in \widetilde{x} \pm (1 + \delta) \cdot z \ .$$

In a practical implementation we need memory for A, b, $\widetilde{x}$ and one additional vector z. Then the algorithm would calculate $A \setminus b =: \widetilde{x}$ using the additional z, then $z := |b - A\widetilde{x}|$, $z = A \nwarrow z$, $b = \epsilon \cdot C \cdot z$ and $b = A \nwarrow b$. Then $\delta = \max b_i/(z_i - b_i)$ and $A^{-1} \cdot b \in \widetilde{x} \pm (1 + \delta) \cdot z$.

Given $\widetilde{G}, \widetilde{x}$ the total additional computational effort for computing guaranteed bounds is less then $n \cdot (6\beta + 8)$ operations. The total effort is therefore $\frac{1}{6} \cdot n \cdot \beta^2 + 8n \cdot (\beta + 1)$.

For M-matrices the interval version of Gaussian elimination is always executable, at least when computing in $\mathbb{IR}$. The computational effort is $\frac{2}{3} \cdot n^3$ when counting one interval operation as two floating-point operations. For symmetric matrices of bandwidth β these are $\frac{2}{3} \cdot n \cdot \beta^2$ operations.

5. Numerical results

All of the following numerical results are obtained using IEEE 754 single precision which is equivalent to roughly 7 decimal digits in the mantissa.

Our proposed approach first computes a floating-point decomposition of the matrix of the linear system (LU, LDL^T or Cholesky) and then

uses an estimation of the residual $A - \widetilde{L}\widetilde{U}$, $A - \widetilde{L}\widetilde{D}\widetilde{M}^T$, $A - \widetilde{L}\widetilde{D}\widetilde{L}^T$ or $A - \widetilde{G}\widetilde{G}^T$, respectively to obtain verified bounds for the solution together with the nonsingularity of A. In the following examples A is symmetric. Extensive tests with nonsymmetric A have been performed yielding very similar results.

The results of our proposed algorithms are referred as new(Gauß) and new(Chol). They are compared with interval Gaussian elimination using LU-decomposition (referred as IGA) and with interval Gaussian elimination using LDL^T-decomposition (referred as IGAS). Counting one interval operation as two floating-point operations IGA requieres $\frac{2}{3} \cdot n \cdot \beta^2$ and IGAS $\frac{1}{3} \cdot n \cdot \beta^2$ operations for symmetric matrix of bandwidth β. This is a factor 4 of IGA against our method. Both algorithms use interval backward substitution for obtaining verified bounds for the solution.

When using interval arithmetic over real numbers IGA without pivoting will produce an inclusion of the solution for M-matrices (see [Neu90]). When executing IGA with interval arithmetic over floating point numbers the small overestimations due to directed rounding of the floating-point bounds may force a breakdown of the algorithm at a certain point, that is division by intervals containing zero occurs.

All of the right hand sides of the following linear systems are constructed such that the ith component of the true solution is $\frac{1}{i}$. Other right hand sides have been tested and showed similar results.

We always display two results for such a linear system $Ax = b$, one for point data and one for interval data with relative perturbations of 1E-5. That is we look at the solution of the interval linear system

$$\{\widehat{x} \mid \widetilde{A}\widehat{x} = \widetilde{b} \text{ for } |A - \widetilde{A}| \leq \text{1E-5} \cdot |A|,\ |b - \widetilde{b}| \leq \text{1E-5} \cdot |b|\}.$$

For point data, interval data inclusions X, Y are produced, respectively. In the following we display the minimum and maximum relative error of X and Y w.r.t. the solution of $Ax = b$, i.e.

$$\min_i \frac{\text{rad}(X_i)}{1/i} \quad \text{and} \quad \max_i \frac{\text{rad}(X_i)}{1/i}$$

and similarly for Y.

Our first example is the well-known matrix

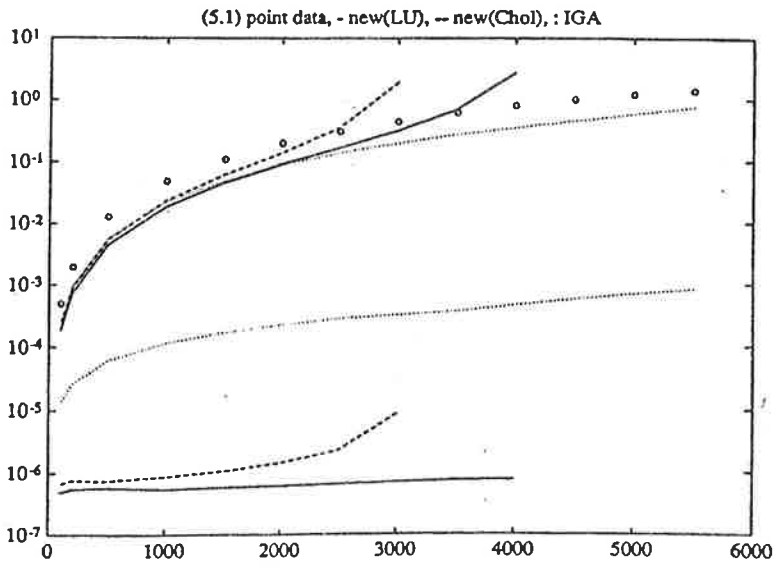
$$\begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & & & \\ & & & -1 & 2 \end{pmatrix} \tag{5.1}$$

The following graph shows the minimum and maximum relative error of the computed inclusion versus dimension for

new(Gauß)	solid line
new(Chol)	dashed line
IGA	dotted line

for point data. All graphical output is in a semilogarithmic scale. For comparison we also display

$\mathrm{cond}(A)/10^7$ circles (we are computing in single precision).



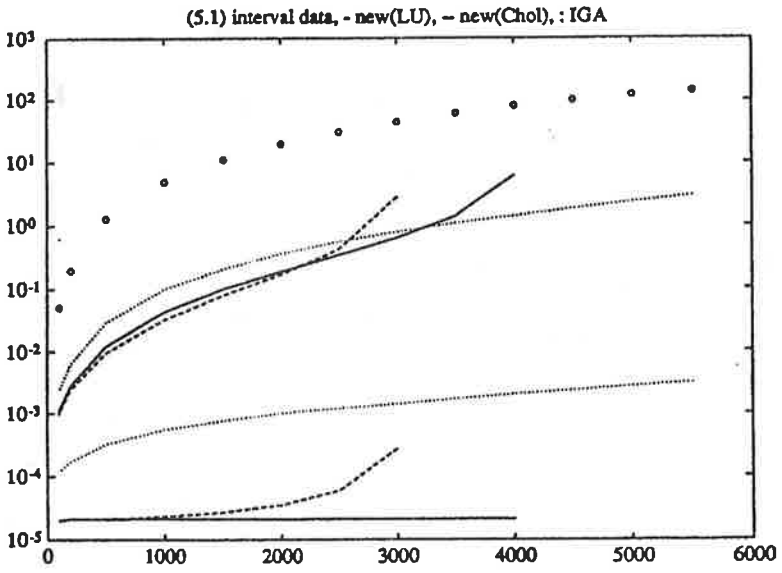
Graph 1: matrices (5.1), point data

The results show that the minimum relative error is almost constant whereas the maximum relative error grows with the condition number. This is due to the fact that the components of the solution differ by several orders of magnitude. IGA works for higher dimension better than new(Gauß). However, for dimensions beyond 4000 some components the

inclusion become wide.

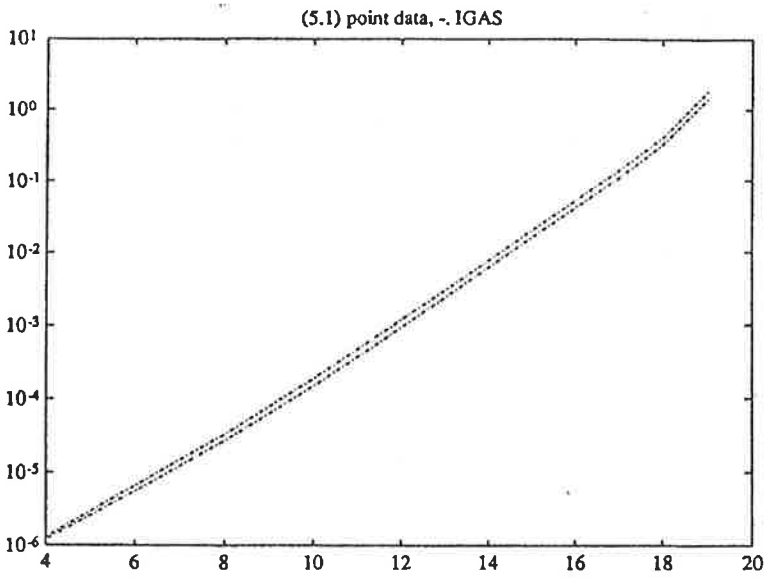
In the following graph the same situation is shown for interval data with relative perturbation 1E-5 in the matrix and right hand side. For comparison we display

$\mathrm{cond}(A)/10^7 \cdot 10^7 \cdot$1E-5 circles..



Graph 2: matrices (5.1), interval data

Following are the results for IGAS. Obviously interval dependencies are responsible for those very bad results. The algorithm fails already for 20 unknows.

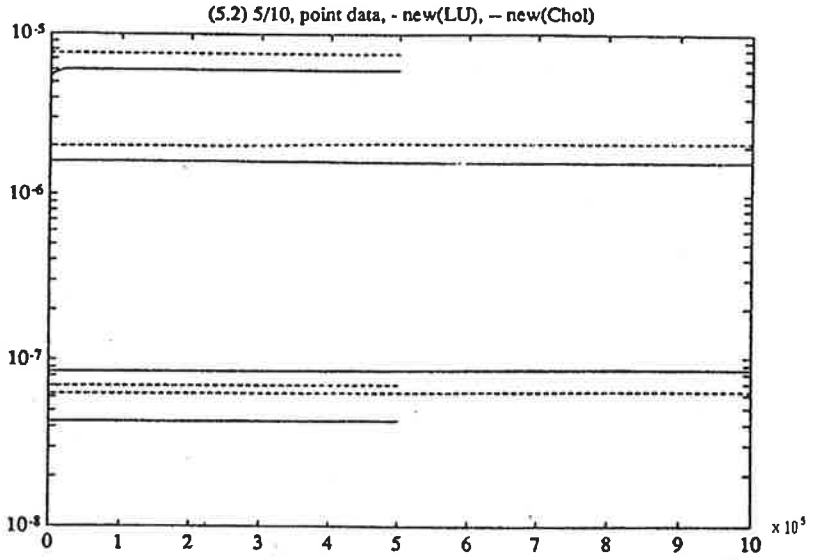


Graph 3: matrices (5.1), IGAS

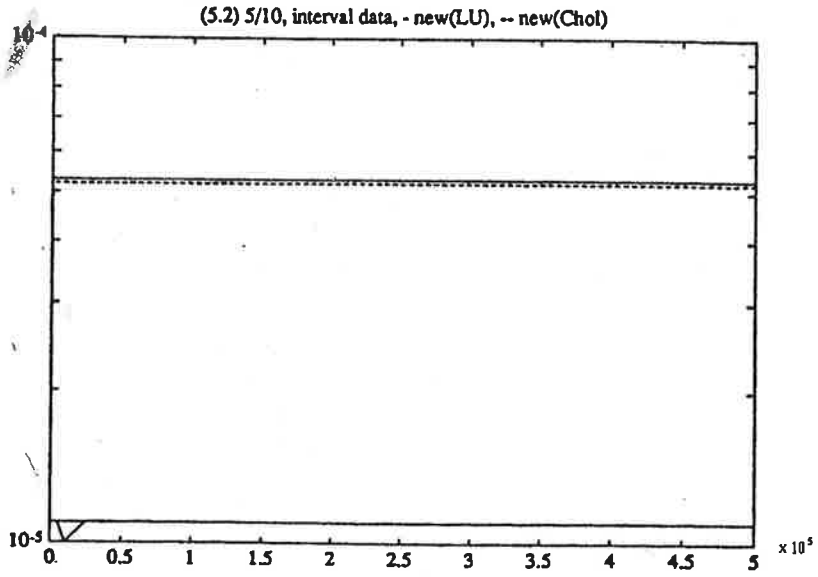
Following is the discretisation of the Poisson equation with

$$M = \begin{pmatrix} 4 & -1 & & \\ -1 & 4 & & \\ & & & \\ & & -1 & 4 \end{pmatrix} \text{ and } A = \begin{pmatrix} M & -I & & \\ -I & M & & \\ & & & \\ & & -I & M \end{pmatrix} \tag{5.2}$$

The bandwidth equals the number of rows of M, the condition number of A increases with the bandwidth. The following graph shows the minimum and maximum relative error for the inclusion computed by new(Gauß) and new(Chol) for bandwidth 5 and 10. Bandwidth 5 has been computed up to 10^6 unknowns, bandwidth 10 up to $5 \cdot 10^5$ unknowns.



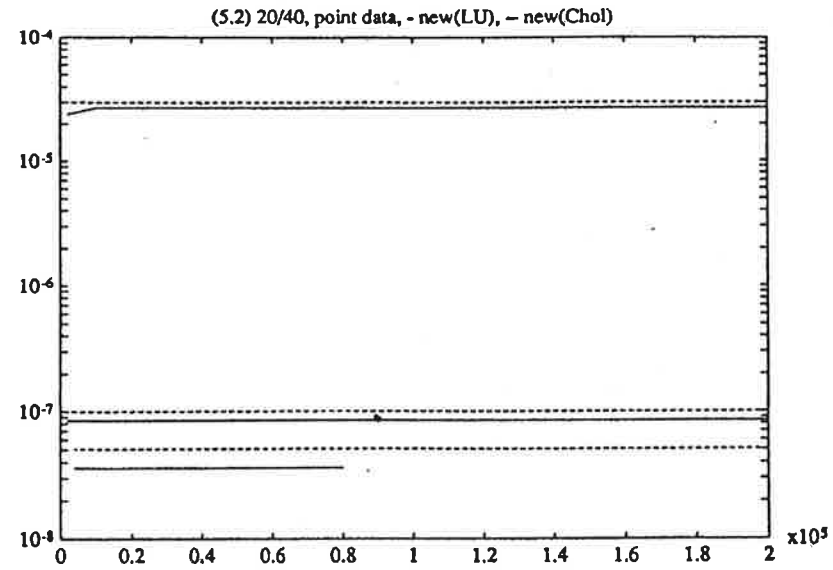
Graph 4: matrices (5.2), bandwidths 5 and 10, point data



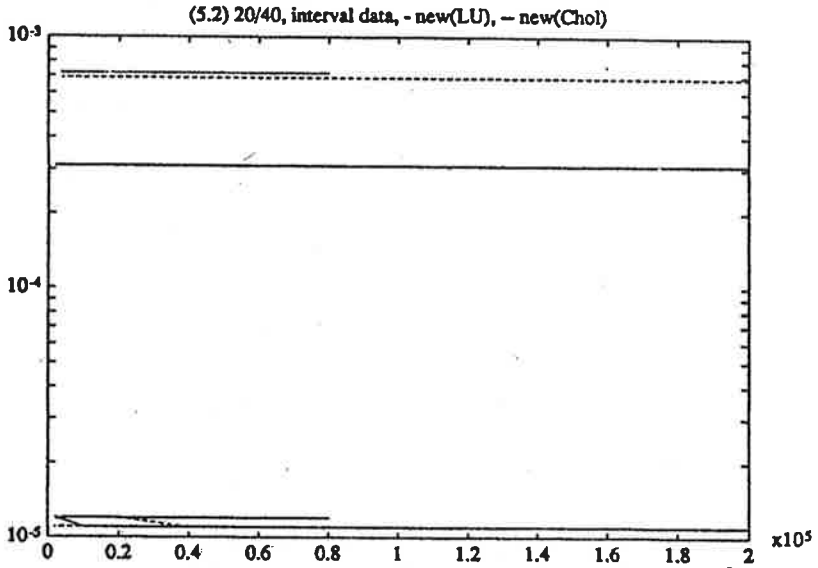
Graph 5: matrices (5.2), bandwidths 5 and 10, interval data

We see that both algorithms produce very sharp bounds independent of the number of unknowns.

For bandwidths 20 and 40 the situation is similar.



Graph 6: matrices (5.2), bandwidths 20 and 40, point data

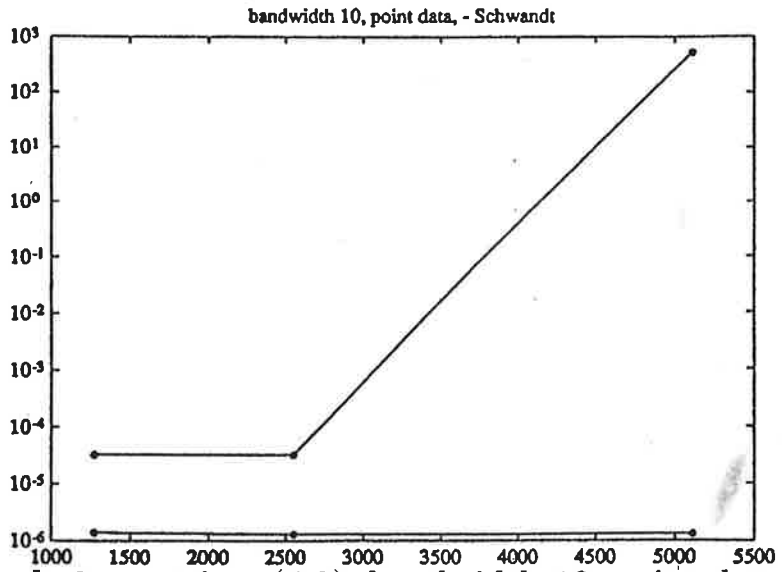


Graph 7: matrices (5.2), bandwidths 20 and 40, interval data

Matrices with bandwidth 20 have been computed up to $2 \cdot 10^5$, with bandwidth 40 up to $8 \cdot 10^4$ unknowns.

We also tested the interval version of Bunemann's algorithm as proposed by Schwandt [Sch84]. We obtained for as small examples like 5110 unknowns, bandwidth 10 inclusions of the solution with relative error far beyond 1.0, i.e. no correct figure of the result.

The following Graph shows the results for a point matrix of bandwidth 10.



Graph 8: matrices (5.2), bandwidth 10, point data

The rapid growth of the maximal relative error is due to the increasing number of recursive calls. Only in the middle (around the 2500^{th} unknown) the relative error is large.

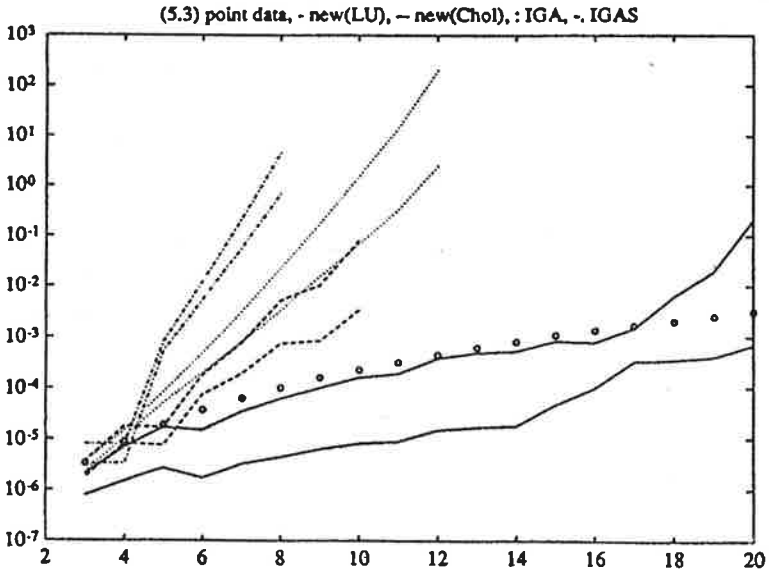
From [Gre69] we used example 4.16

$$A = \begin{pmatrix} 5 & -4 & 1 & & & & & \\ -4 & 6 & -4 & 1 & & & & \\ 1 & -4 & 6 & -4 & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & 1 & -4 & 6 & -4 & \\ & & & & 1 & -4 & 5 & \end{pmatrix} \tag{5.3}$$

with eigenvalues $\lambda_k = 16 \cdot \sin^4\left(\frac{k\pi}{2(n+1)}\right)$, $k = 1 \ldots n$.

Following we display

new(Gauß)	solid line
new(Chol)	dashed line
IGA	dotted line
IGAS	dashdot line
cond$(A)/10^7$	circles .



Graph 9: matrices (5.3), point data

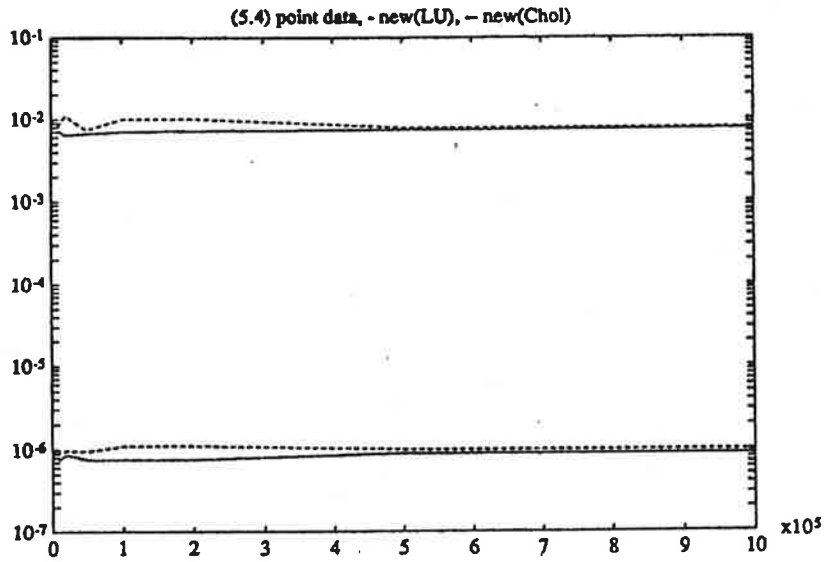
Obviously the new method with LDL^T-decomposition is superior to the others. It yields at least one correct figure for dimensions up to 20

whereas the others give up at much lower dimensions. The graph for interval data looks totally similar.

Finally dependency on the dimension for constant condition number was tested. We choose

$$A = \begin{pmatrix} a & -1 & & & \\ -1 & a & -1 & & \\ & & & & -1 \\ & & & -1 & a \end{pmatrix} \tag{5.4}$$

with value for a such that $\mathrm{cond}(A) = 10^5$.



Graph 10: matrices (5.4), point data

The new algorithms show equal performance independent of the dimension. The maximum relative error of the inclusion is equal to $\mathrm{cond}(A)/10^7$. That means the accuracy of the inclusion equals the sensitivity of the solution w.r.t. perturbations in the last bit of the input data.

The results for interval data are totally similar.

References

[AlHe83] Alefeld, G.; Herzberger, J., *Introduction to interval computations*, Academic Press, 1983.

[ArDeDu89] Arioli, M., Demmel and J.W., Duff, I.S., *Solving sparse linear systems with backward error*, SIAM J. Matrix Anal. Appl. 10 no. 2 (1989), 165–190.

[Gre69] Gregory, R.T., *A Collection of matrices for testing computational algorithms*, John Wiley & Sons, New York, 1969.

[Kie87] Kielbaszinski A., *A Note on rounding-error analysis of cholesky factorization*, LAA 88/89 (1987), 487–494.

[Ku76] Kulisch, U., *Grundlagen des numerischen Rechnens*, (Reihe Informatik, 19) Mannheim-Wien-Zürich, Bibliographisches Institut, 1976.

[Neu90] Neumaier, A., *Interval methods for systems of equations*, Encyclopedia of Mathematics and its Applications, Cambridge University Press, 1990.

[Ru86] Rump, S.M., *New results on verified inclusions*, Accurate Scientific Computations, W.L. Miranker and R. Toupin, Springer Lecture Notes in Computer Science 235, 39 Seiten, 1986.

[Ru90] Rump, S.M., *Rigorous sensitivity analysis for systems of linear and nonlinear equations*, MATH. of COMP. **54** no. 10 (1990), 721–736.

[Sch84] Schwandt, H., *An interval arithmetic approach for the construction of an almost globally convergent method for the solution of the nonlinear poisson equation on the unit square*, SIAM J. Sci. Stat. Comp. **5** no. 2 (1984).

Institut fuer Informatik III, TU Hamburg
Eissendorfer Strasse 38
D-2100, Hamburg–Harburg 90
Germany