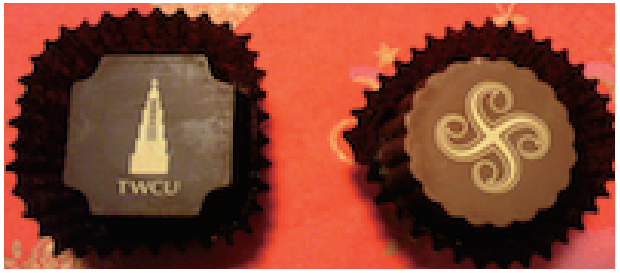


Verified solutions of sparse linear systems

Takeshi Ogita
Division of Mathematical Sciences
Tokyo Woman's Christian University
Japan

joint work with
Shin'ichi Oishi and **Siegfried M. Rump**

SCAN2012, Novosibirsk
September 26, 2012

Outline

$\mathbb{F}$: a set of fixed precision floating-point numbers, e.g., IEEE 754 binary64
Let us consider $Ax = b$ where $A \in \mathbb{F}^{n \times n}$, $b \in \mathbb{F}^n$.

Task By the use of floating-point arithmetic (with intervals), we aim to

- prove A is nonsingular, and
- compute a forward error bound of an approximate solution $\widetilde{x}$ of $Ax = b$ s.t.

$$|(A^{-1}b)_i - \widetilde{x}_i| \leq \epsilon_i \quad \text{for } 1 \leq i \leq n.$$

Brief assumptions and conditions

- The matrix A is large, sparse and moderately ill-conditioned.
- The verification process should be as fast as possible.
- Obtained error bounds should be tight (meaningful).

$$\Downarrow$$

Some information on A^{-1} is necessary. (To estimate $\|A^{-1}\|$ is essential.)

$$\Downarrow$$

One of the Grand Challenges in Interval Analysis

[1] A. Neumaier: Grand Challenges and Scientific Standards in Interval Analysis, *Reliable Computing*, **8** (2002), 313–320.

"Apart from a paper by Rump, nothing bas been done on the interval side."

Notation

- For $x = (x_1, x_2, \ldots, x_n)^T \in \mathbb{R}^n$

$$|x| = (|x_1|, |x_2|, \ldots, |x_n|)^T \in \mathbb{R}^n$$
$$\|x\|_\infty = \max_{1 \le i \le n} |x_i|$$

- For $A = (a_{ij}) \in \mathbb{R}^{m \times n}$

$$|A| = (|a_{ij}|) \in \mathbb{R}^{m \times n}$$
$$\|A\|_2 = \sqrt{\rho(A^T A)} = \sqrt{\lambda_{\max}(A^T A)}$$
$$\|A\|_\infty = \max_{1 \le i \le m} \sum_{1 \le j \le n} |a_{ij}|$$

- For $A = (a_{ij}),\ B = (b_{ij}) \in \mathbb{R}^{m \times n}$

$$A \leq B \quad \Longleftrightarrow \quad a_{ij} \leq b_{ij} \quad \text{for all } (i, j)$$

- $\mathbf{o}$: zero vector
- $\mathbf{e}$: vector of all ones
- O: matrix of all zeros
- I: identity matrix
- $\mathbf{u}$: rounding error unit (unit round-off), $\mathbf{u} \approx 10^{-16}$ in IEEE 754 binary64
- $\kappa(A) = \|A\| \cdot \|A^{-1}\|$: condition number

Difficult points for sparse matrices

For **dense** linear systems there are several efficient methods for this purpose (e.g. by Rump (1980), Oishi-Rump (2002), Hansen-Bliek-Rohn-Ning-Kearfott ([Neumaier] 1999)).

- **Common basis**: use of an approximate full inverse of either A or its LU factors.
- **Cost**: comparable with a standard numerical algorithm, Gaussian elimination with partial pivoting.
- **Applicability**: $\kappa(A) \lesssim 1/\mathbf{u} \sim 10^{16}$ in binary64.
- **Model implementation**: `verifylss` in INTLAB, a Matlab toolbox for reliable computing.

For **sparse** cases things are much different: Still difficult in terms of both computational complexity and memory requirements.

- **Difficulty**: destruction of the sparsity of A if using full inverses.
- **Exception**: diagonally dominant and M-matrix or alike.

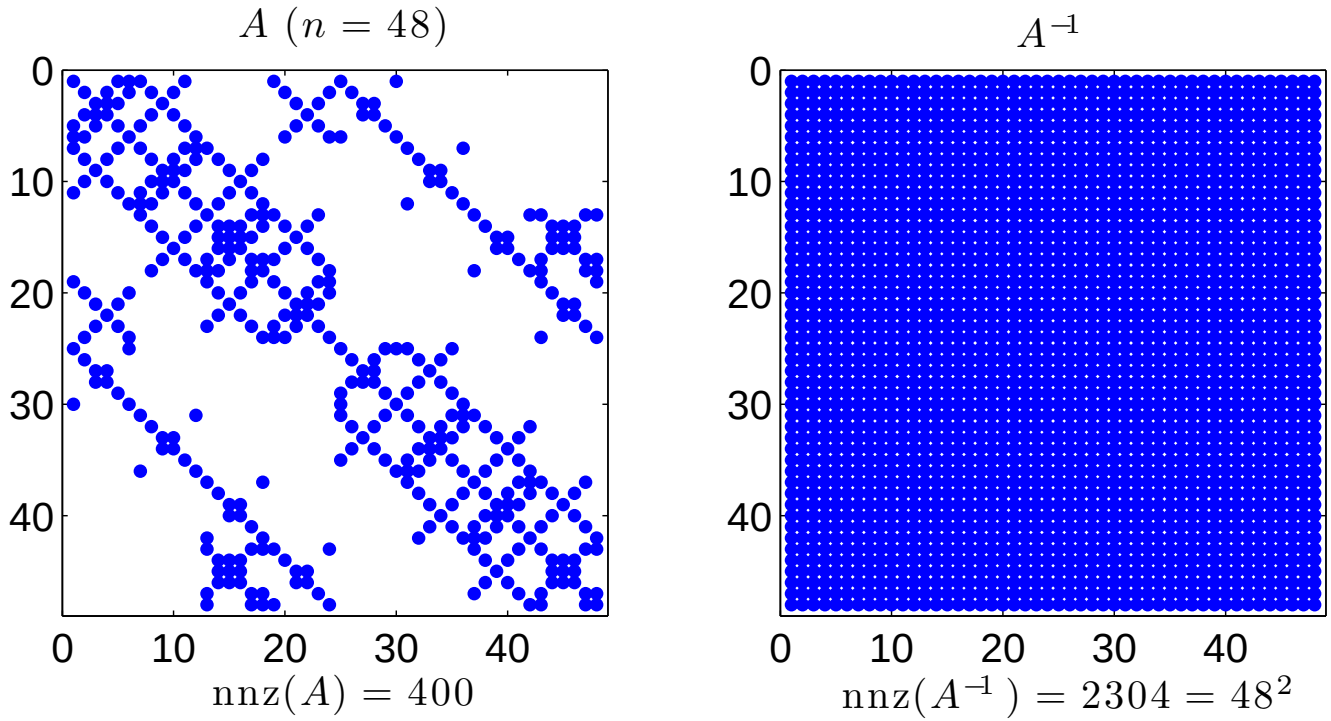


Figure 1: Destruction of sparsity of A $(n = 48)$.

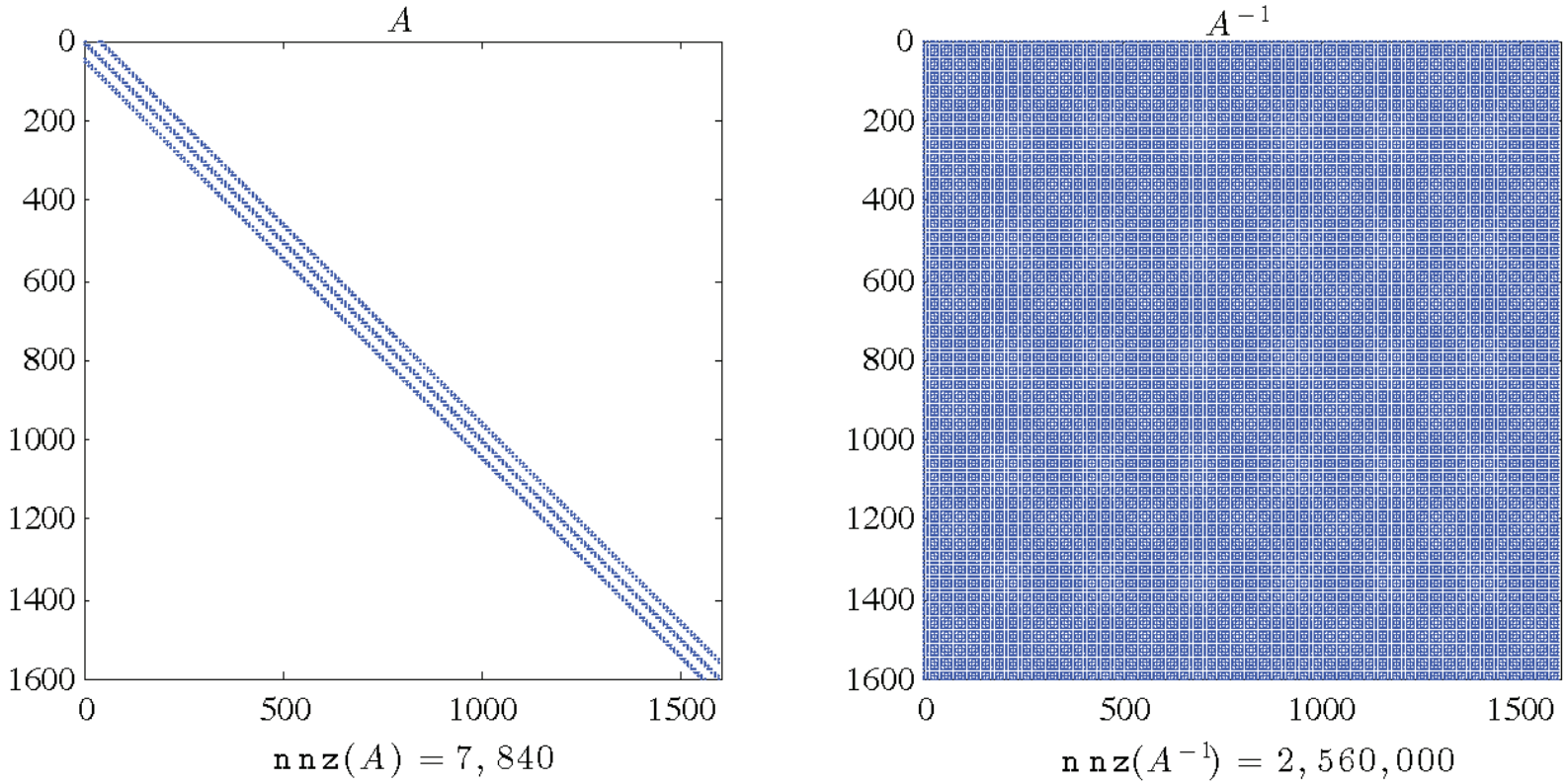


Figure 2: Destruction of sparsity of A ($n = 1600$).

More precisely | Prof. Rump formulated the following challenge:

> Derive a verification algorithm which computes an inclusion of the solution of a linear system with a general symmetric sparse matrix of dimension 10000 with condition number 10^{10} in IEEE 754 double precision, and which is no more than 10 times slower than the best numerical algorithm for that problem.

[2] S. M. Rump: Verification methods: Rigorous results using floating-point arithmetic, *Acta Numerica*, **19** (2010), 287–449.

- $\kappa(A^T A) = \kappa(A)^2$
- Treatable range in fl-pt: $\kappa(A) \lesssim \mathbf{u}^{-1} \approx 10^{16}$ in binary64
- If $\kappa(A)$ is small, then a super-fast verification method for s.p.d. matrices (to be explained) can be used after calculating $A^T A$.

In this talk we aim to do the following things:

1. survey existing verification methods for sparse linear systems.
   - monotone (including M-matrix) [e.g. heat equation]
   - H-matrix [e.g. fluid dynamics, electromagnetics]
   - symmetric and positive definite [e.g. structure analysis]
   - general symmetric

2. try to partially solve the problem for general symmetric matrices:
   - A is large, e.g. $n \geq 10000$, and sparse.
   - A is moderately ill-conditioned, e.g. $\sqrt{\mathbf{u}^{-1}} < \kappa(A) < \mathbf{u}^{-1}$.

Basic principles of verified numerical computations

1. Utilize results by standard (non-interval) numerical algorithms with pure floating-point arithmetic as much as possible.
   - Quality of such results are usually good.
   - There are many fast and reliable (but not verified) numerical libraries such as BLAS/LAPACK and sparse routines.

2. Use interval arithmetic only if absolutely necessary.
   - To avoid slowing down computational speed.
   - To avoid explosions of interval width.

$\Longrightarrow$ Leave the use of interval arithmetic as late as possible. [Wilkinson]

Current status of fast verified solutions of linear systems

dense	direct solver	general	Rump (1980), Oishi-Rump (2002)
		s.p.d.	Rump (1993), Rump-Ogita (2007)
		H-matrix	Ning-Kearfott (1997)
sparse	direct solver	general	Rump (1994)
		s.p.d.	Rump (1993), Rump-Ogita (2007)
		symmetric	Rump (1995)
	any (including iterative solver)	strictly diagonally dominant	(trivial)
		monotone*	Ogita-Oishi-Ushiro (2001)
		H-matrix	Ogita-Oishi (2006)
		TN matrix	similar to monotone
		others	–

$*$) It is not trivial to determine whether a give matrix is monotone.

Dense matrices (for reference)

Verification methods for dense matrices: (1) Krawczyk-Rump

Theorem (Rump, 1980)

Let $A \in \mathbb{R}^{n\times n}$, $R \in \mathbb{R}^{n\times n}$, $b \in \mathbb{R}^n$ and $\widetilde{x} \in \mathbb{R}^n$ be given. Let $[\epsilon] \in \mathbb{IR}^n$ be closed and bounded with $[\epsilon] \neq \emptyset$. Let $\mathrm{int}([\epsilon])$ denote the interior of $[\epsilon]$. If

$$[y] := R(b - A\widetilde{x}) + (I - RA)[\epsilon] \subseteq \mathrm{int}([\epsilon]),$$

then A is nonsingular and

$$A^{-1}b \in \widetilde{x} + [y].$$

The 1st stage of INTLAB's `verifylss` for dense linear (interval) systems.

Verification methods for dense matrices: (2) Hansen-Bliek-Rohn-Ning-Kearfott

Theorem (Ning-Kearfott, 1997)

Let an H-matrix $A \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$ be given. Let $y, z \in \mathbb{R}^n$ be defined by $y := \mathcal{M}(A)^{-1}|b|$ and $z_i := [\mathcal{M}(A)^{-1}]_{ii}$. Let $p, q \in \mathbb{R}^n$ be defined by $p_i := [\mathcal{M}(A)]_{ii} - z_i$ and $q_i := y_i/z_i - |b_i|$. Then $A^{-1}b \in [x]$ where

$$[x_i] := \frac{b_i + [-q_i, q_i]}{A_{ii} + [-p_i, p_i]}.$$

- The 2nd stage of `verifylss` for dense linear (interval) systems.
- The results may be of better quality than those of the Rump's approach for ill-conditioned linear systems; normally the quality is similar.

Verification methods for dense matrices: (3) Yamamoto

Theorem (Yamamoto, 1984)

Let $A, R \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$.
If $\|I - RA\|_\infty < 1$, then

$$|A^{-1}b - \widetilde{x}| \leq |R(b - A\widetilde{x})| + \frac{\|R(b - A\widetilde{x})\|_\infty}{1 - \|I - RA\|_\infty}|I - RA|\mathbf{e}.$$

- It is easy to implement the method.
- The results are usually as good as those of the Rump's approach.

Verification methods for dense matrices: (4) Oishi-Rump

Key estimation: $\|I - RA\|_\infty$

1. $PA \approx LU$. [$\frac{2}{3}n^3$ flops]
2. $X_L \approx L^{-1}$ and $X_U \approx U^{-1}$. [$\frac{2}{3}n^3$ flops in total]
3. $R := X_U X_L P$ (not explicitly compute it).
4. Use a priori error bounds by backward error analysis.

Evaluation in $\mathcal{O}(n^2)$ flops:

$$\|I - RA\|_\infty \leq c_1 \mathbf{u} \| |X_U|(|X_L|(|L|(|U|e))) \|_\infty + c_2 \underline{\mathbf{u}},$$

where c_1, c_2 are some computable factors and $\underline{\mathbf{u}}$ is the underflow unit.

- The same computational effort for calculating an approximate solution.

Sparse matrices

Verification methods for sparse matrices: strictly diagonally dominant

Suppose $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ is strictly (row) diagonally dominant.

Let $D := \operatorname{diag}(a_{11}, \ldots, a_{nn})$ and $\widetilde{A} := A - D$.

Setting $R := D^{-1} = \operatorname{diag}(a_{11}^{-1}, \ldots, a_{nn}^{-1})$ yields

$$\|I - RA\|_\infty = \|I - D^{-1}A\|_\infty = \|D^{-1}\widetilde{A}\|_\infty < 1,$$

since $\sum_{j \neq i} |a_{ij}| < |a_{ii}|$ for all i.

Verification methods for sparse matrices: monotone (including M-matrix)

monotone $=$ inverse nonnegative

Definition 1. [monotone] *A matrix $A \in \mathbb{R}^{n\times n}$ is called monotone if $Av \geq \mathbf{o}$ for $v \in \mathbb{R}^n$ implies $v \geq \mathbf{o}$.*

Lemma 2. *A is monotone if and only if A is nonsingular with $A^{-1} \geq O$.*

Definition 3. [M-matrix] *Let $A = (a_{ij}) \in \mathbb{R}^{n\times n}$ with $a_{ii} > 0$ and $a_{ij} \leq 0$ for $i \neq j$. Then A is called an M-matrix if A is nonsingular and $A^{-1} \geq O$.*

Theorem (Ogita-Oishi-Ushiro, 2001)

Let $A \in \mathbb{R}^{n\times n}$ with A being monotone and $b, \widetilde{y} \in \mathbb{R}^n$.
If $\|\mathbf{e} - A\widetilde{y}\|_\infty < 1$, then

$$\|A^{-1}\|_\infty \leq \frac{\|\widetilde{y}\|_\infty}{1 - \|\mathbf{e} - A\widetilde{y}\|_\infty}.$$

- To solve $Ay = \mathbf{e}$, the same solver for solving $Ax = b$ can be applied.
- $\|\mathbf{e} - A\widetilde{y}\|_\infty < 1$ is suited as a stopping criterion for iterative solvers.
- It is not trivial to determine whether A is monotone.

Proof of the theorem for monotone matrices

Since $A^{-1} \geq O$, we have

$$\begin{aligned} \|A^{-1}\|_\infty &= \||A^{-1}|e\|_\infty = \|A^{-1}e\|_\infty \\ &\leq \|A^{-1}e - \widetilde{y}\|_\infty + \|\widetilde{y}\|_\infty \\ &\leq \|A^{-1}\|_\infty \|e - A\widetilde{y}\|_\infty + \|\widetilde{y}\|_\infty. \end{aligned}$$

This yields

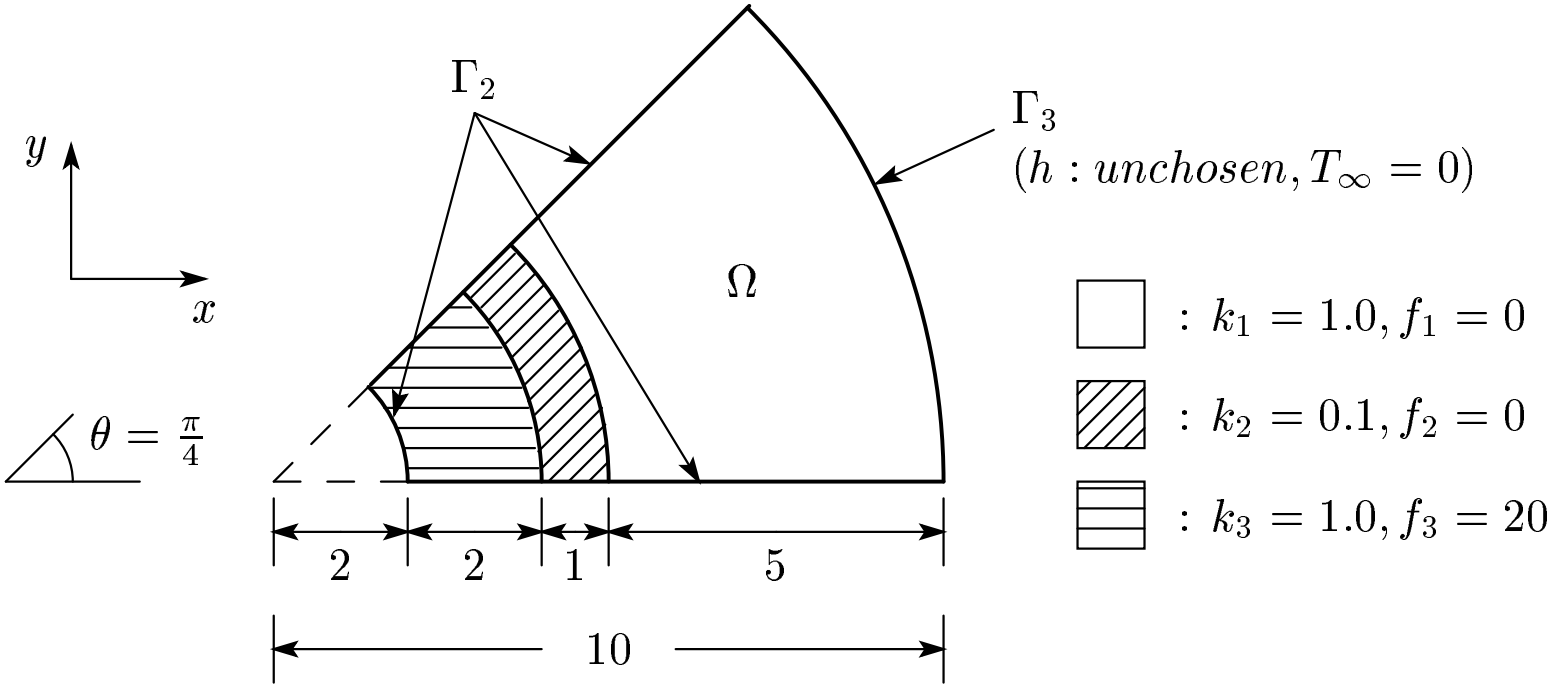
$$(1 - \|e - A\widetilde{y}\|_\infty)\|A^{-1}\|_\infty \leq \|\widetilde{y}\|_\infty.$$

If $\|e - A\widetilde{y}\|_\infty < 1$, then

$$\|A^{-1}\|_\infty \leq \frac{\|\widetilde{y}\|_\infty}{1 - \|e - A\widetilde{y}\|_\infty}. \qquad \square$$

Numerical results (1)

- A, b: from discretizing 2-D Poisson's equation by FEM
- The problem size n is varied from $10,000$ to $250,000$.
- Solver: MICCG method
  - stopping criteria: $\dfrac{\|b - A\widetilde{x}\|_2}{\|b\|_2} \leq 10^{-12}, \quad \|e - A\widetilde{y}\|_\infty \leq 10^{-3}$



$$
\begin{cases}
\mathrm{div}\{-k \cdot \mathrm{grad}(u)\} = f & \text{in } \Omega \\
\{-k \cdot \mathrm{grad}(u)\} \times \mathbf{n} = 0 & \text{on } \Gamma_2 \\
\{-k \cdot \mathrm{grad}(u)\} \times \mathbf{n} = h(u - T_\infty) & \text{on } \Gamma_3
\end{cases}
$$

Table 1: Computing time and relative error bound $\|A^{-1}b - \widetilde{x}\|_\infty / \|A^{-1}b\|_\infty$

$\dim(A)$ (n)	approx. solution [s]	verification [s]	rel. error bound
10,000	3.3	1.7	4.1×10^{-10}
40,000	27.1	10.2	2.5×10^{-9}
90,000	90.7	32.3	7.1×10^{-9}
160,000	216.2	77.0	1.6×10^{-8}
250,000	458.5	146.8	3.3×10^{-8}

Intel Celeron 566MHz CPU [Computing, Suppl. **15** (2001)]

Verification process can be faster than approximation one!

Verification methods for sparse matrices: H-matrix

For $A = (a_{ij}) \in \mathbb{R}^{n \times n}$, the comparison matrix $\mathcal{M}(A) = (\widehat{a}_{ij})$ of A is defined as

$$\widehat{a}_{ij} = \begin{cases} |a_{ij}| & (i = j) \\ -|a_{ij}| & (i \neq j) \end{cases}.$$

Definition 4. [H-matrix] *A is called an H-matrix if $\mathcal{M}(A)$ is an M-matrix.*

Lemma 5. *A is an H-matrix if and only if there exists a vector $v > \mathbf{o}$ such that $\mathcal{M}(A)v > \mathbf{o}$.*

Lemma 6. *If A is an H-matrix, then $|A^{-1}| \leq \mathcal{M}(A)^{-1}$.*

From Lemma 6, it follows that

$$\|A^{-1}\|_\infty \leq \|\mathcal{M}(A)^{-1}\|_\infty.$$

How to determine whether A is an H-matrix?

Suppose we do not know whether A is an H-matrix.

Put $\widehat{A} := \mathcal{M}(A)$. (At least $\widehat{A}$ is an L-matrix for any A with nonzero diagonals; $\widehat{a}_{ij} > 0$ for $i = j$ ($\widehat{a}_{ii} > 0$) and $\widehat{a}_{ij} \le 0$ for $i \ne j$.)

There are some possibilities:

1. Use an approximation $\widetilde{v}$ of the eigenvector corresponding to the minimum eigenvalue of $\widehat{A}$ (that is the Perron vector of $\widehat{A}^{-1}$ if A is an H-matrix) [Rump, 2012], or

2. Use an approximate solution of $\widehat{A}v = \mathbf{e}$. [Neumaier, 1999]
   $\Rightarrow \quad \|\mathbf{e} - \widehat{A}\widetilde{v}\|_\infty < 1$ implies $\widehat{A}\widetilde{v} > \mathbf{o}$.

Verification methods for sparse matrices: another approach

For symmetric A

$\lambda_i(A)$: eigenvalues of A

$$\|A^{-1}\|_2 = \frac{1}{\min|\lambda_i(A)|}$$

For non-symmetric A

$\sigma_i(A)$: singular values of A ($= \sqrt{\lambda_i(A^T A)}$)

$$\|A^{-1}\|_2 = \frac{1}{\min \sigma_i(A)}$$

Verification methods for sparse matrices: symmetric and positive definite

Rump's algorithm

1. Set $\alpha := \psi\mathbf{u} \cdot \mathrm{tr}(A)$. ($\psi$: computable)
2. Execute a Cholesky factorization for $A - 2\alpha I \approx LL^T$.
3. If succeeded, then $\lambda_{\min} \geq \alpha$.

An a priori error estimate by a backward error analysis:

$$\|LL^T - (A - 2\alpha I)\|_2 \leq \psi\mathbf{u} \cdot \mathrm{tr}(A - 2\alpha I) \leq \psi\mathbf{u} \cdot \mathrm{tr}(A) = \alpha$$

$\Longrightarrow$ INTLAB function: `verifylss`, `isspd`

Property

- Only one fl-pt Cholesky factorization $\texttt{chol}(A - 2\alpha I)$ is nesessary.
  (Direct sparse solvers can be used.) Super-fast!

- If $\texttt{chol}(A - 2\alpha I)$ runs to completion, then it is verified that "A is positive definite". (and $\lambda_{\min}(A) \geq \alpha > 0$)
  (It is verified rigorously.)

- Even if $\texttt{chol}(A - 2\alpha I)$ failed, it is not verified that "A is not positive definite".
  (A may be positive definite, although it is unlikely.)

Numerical results (2)

Test matrices: University of Florida Sparse Matrix Collection

Computer environment:
CPU: Intel Dual-Core Xeon 2.80GHz $\times$ 4 processors
Memory: 32GB
OS: Red Hat Enterprise Linux WS
Software: Matlab Version 7.1.0.183 (R14) Service Pack 3

name	n	bw w/wo RCM	time (sec)
ship_003	121,728	3659/3659	260
shipsec1	140,874	5238/5238	538
cfd2	123,440	2179/4333	127
af_shell(3,4,7,8)	504,855	2470/4909	633
apache2	715,176	2993/65837	1176

Verification methods for sparse matrices: general symmetric

Rump's algorithm

1. Estimate the smallest magnitude eigenvalue (denoted by $\widetilde{\tau}_1$).
2. Set $\alpha := 0.9 \cdot |\widetilde{\tau}_1|$.
3. Execute an $\mathrm{LDL}^{\mathrm{T}}$ factorization for $A - \alpha I \approx L_1 D_1 L_1^T$.
4. Compute $\beta_1 \geq \|L_1 D_1 L_1^T - (A - \alpha I)\|_2$.
5. Check the inertia of D_1.
6. Execute an $\mathrm{LDL}^{\mathrm{T}}$ factorization for $A + \alpha I \approx L_2 D_2 L_2^T$.
7. Compute $\beta_2 \geq \|L_2 D_2 L_2^T - (A + \alpha I)\|_2$.
8. Check the inertia of D_2.
9. Compute a lower bound of $\min |\lambda_i(A)|$: $\underline{\sigma} \geq \alpha - \max\{\beta_1, \beta_2\}$.

$$\Longrightarrow \quad \text{a little unstable: } \kappa(A) \leq \kappa(A \pm \alpha I)$$

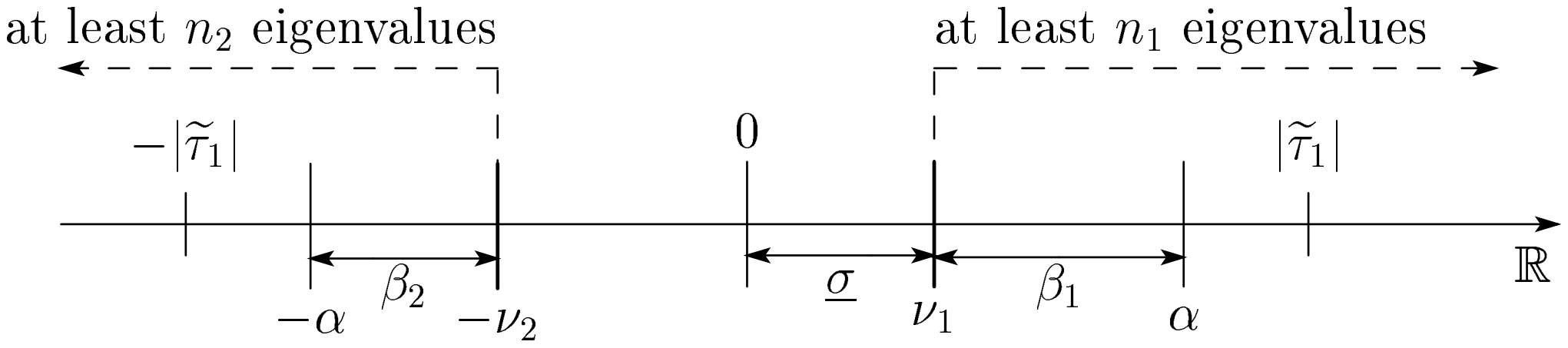


Figure 3: Lower bound of the smallest magnitude eigenvalue

A similar approach for bounding eigenvalues can be found in
[3] N. Yamamoto: A simple method for error bounds of eigenvalues of symmetric matrices, *Linear Alg. Appl.*, **324** (2001), 227–234.

Verification methods for sparse matrices: non-symmetric

The following three approaches are known (Rump):

1. $B = A^T A$ and apply the super-fast method for s.p.d. matrices.

2. $A = LDM^T \quad \Rightarrow \quad \sigma_1(A) \geq \sigma_1(L) \cdot \sigma_1(D) \cdot \sigma_1(M)$.
   - In practice, $A = \widetilde{L}\widetilde{D}\widetilde{M}^T + \Delta$ (due to rounding errors) and

$$\sigma_1(A) \geq \sigma_1(\widetilde{L}) \cdot \sigma_1(\widetilde{D}) \cdot \sigma_1(\widetilde{M}) - \|\Delta\|_2.$$

3. $G := \begin{bmatrix} O & A^T \\ A & O \end{bmatrix}$ and apply any method for symmetric matrices to G.
   - $\{\lambda_i(G), 1 \leq i \leq 2n\} = \{\pm\sigma_j(A), 1 \leq j \leq n\}. \quad \Rightarrow \quad \kappa(G) = \kappa(A)$
   - For small α, an $\mathrm{LDL^T}$ factorization for $G - \alpha I$ is a little unstable.

A new approach for sparse matrices

Lower bound of the smallest singular value

- **Present status**: Few methods of obtaining $\underline{\sigma} \leq \sigma_1(A)$ are known except some methods by Rump based on $\mathrm{LDL}^{\mathrm{T}}$ factorization.

- **Special case**: A super-fast verification method for SPD matrices by Rump using Cholesky factorization.
  – applicable up to $\kappa(A) \sim \mathbf{u}^{-1}/\psi$ where $\psi := \max_i \mathrm{nnz}(L(i,:))$ for a Cholesky factor L.

- **Suboptimal approach**: use of A^TA or AA^T.
  – An obvious drawback: it squares the condition number of A, so that applicable up to $\kappa(A) \sim 1/\sqrt{\mathbf{u}} \sim 10^8$.

Preliminaries

Theorem 7. [eigenvalue perturbation] *Let A and B be real symmetric $n \times n$ matrices. Then it holds for $i = 1, 2, \ldots, n$*

$$|\lambda_i(A) - \lambda_i(B)| \leq \|A - B\|_2.$$

Theorem 8. *Let $A = A^T \in \mathbb{R}^{n \times n}$. For some $\alpha \in \mathbb{R}$, suppose*

$$A - \alpha I = XDX^T$$

where X is some nonsingular matrix and $D \in \mathbb{R}^{n \times n}$. Then the inertia of D is equivalent to a triplet of the number of eigenvalues of A which are larger than, smaller than or equal to α.

Theorem 9. [Lehmann bounds] *Let* $A = A^T \in \mathbb{R}^{n\times n}$. *Let* λ_i, $1 \leq i \leq n$, *be eigenvalues of* A *with*

$$\lambda_1 \leq \cdots \leq \lambda_n.$$

Suppose $\nu \in \mathbb{R}$ *satisfies* $\lambda_k < \nu \leq \lambda_{k+1}$ *for some* k. *Let* X *be a real* $n \times k$ *matrix of full rank. Put* $A_1 = X^T X$, $A_2 = X^T A X$, $A_3 = X^T A^2 X$, $B_1 = \nu A_1 - A_2$ *and* $B_2 = \nu^2 A_1 - 2\nu A_2 + A_3$. *Let* μ_j, $1 \leq j \leq k$, *be generalized eigenvalues of* (B_1, B_2) *with*

$$\mu_1 \leq \cdots \leq \mu_k.$$

If B_1 *is positive definite, then it holds for* $j = 1, \ldots, k$ *that*

$$\lambda_{k-j+1} \geq \nu - \frac{1}{\mu_j}.$$

Principle of the proposed algorithm

We try to derive a verification algorithm which is

- fast (comparable with the cost for $\mathrm{LDL}^{\mathrm{T}}$ factorization)
- stable (applicable for cases $\kappa(A) > 10^{10}$)

For this purpose,

- Find two approximate eigenvalues $\widetilde{\tau}_k, \widetilde{\tau}_{k+1}$ where the gap $|\widetilde{\tau}_{k+1}| - |\widetilde{\tau}_k|$ is sufficiently large.
- Use block $\mathrm{LDL}^{\mathrm{T}}$ factorizations and their a priori error estimates.
- Apply Lehmann bounds for $\widetilde{\tau}_j$, $j = 1, \ldots, k$.

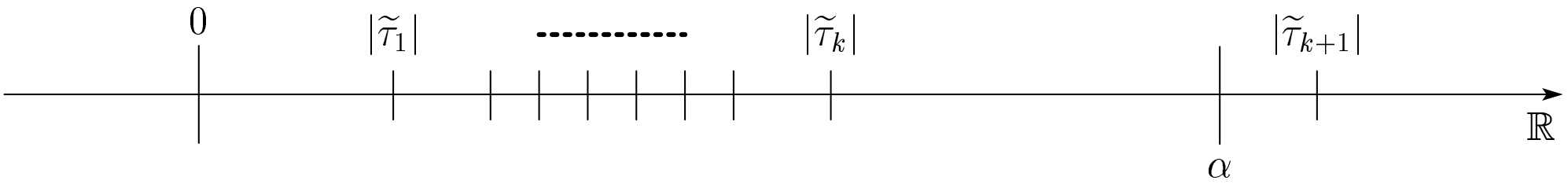


Figure 4: Distribution of absolute values of eigenvalues.

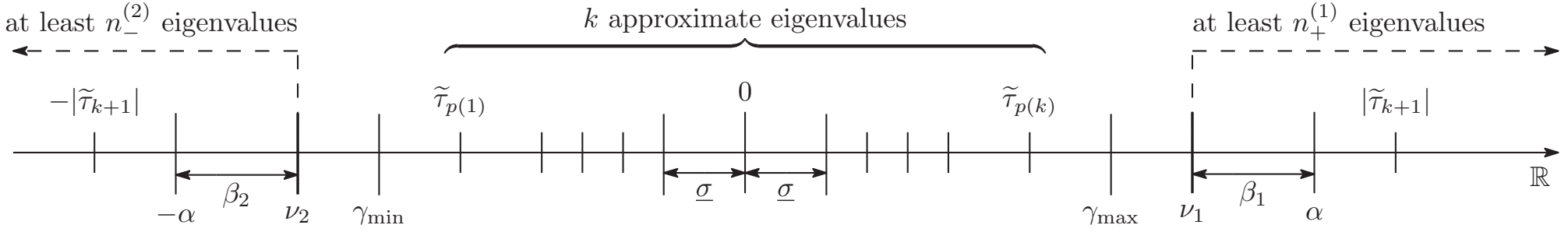


Figure 5: Distribution of eigenvalues around zero.

Rounding error analysis on block $\mathrm{LDL}^{\mathrm{T}}$ factorizations

Block $\mathrm{LDL}^{\mathrm{T}}$ factorization: $PAP^T = LDL^T$, where

$$D = \begin{bmatrix} D_1 & & & \\ & D_2 & & \\ & & \ddots & \\ & & & D_\ell \end{bmatrix}, \; L = \begin{bmatrix} L_{11} & & & \\ L_{21} & L_{22} & & \\ \vdots & \vdots & \ddots & \\ L_{\ell 1} & L_{\ell 2} & \cdots & L_{\ell\ell} \end{bmatrix}$$

Each D_i and L_{ii} is a 1×1 or 2×2 block, with L_{ii} being 1 or the 2×2 identity matrix, respectively. The rest of L is partitioned accordingly.

There are several methods with different pivoting strategies:

- Bunch–Parlett (1971)
- Bunch–Kaufman (1977)
- others

For the rounding error analysis, we need a backward error bound for the solution of linear systems involving 2×2 pivots.

We assume that the 2×2 linear system $Ey = f$ is solved successfully with a computed solution $\widetilde{y}$ satisfying

$$(E + \Delta E)\widetilde{y} = f, \quad |\Delta E| \leq \epsilon_c |E| \tag{1}$$

for some constant $\epsilon_c > 0$.

Under some conditions we can prove that the condition (1) is rigorously satisfied with

$$\epsilon_c = \begin{cases} 4\gamma_2 & \text{(GEPP)} \\ \frac{1}{6}\gamma_{298} & \text{(the explicit inverse without scaling)} \\ \frac{1}{6}\gamma_{556} & \text{(the explicit inverse with scaling)} \end{cases}$$

where $\gamma_m = m\mathbf{u}/(1 - m\mathbf{u})$ [$\approx m\mathbf{u}$ for not so large m].

Theorem 10. [Ogita-Rump] *Let* $A = A^T \in \mathbb{F}^{n\times n}$. *Let* $L = (l_{ij})$, $D = (d_{ij})$ *and* P *be a computed block* $\mathrm{LDL^T}$ *factors of* A. *Suppose the condition (1) is satisfied for some constant* $\epsilon_c > 0$. *For* $1 \le i, j \le n$ *define*

$$s(i,j) := |\{k \in \mathbb{N} : 1 \le k < \min(i,j) \text{ and } l_{ik}d_{kk}l_{jk} \neq 0\}|$$

and denote

$$\alpha_{ij} := \begin{cases} \gamma_{s(i,j)+1} & \text{if } s(i,j) \neq 0 \\ 0 & \text{otherwise} \end{cases}.$$

Put $\epsilon_2 = \max\{\epsilon_c, \overline{\alpha}\}$ *where* $\overline{\alpha} = \max_{i,j} \alpha_{ij}$ *for* $1 \le i, j \le n$. *Then it holds that*

$$|PAP^T - LDL^T| \le \epsilon_2(P|A|P^T + |L||D||L|^T).$$

Proposed algorithm

1. Find two approximate eigenvalues $\widetilde{\tau}_k, \widetilde{\tau}_{k+1}$ where the gap $|\widetilde{\tau}_{k+1}| - |\widetilde{\tau}_k|$ is sufficiently large.
2. Take α in $(|\widetilde{\tau}_k|, |\widetilde{\tau}_{k+1}|)$.
3. Execute a block $\mathrm{LDL^T}$ for $P(A - \alpha I)P^T \approx L_1 D_1 L_1^T$.
4. Compute $\beta_1 \geq \|L_1 D_1 L_1^T - P(A - \alpha I)P^T\|_2$.
5. Check the inertia of D_1.
6. Execute a block $\mathrm{LDL^T}$ for $P(A + \alpha I)P^T \approx L_2 D_2 L_2^T$.
7. Compute $\beta_2 \geq \|L_2 D_2 L_2^T - P(A + \alpha I)P^T\|_2$.
8. Check the inertia of D_2.
9. Apply Lehmann bounds for $\widetilde{\tau}_j$, $j = 1, \ldots, k$.

β_1, β_2: computed by a priori error estimates

Numerical results (for reference)

We evaluate the performance of the proposed algorithm.
CPU: 2.66 GHz Intel Core 2 Duo, Memory: 8GB

We implement a hybrid algorithm:
Stage-1: Rump's algorithm
Stage-2: Proposed algorithm

Example: Random sparse symmetric matrices having 5 clustered eigenvalues in $[1, 1.1]$ and $n - 5$ eigenvalues in $[10^2, \mathrm{cnd}]$ in magnitude.

$$|\lambda(A)| = \{1, 1.025, 1.05, 1.075, 1.1, 10^2, \ldots, \mathrm{cnd}\}$$

$\Longrightarrow$ $\kappa(A) = \mathrm{cnd}$, symmetric and indefinite.

Example: random sparse, density $= 5/\mathrm{n}$, $\min|\lambda(A)| = 1$

Computing time (sec) and lower bound of the smallest magnitude eigenvalue

n	cond	Stage-1	Stage-2	lower bound
10,000	1e12	0.86	--	0.916
10,000	1e13	failed	1.72	0.998
10,000	1e14	failed	1.41	0.974
20,000	1e15	failed	failed	--
20,000	1e12	1.80	--	0.589
20,000	1e13	failed	5.40	0.999
20,000	1e14	failed	6.54	0.996
20,000	1e15	failed	failed	--

n	cond	Stage-1	Stage-2	lower bound
50,000	1e12	8.03	--	0.387
50,000	1e13	failed	74.89	0.999
50,000	1e14	failed	48.88 *	0.997
50,000	1e15	failed	failed	--
100,000	1e12	8.81	--	0.426
100,000	1e13	failed	153.62	0.999
100,000	1e14	failed	485.85 *	0.999
100,000	1e15	failed	failed	--

Conclusions

- There exists efficient (practically useful) verification methods for sparse matrices having special property.
- Verified numerical computation for general sparse linear systems is still difficult.
- Nevertheless, (we) never ever give up!

Thanks for your kind attention!