Yet another method for solving interval linear equations

Milan Hladík

Department of Applied Mathematics
Faculty of Mathematics and Physics,
Charles University in Prague, Czech Republic
http://kam.mff.cuni.cz/~hladik/

SWIM 2015
Prague
June 9–11, 2015
An interval matrix $\mathbf{A}$ is defined as

$$
\mathbf{A} := [\mathbf{A}, \mathbf{\overline{A}}] = \{ \mathbf{A} \in \mathbb{R}^{m \times n} : \mathbf{A} \leq \mathbf{A} \leq \mathbf{\overline{A}} \},
$$

The center and radius of $\mathbf{A}$ are respectively defined as

$$
\mathbf{A}_c := \frac{1}{2} (\mathbf{A} + \mathbf{\overline{A}}), \quad \mathbf{A}_\Delta := \frac{1}{2} (\mathbf{\overline{A}} - \mathbf{A}).
$$

The set of all $m$-by-$n$ interval matrices is denoted by $\mathbb{IR}^{m \times n}$. 
An interval matrix $A$ is defined as

$$A := [A, \overline{A}] = \{ A \in \mathbb{R}^{m \times n} : A \leq A \leq \overline{A} \},$$

The center and radius of $A$ are respectively defined as

$$A_c := \frac{1}{2}(A + \overline{A}), \quad A_\Delta := \frac{1}{2}(\overline{A} - A).$$

The set of all $m$-by-$n$ interval matrices is denoted by $\mathbb{IR}^{m \times n}$.

The magnitude of an $A \in \mathbb{IR}^{m \times n}$ is defined as

$$\text{mag}(A) := \max(|A|, |\overline{A}|).$$
An interval matrix $\mathbf{A}$ is defined as

$$\mathbf{A} := [\underline{A}, \bar{A}] = \{ A \in \mathbb{R}^{m \times n} : \underline{A} \leq A \leq \bar{A} \},$$

The center and radius of $\mathbf{A}$ are respectively defined as

$$\mathbf{A}_c := \frac{1}{2}(\underline{A} + \bar{A}), \quad \mathbf{A}_\Delta := \frac{1}{2}(\bar{A} - \underline{A}).$$

The set of all $m$-by-$n$ interval matrices is denoted by $\mathbb{IR}^{m \times n}$.

The magnitude of an $\mathbf{A} \in \mathbb{IR}^{m \times n}$ is defined as

$$\text{mag}(\mathbf{A}) := \max(\|\underline{A}\|, \|\bar{A}\|).$$

The comparison matrix of $\mathbf{A} \in \mathbb{IR}^{n \times n}$ is the matrix $\langle \mathbf{A} \rangle \in \mathbb{R}^{n \times n}$ with entries

$$\langle \mathbf{A} \rangle_{ii} := \min\{|a| : a \in \mathbf{a}_{ii} \}, \quad i = 1, \ldots, n,$$

$$\langle \mathbf{A} \rangle_{ij} := -\text{mag}(\mathbf{a}_{ij}), \quad i \neq j.$$
Interval linear equations

Definition

Let $A \in \mathbb{IR}^{n \times n}$, $b \in \mathbb{IR}^n$, and consider a set of systems of linear equations

$$Ax = b, \quad A \in A, \ b \in b,$$

The corresponding solution set is defined as

$$\Sigma := \{x \in \mathbb{R}^n : \exists A \in A \exists b \in b : Ax = b\}.$$

By $\Sigma$ we denote the interval hull of $\Sigma$, i.e., the smallest interval enclosure of $\Sigma$ with respect to inclusion.
Interval linear equations

**Definition**

Let $A \in \mathbb{IR}^{n \times n}$, $b \in \mathbb{IR}^n$, and consider a set of systems of linear equations

$$Ax = b, \quad A \in A, \ b \in b,$$

The corresponding solution set is defined as

$$\Sigma := \{x \in \mathbb{R}^n : \exists A \in A \exists b \in b : Ax = b\}.$$

By $\Sigma$ we denote the interval hull of $\Sigma$, i.e., the smallest interval enclosure of $\Sigma$ with respect to inclusion.

**Problem formulation**

The aim is to compute $\Sigma$ or an as tight as possible enclosure of $\Sigma$ by an interval vector $x \in \mathbb{IR}^n$, meaning that $\Sigma \subseteq x$. 
Assumption

Assume that $A_c = I_n$. 
Assumption

Assume that $A_c = I_n$.

- Easily satisfied by preconditioning $A = b$ by $A_c^{-1}$.
- Rigorously precondition as
  \[ A'x = b', \quad A' \in [I_n - \text{mag}(I_n - RA), I_n + \text{mag}(I_n - RA)], \quad b' \in Rb. \]
  where $R \approx A_c^{-1}$. 
Assumption

Assume that $A_c = I_n$.

- Easily satisfied by preconditioning $A = b$ by $A_c^{-1}$.
- Rigorously precondition as
  \[ A'x = b', \quad A' \in [I_n - \text{mag}(I_n - RA), I_n + \text{mag}(I_n - RA)], \quad b' \in Rb. \]
where $R \approx A_c^{-1}$.

Consequences

- $\Sigma$ is bounded (i.e., $A$ contains no singular matrix) if and only if the spectral radius $\rho(A_\Delta) < 1$,
- $\Sigma$ can be determined in polynomial time.
Two (equivalent) formulas for computing the interval hull $\Sigma$:

- Hansen–Bliek–Rohn method (1993),
Two (equivalent) formulas for computing the interval hull $\Sigma$:

- Hansen–Bliék–Rohn method (1993),

Denote:

$$u := \langle A \rangle^{-1} \text{mag}(b),$$
$$d_i := (\langle A \rangle^{-1})_{ii}, \quad i = 1, \ldots, n,$$
$$\alpha_i := \langle a_{ii} \rangle - 1/d_i, \quad i = 1, \ldots, n.$$
Interval hull computation

Two (equivalent) formulas for computing the interval hull $\Sigma$:
- Hansen–Bliek–Rohn method (1993),

Denote:

\[
\begin{align*}
    u & := \langle A \rangle^{-1} \text{mag}(b), \\
    d_i & := (\langle A \rangle^{-1})_{ii}, \quad i = 1, \ldots, n, \\
    \alpha_i & := \langle a_{ii} \rangle - 1/d_i, \quad i = 1, \ldots, n.
\end{align*}
\]

Theorem (Ning–Kearfott, 1997)

\[
\Sigma_i = \frac{b_i + (u_i/d_i - \text{mag}(b_i))[-1, 1]}{a_{ii} + \alpha_i[-1, 1]}, \quad i = 1, \ldots, n.
\]
Interval hull computation

Two (equivalent) formulas for computing the interval hull $\Sigma$:
- Hansen–Bliek–Rohn method (1993),

Denote:

\[
\begin{align*}
    u &:= \langle A \rangle^{-1} \text{mag}(b), \\
    d_i &:= (\langle A \rangle^{-1})_{ii}, \quad i = 1, \ldots, n, \\
    \alpha_i &:= \langle a_{ii} \rangle - 1/d_i, \quad i = 1, \ldots, n.
\end{align*}
\]

**Theorem (Ning–Kearfott, 1997)**

\[
\Sigma_i = \frac{b_i + (u_i/d_i - \text{mag}(b_i))[−1, 1]}{a_{ii} + \alpha_i[−1, 1]}, \quad i = 1, \ldots, n.
\]

**Disadvantage**
- We have to safely compute the inverse of $\langle A \rangle$. 
Iteration methods can usually be expressed by an operator $\mathcal{P} : \mathbb{IR}^n \mapsto \mathbb{IR}^n$

$$(x \cap \Sigma) \subseteq \mathcal{P}(x).$$
Interval operators

Iteration methods can usually be expressed by an operator $\mathcal{P} : \mathbb{IR}^n \mapsto \mathbb{IR}^n$

$$(x \cap \Sigma) \subseteq \mathcal{P}(x).$$

Basically, iterations then can have the plain form $x \mapsto \mathcal{P}(x)$, or the form with intersections $x \mapsto \mathcal{P}(x) \cap x$. 
Interval operators

Iteration methods can usually be expressed by an operator \( \mathcal{P} : \mathbb{IR}^n \mapsto \mathbb{IR}^n \):

\[
(x \cap \Sigma) \subseteq \mathcal{P}(x).
\]

Basically, iterations then can have the plain form \( x \mapsto \mathcal{P}(x) \), or the form with intersections \( x \mapsto \mathcal{P}(x) \cap x \).

Known operators

- The Krawczyk operator
  \[
  x \mapsto b + (I_n - A)x.
  \]

- Denote by \( D \) the interval diagonal matrix, whose diagonal is the same as that of \( A \), and \( A' \) is used for the interval matrix \( A \) with zero diagonal. The interval Jacobi operator reads
  \[
  x \mapsto D^{-1}(b - A'x).
  \]

- The interval Gauss–Seidel operator is similar to Jacobi, but evaluated raises by evaluating successively.
By $\mathbf{x}^{\text{GS}}$ and $\mathbf{x}^{\text{K}}$ we denote the limit enclosures computed by the interval Gauss–Seidel and Krawczyk methods, respectively.
By $x^{GS}$ and $x^K$ we denote the limit enclosures computed by the interval Gauss–Seidel and Krawczyk methods, respectively.

**Theorem**

Recall

$$u := \langle A \rangle^{-1} \text{mag}(b).$$

We have

$$x^{GS} = D^{-1}(b + \text{mag}(A')u[-1, 1]),$$

$$x^K = b + A_\Delta u[-1, 1].$$

Moreover,

$$u = \text{mag}(\Sigma) = \text{mag}(x^{GS}) = \text{mag}(x^K).$$
Limiting enclosures

By $x^{GS}$ and $x^K$ we denote the limit enclosures computed by the interval Gauss–Seidel and Krawczyk methods, respectively.

**Theorem**

Recall

\[ u := \langle A \rangle^{-1} \text{mag}(b). \]

We have

\[ x^{GS} = D^{-1}(b + \text{mag}(A')u[-1, 1]), \]
\[ x^K = b + A_\Delta u[-1, 1]. \]

Moreover,

\[ u = \text{mag}(\Sigma) = \text{mag}(x^{GS}) = \text{mag}(x^K). \]

**Corollary**

We have $\Sigma \in [-u, u]$. 
Limiting enclosures

Example (Typical case)

The solution set, the preconditioned solution set and its enclosure.
**Theorem (Hladík, 2014)**

Let $\Sigma \subseteq \mathbb{x} \in \mathbb{IR}^n$. Then

$$\Sigma_i \subseteq \frac{b_i - \sum_{j \neq i} a_{ij}x_j + [\gamma_i, -\gamma_i]u_i}{a_{ii} + \gamma_i[-1, 1]}$$

for every $\gamma_i \in [0, \alpha_i]$, and $i = 1, \ldots, n$, where

$$d_i := (\langle A \rangle^{-1})_{ii}, \quad i = 1, \ldots, n,$$

$$\alpha_i := \langle a_{ii} \rangle - 1/d_i, \quad i = 1, \ldots, n.$$
Theorem (Hladík, 2014)

Let $\Sigma \subseteq \mathbb{R}^n$. Then

$$\Sigma_i \subseteq \frac{b_i - \sum_{j \neq i} a_{ij} x_j + [\gamma_i, -\gamma_i] u_i}{a_{ii} + \gamma_i[-1, 1]}$$

for every $\gamma_i \in [0, \alpha_i]$, and $i = 1, \ldots, n$, where

$$d_i := (\langle A^{-1} \rangle)_{ii}, \quad i = 1, \ldots, n,$$

$$\alpha_i := \langle a_{ii} \rangle - 1/d_i, \quad i = 1, \ldots, n.$$

Remarks

- Generalization of the interval Gauss–Seidel operator (let $\gamma := 0$).
- Its performance depends on computation of $u$ and $d$. Tight lower bounds are sufficient.

---

Theorem

We have

\[ u \geq \text{mag}(b) + A_\Delta (\text{mag}(b) + A_\Delta \text{mag}(b)), \]
\[ d_i \geq \bar{d}_i := \bar{a}_{ii}/(1 - ((A_\Delta)^2)_{ii}), \quad i = 1, \ldots, n. \]
### Theorem

*We have*

\[
    u \geq \text{mag}(b) + A_{\Delta}(\text{mag}(b) + A_{\Delta}\text{mag}(b)),
\]

\[
    d_i \geq d_i := \frac{\bar{a}_{ii}}{1 - (A_{\Delta})^2_{ii}}, \quad i = 1, \ldots, n.
\]

### Remarks

- Both bounds computable in time $O(n^2)$.
- For $\gamma_i > 0$, it outperforms the interval Gauss–Seidel operator if $x$ is sufficiently tight.
Theorem

We have

\[ u \geq \text{mag}(b) + A_\Delta (\text{mag}(b) + A_\Delta \text{mag}(b)), \]

\[ d_i \geq d_i := \bar{a}_{ii} / (1 - ((A_\Delta)^2)_{ii}), \quad i = 1, \ldots, n. \]

Remarks

- Both bounds computable in time \( O(n^2) \).
- For \( \gamma_i > 0 \), it outperforms the interval Gauss–Seidel operator if \( x \) is sufficiently tight.

Efficient implementation of the new operator

Call one iteration of the operator on the initial box \([-u, u]\).
New enclosing method

Algorithm (Magnitude method)

1. Compute $\mathbf{u}$, an enclosure to the solution of $\langle \mathbf{A} \rangle \mathbf{u} = \text{mag}(\mathbf{b})$.
2. Calculate $d$, a lower bound on $d$ (e.g., by the above theorem).
3. Evaluate
   \begin{align*}
   x^*_i &:= \frac{\mathbf{b}_i + (\sum_{j \neq i} \mathbf{a}_{ij} \bar{u}_j - \gamma_i \bar{u}_i) [-1, 1]}{\mathbf{a}_{ii} + \gamma_i [-1, 1]}, \quad i = 1, \ldots, n, \\
   \text{where} \quad \gamma_i &:= \langle \mathbf{a}_{ii} \rangle - 1/d_i.
   \end{align*}
New enclosing method

Algorithm (Magnitude method)

1. Compute $u$, an enclosure to the solution of $\langle A \rangle u = \text{mag}(b)$.
2. Calculate $d$, a lower bound on $d$ (e.g., by the above theorem).
3. Evaluate

$$x_i^* := \frac{b_i + \left( \sum_{j \neq i} a_{ij} \Delta u_j - \gamma_i u_i \right) [-1, 1]}{a_{ii} + \gamma_i [-1, 1]}, \quad i = 1, \ldots, n,$$

where $\gamma_i := \langle a_{ii} \rangle - 1/d_i$.

Theorem

If $u$ and $d$ are calculated exactly, then $x^* = \Sigma$. 
New enclosing method

Algorithm (Magnitude method)

1. Compute \( \mathbf{u} \), an enclosure to the solution of \( \langle \mathbf{A} \rangle \mathbf{u} = \text{mag}(\mathbf{b}) \).
2. Calculate \( \mathbf{d} \), a lower bound on \( d \) (e.g., by the above theorem).
3. Evaluate

\[
x_i^* := \frac{\mathbf{b}_i + (\sum_{j \neq i} \mathbf{a}_{ij} \mathbf{u}_j - \gamma_i \mathbf{u}_i) [-1, 1]}{\mathbf{a}_{ii} + \gamma_i [-1, 1]}, \quad i = 1, \ldots, n,
\]

where \( \gamma_i := \langle \mathbf{a}_{ii} \rangle - 1/d_i \).

Theorem

If \( u \) and \( d \) are calculated exactly, then \( x^* = \Sigma \).

Theorem

We have \( x^* \subseteq x^{GS} \). If \( \gamma = 0 \), then equality holds.
Numerical experiments

Example

- Randomly generated examples for various dimensions and interval radii.
- The entries of $A_c$ and $b_c$ were generated randomly in $[-10, 10]$ with uniform distribution.
- All radii of $A$ and $b$ were equal to the parameter $\delta > 0$.
- The computations were carried out in MATLAB with INTLAB.
- Tightness of the computed enclosure $x$ was measured by

$$\frac{\sum_{i=1}^{n} x_i \Delta}{\sum_{i=1}^{n} \sum_{j=1}^{n} i \Delta}.$$ 

(Thus, the closer to 1, the sharper enclosure.)
### Numerical experiments

#### Example (Tightness of enclosures for randomly generated data)

<table>
<thead>
<tr>
<th>$n$</th>
<th>$\delta$</th>
<th>verifylss</th>
<th>Gauss-Seidel</th>
<th>magnitude</th>
<th>magnitude ($\gamma = 0$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1</td>
<td>1.1520</td>
<td>1.1510</td>
<td>1.09548</td>
<td>1.1196</td>
</tr>
<tr>
<td>5</td>
<td>0.1</td>
<td>1.08302</td>
<td>1.01645</td>
<td>1.00591</td>
<td>1.0164</td>
</tr>
<tr>
<td>5</td>
<td>0.01</td>
<td>1.01755</td>
<td>1.00148</td>
<td>1.00037</td>
<td>1.00148</td>
</tr>
<tr>
<td>10</td>
<td>0.1</td>
<td>1.07756</td>
<td>1.02495</td>
<td>1.01107</td>
<td>1.02474</td>
</tr>
<tr>
<td>10</td>
<td>0.01</td>
<td>1.02362</td>
<td>1.00378</td>
<td>1.00132</td>
<td>1.00378</td>
</tr>
<tr>
<td>15</td>
<td>0.1</td>
<td>1.06994</td>
<td>1.03121</td>
<td>1.01755</td>
<td>1.03074</td>
</tr>
<tr>
<td>15</td>
<td>0.01</td>
<td>1.02125</td>
<td>1.00217</td>
<td>1.00047</td>
<td>1.00216</td>
</tr>
<tr>
<td>20</td>
<td>0.1</td>
<td>1.05524</td>
<td>1.03076</td>
<td>1.02007</td>
<td>1.02989</td>
</tr>
<tr>
<td>20</td>
<td>0.01</td>
<td>1.02643</td>
<td>1.00348</td>
<td>1.00097</td>
<td>1.00348</td>
</tr>
<tr>
<td>30</td>
<td>0.01</td>
<td>1.02539</td>
<td>1.00402</td>
<td>1.00129</td>
<td>1.00401</td>
</tr>
<tr>
<td>30</td>
<td>0.001</td>
<td>1.00574</td>
<td>1.00026</td>
<td>1.000039</td>
<td>1.000256</td>
</tr>
<tr>
<td>50</td>
<td>0.01</td>
<td>1.02688</td>
<td>1.00533</td>
<td>1.00226</td>
<td>1.00531</td>
</tr>
<tr>
<td>50</td>
<td>0.001</td>
<td>1.00902</td>
<td>1.00051</td>
<td>1.00011</td>
<td>1.00051</td>
</tr>
<tr>
<td>100</td>
<td>0.001</td>
<td>1.01303</td>
<td>1.00057</td>
<td>1.00013</td>
<td>1.00057</td>
</tr>
<tr>
<td>100</td>
<td>0.0001</td>
<td>1.0024988</td>
<td>1.0000274</td>
<td>1.000022</td>
<td>1.0000274</td>
</tr>
</tbody>
</table>
## Numerical experiments

### Example (Computational time in sec. for randomly generated data)

<table>
<thead>
<tr>
<th>n</th>
<th>δ</th>
<th>verifylss</th>
<th>Gauss-Seidel</th>
<th>magnitude</th>
<th>magnitude (γ = 0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1</td>
<td>3.2903</td>
<td>0.10987</td>
<td>0.004466</td>
<td>0.003429</td>
</tr>
<tr>
<td>5</td>
<td>0.1</td>
<td>0.004234</td>
<td>0.02937</td>
<td>0.004513</td>
<td>0.003502</td>
</tr>
<tr>
<td>5</td>
<td>0.01</td>
<td><strong>0.002342</strong></td>
<td>0.02500</td>
<td>0.004473</td>
<td>0.003456</td>
</tr>
<tr>
<td>10</td>
<td>0.1</td>
<td>0.018845</td>
<td>0.08370</td>
<td>0.004877</td>
<td><strong>0.003777</strong></td>
</tr>
<tr>
<td>10</td>
<td>0.01</td>
<td><strong>0.003161</strong></td>
<td>0.05305</td>
<td>0.004821</td>
<td>0.003799</td>
</tr>
<tr>
<td>15</td>
<td>0.1</td>
<td>0.246779</td>
<td>0.21868</td>
<td>0.005212</td>
<td><strong>0.004162</strong></td>
</tr>
<tr>
<td>15</td>
<td>0.01</td>
<td>0.005403</td>
<td>0.09163</td>
<td>0.005260</td>
<td><strong>0.004172</strong></td>
</tr>
<tr>
<td>20</td>
<td>0.1</td>
<td>16.9678</td>
<td>0.95238</td>
<td>0.005554</td>
<td><strong>0.004251</strong></td>
</tr>
<tr>
<td>20</td>
<td>0.01</td>
<td>0.008950</td>
<td>0.15602</td>
<td>0.005736</td>
<td><strong>0.004622</strong></td>
</tr>
<tr>
<td>30</td>
<td>0.01</td>
<td>0.019111</td>
<td>0.32294</td>
<td>0.006457</td>
<td><strong>0.005289</strong></td>
</tr>
<tr>
<td>30</td>
<td>0.001</td>
<td><strong>0.004488</strong></td>
<td>0.19544</td>
<td>0.006460</td>
<td>0.005260</td>
</tr>
<tr>
<td>50</td>
<td>0.01</td>
<td>0.210430</td>
<td>1.01155</td>
<td>0.008483</td>
<td><strong>0.007062</strong></td>
</tr>
<tr>
<td>50</td>
<td>0.001</td>
<td>0.010190</td>
<td>0.54813</td>
<td>0.008343</td>
<td><strong>0.006879</strong></td>
</tr>
<tr>
<td>100</td>
<td>0.001</td>
<td>0.044463</td>
<td>2.42025</td>
<td>0.016706</td>
<td><strong>0.014645</strong></td>
</tr>
<tr>
<td>100</td>
<td>0.0001</td>
<td><strong>0.013940</strong></td>
<td>1.48693</td>
<td>0.017089</td>
<td>0.014847</td>
</tr>
</tbody>
</table>
Conclusion

Performance

- The magnitude method overcomes the Gauss–Seidel iteration method with respect to both computational time and sharpness of enclosures.
- Compared to the INTLAB function verifylss, the magnitude method produces always tighter enclosures. Unless the input interval data are very narrow, it also overcomes verifylss with respect to computational time.
Performance

- The magnitude method overcomes the Gauss–Seidel iteration method with respect to both computational time and sharpness of enclosures.
- Compared to the INTLAB function verifylss, the magnitude method produces always tighter enclosures. Unless the input interval data are very narrow, it also overcomes verifylss with respect to computational time.

Open problems

- Extension our approach to parametric interval systems,
- Overcoming the assumption $A_c = I_n$. 