MATEMATICKO-FYZIKÁLNÍ
FAKULTA
Univerzita Karlova

BAKALÁŘSKÁ PRÁCE

David Tomandl

Řešení intervalových soustav metodou nejmenších čtverců

Katedra aplikované matematiky

Vedoucí bakalářské práce:	doc. Mgr. Milan Hladík, Ph.D.
Studijní program:	Informatika
Studijní obor:	Obecná informatika

Praha 2016

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V .... dne .... Podpis autora

Název práce: Řešení intervalových soustav metodou nejmenších čtverců

Autor: David Tomandl

Katedra: Katedra aplikované matematiky

Vedoucí bakalářské práce: doc. Mgr. Milan Hladík, Ph.D., Katedra aplikované matematiky

Abstrakt: Tato práce popisuje, porovnává a implementuje metody na zapouzdření všech řešení přeurčených soustav lineárních rovnic metodou nejmenších čtverců s tím, že vstupní data se pohybují v rámci daných intervalů. Popsána je struktura množiny řešení, ze které vyplývá návrh některých algoritmů pro výpočet intervalového obalu množiny řešení. Výpočet intervalového obalu je obecně NP-těžká úloha. Přesto existují algoritmy, které zapouzdří všechna řešení dříve než po exponenciálně mnoha krocích. Těmito algoritmy se práce zabývá. Systém řešení je tvořen symetrickou intervalovou maticí, proto je součástí práce také implementace řešičů symetrických soustav. Práce také obsahuje numerické porovnání různých přístupů. Algoritmy jsou implementovány v prostředí Matlab za použití intervalové knihovny Intlab.

Klíčová slova: intervalové soustavy rovnic, metoda nejmenších čtverců, symetrické intervalové soustavy, Matlab, Intlab

Title: Solving interval systems by the least squares method

Author: David Tomandl

Department: Department of Applied Mathematics

Supervisor: doc. Mgr. Milan Hladík, Ph.D., Department of Applied Mathematics

Abstract: This thesis is describing, comparing and implementing enclosure methods for solving overdetermined system of interval linear equations by the least squares method. Input data of the methods are within given intervals. We describe the structure of the solution set, which is the basis of algorithms for computing interval hull of the solution set. Although computation of the interval hull is NP-hard problem, there exist algorithms which encloses the solution set with less than exponential steps. We are heavily focusing on these algorithms. The solution set can be alternatively characterized as a solution to the symmetric interval system. Therefore the work includes solvers of the symmetric interval system. This thesis contains numerical experiments for comparing the methods. All methods are implemented in Matlab with utilisation of the interval toolbox Intlab.

Keywords: interval linear equations, least squares method, symmetric interval system, Matlab, Intlab

Děkuji svému vedoucímu práce doc. Mgr. Milanu Hladíkovi Ph.D. za jeho přístup, cenné rady, vypůjčené materiály a konzultace, které mi vždy ochotně poskytl.

Obsah

Úvod

Slovo interval se objevuje v mnoha oborech. Například v hudbě znamená vzdálenost mezi dvěma tóny. Tento pojem se objevuje i ve sportu. Cyklisté a běžci používají intervalový trénink, při kterém v časových úsecích mění intenzitu tréninku. V jiných sportech se například můžeme setkat s pojmem intervalový závod, kde závodníci startují za sebou v časových intervalech. Nás však bude zajímat matematický význam slova interval. V matematice tímto pojmem označuje množinu reálných čísel, která leží mezi dvěma určenými čísly, jimž říkáme meze intervalu.

Intervalová analýza není úplně nový fenomén moderní matematiky, objevuje se několikrát pod různými názvy v průběhu dějin. Například Archimedes odhadl dolní a horní mez $223/71 < \pi < 22/7$ ve 3. století př. n. l. Počítání s intervaly však nebylo tak populární jako ostatní numerické techniky. Pravidla, jak počítat s intervaly, byla publikována v roce 1931 v práci, kterou zveřejnil doktorand Cambridgeské univerzity Rosalind Cicely Young. V průběhu padesátých let byly vydány další práce věnující se využití intervalů. Za základ moderní intervalové analýzy je považována kniha Interval Analysis napsaná R. E. Moorem v roce 1966.

Postupem let začali přibývat práce věnující se intervalovým metodám; to bylo způsobeno i díky rozmachu počítačů. Intervalové metody se začaly používat i v jiných odvětvích než matematika, například v počítačové grafice, robotice nebo ekonomii.

V praxi se často setkáváme s problémy, že dochází k různým nepřesnostem (např. měření hodnot). Také počítače nemohou počítat s neomezenou přesností, ale čísla zaokrouhlují. Způsob, jak se vypořádat s nepřesnostmi, je využít intervalový přístup. Hlavní myšlenkou intervalového počítání je, že nepočítáme s konkrétními hodnotami, ale s intervaly, kde se daná hodnota určitě nachází.

Při strojové reprezentaci čísel může dojít k nepřesnostem vinou omezeného počtu bitů, pomocí kterých číslo reprezentujeme. Iracionální čísla nelze přesně reprezentovat na počítači, musíme se zpokojit se zaokrouhlením dané hodnoty. Ale i u zdánlivě nevinných čísel (např. 0.1) může dojít k nekonečnému binárnímu rozvoji. Při zaokrouhlování hodnot může dojít nastřádání chyb, takže pak o výsledné hodnotě nejsme schopni říci, do jaké míry je dobrá nebo špatná. To může mít fatální následky, dokonce může dojít i ke katastrofě. Mezi nejznámější katastrofy, které vznikly kvůli nepřesné reprezentaci čísel, patří:

- špatné načasování výbuchu americké protiraketové střely v roce 1991 ve válce v Perském zálivu způsobilo smrt 28 vojákům (kvůli binární reprezentaci čísla 0.1 došlo ke zpoždění 1/3 sekund),

- zřícení ropné plošiny v severním Norsku v roce 1991 (kvůli diskretizaci diferenciálních rovnic došlo k podcenění smykového napětí o 47%).

Možné řešení těchto problémů je reprezentovat dané číslo jako interval, o kterém víme, že jistě obsahuje původní hodnotu. V intervalovém počítání sice může nějaká chyba nastat, ale vždy máme jistotu, že výsledná hodnota se nachází ve spočteném intervalu.

Cíle práce

- uvedení metod na zapouzdření řešení intervalových soustav lineárních rovnic metodou nejmenších čtverců
- uvedení metod na zapouzdření řešení symetrických intervalových soustav lineárních rovnic
- implementace těchto metod v prostředí Matlab s využitím knihovny Intlab
- numerické porovnání těchto metod

Struktura práce

Práce se skládá ze sedmi kapitol. V první a druhé kapitole uvádíme stručný úvod do problematiky intervalové analýzy. Ve třetí kapitole se zabýváme soustavami intervalových lineárních rovnic. Ve čtvrté kapitole prezentujeme a v páté kapitole porovnáváme metody na řešení intervalových soustav lineárních rovnic metodou nejmenších čtverců. V šesté kapitole uvedeme stručný úvod do Intlabu a návod, jak používat naimplementované metody. V poslední kapitole popíšeme, jak implementujeme dané metody.

1. Intervaly

V této kapitole zavedeme základní pojmy a úvod do problematiky intervalového počtu. Tato část čerpá především z knihy Moore a kol. (2009).

1.1 Definice intervalu

V následující sekci definujeme reálné intervaly a jejich značení. Dále zavedeme reálnou intervalovou aritmetiku. O intervalové aritmetice se později již příliš zmiňovat nebudeme, ale budeme ji automaticky používat. Pokud nebude uvedeno jinak, bude pojem „interval" označovat množinu zavedenou definicí 1.

Definice 1 (Reálný interval)**.** *Mějme* $\underline{x}, \bar{x} \in \mathbb{R}$, *pro které platí* $\underline{x} \leq \overline{x}$, *pak množinu*

$$\boldsymbol{x} = [\underline{x}, \overline{x}] = \{y \in \mathbb{R}; \underline{x} \leq y \leq \overline{x}\}$$

nazveme reálným intervalem. *Číslo* $\underline{x}$ *nazýváme* dolní mez *a číslo* $\overline{x}$ horní mez.

Pokud platí $\underline{x} = \overline{x}$, potom se intervalu říká *degenerovaný.* Pokud platí $\underline{x} = -\overline{x}$, říkáme, že interval je *symetrický.*

Definice 2. *Symbolem* $\mathbb{IR}$ *označujeme množinu reálných intervalů.*

V následující definici zavedeme důležité pojmy, které jsou úzce spjaty s intervaly.

Definice 3 (Intervalové pojmy)**.** *Nechť* $\boldsymbol{x} = [\underline{x}, \overline{x}] \in \mathbb{IR}$, *definujeme následující pojmy:*

- střed *intervalu* $x^c = \frac{1}{2}(\underline{x} + \overline{x})$,
- poloměr *intervalu* $x^\Delta = \frac{1}{2}(\overline{x} - \underline{x})$,
- šířka *intervalu* $w(\boldsymbol{x}) = \overline{x} - \underline{x}$,
- magnituda *intervalu* $\mathrm{mag}(\boldsymbol{x}) = \max(|\overline{x}|, |\underline{x}|)$.

Předchozí definice nám dává alternativní vyjádření intervalu jako:

$$\boldsymbol{x} = [x^c - x^\Delta, x^c + x^\Delta].$$

1.2 Množinové operace a relace

Definice 4 (Průnik intervalů)**.** *Mějme* $\boldsymbol{x} = [\underline{x}, \overline{x}], \boldsymbol{y} = [\underline{y}, \overline{y}]$ *dvojici reálných intervalů. Průnik* $\boldsymbol{x}, \boldsymbol{y}$ *je prázdný, pokud platí buď* $\overline{y} < \underline{x}$, *nebo* $\overline{x} < \underline{y}$. *V tomto případě je průnik prázdná množina. Píšeme*

$$\boldsymbol{x} \cap \boldsymbol{y} = \emptyset.$$

Jinak můžeme definovat průnik $\boldsymbol{x} \cap \boldsymbol{y}$ *jako interval*

$$\boldsymbol{x} \cap \boldsymbol{y} = \{z \in \mathbb{R}; z \in \boldsymbol{x} \wedge z \in \boldsymbol{y}\} = [\max(\underline{x}, \underline{y}), \min(\overline{x}, \overline{y})].$$

Průnik intervalů odpovídá množinovému průniku, neboť průnik dvou libovolných intervalů je opět interval. Obdobné tvrzení neplatí pro sjednocení intervalů. Množinové sjednocení dvou intervalů nemusí být obecně souvislý interval. Například množinové sjednocení intervalů $[0,1]$ a $[2,3]$ není interval dle definice 1. Proto zavedeme obálku intervalů, pro kterou bude platit, že obálka dvou intervalů je vždy interval.

Definice 5 (Obálka intervalů)**.** *Mějme* $\boldsymbol{x} = [\underline{x}, \overline{x}], \boldsymbol{y} = [\underline{y}, \overline{y}]$ *dvojici reálných intervalů. Obálka intervalů* $\boldsymbol{x}, \boldsymbol{y}$ *je definována jako*

$$\boldsymbol{x} \mathbin{\underline{\cup}} \boldsymbol{y} = [\min(\underline{x}, \underline{y}), \max(\overline{x}, \overline{y})].$$

Pro libovolné dva intervaly $\boldsymbol{x}, \boldsymbol{y}$ platí $\boldsymbol{x} \cup \boldsymbol{y} \subseteq \boldsymbol{x} \mathbin{\underline{\cup}} \boldsymbol{y}$.

Víme, že reálná čísla jsou uspořádaná relací $<$. Tato relace je tranzitivní. Podobná relace může být zavedena i pro intervaly.

Definice 6 (Binární relace)**.** *Nechť* $\boldsymbol{x} = [\underline{x}, \overline{x}], \boldsymbol{y} = [\underline{y}, \overline{y}]$ *jsou reálné intervaly. Platí* $\boldsymbol{x} < \boldsymbol{y}$*, jestliže*

$$\overline{x} < \underline{y}.$$

Platí $\boldsymbol{x} = \boldsymbol{y}$*, jestliže*

$$\underline{x} = \underline{y} \wedge \overline{x} = \overline{y}.$$

Analogicky se zavede relace $\leq$*.*

1.3 Intervalová aritmetika

Na množině intervalů zavedeme základní aritmetické operace, které běžně používáme s reálnými čísly. Intervalové aritmetické operace definujeme tak, aby výsledné intervaly zahrnovaly hodnoty pro všechny možné volby prvků z intervalových operandů.

Definice 7 (Intervalová aritmetika)**.** *Nechť* $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{IR}$*. Intervalová operace* $\circ$ *je definována jako*

$$\boldsymbol{x} \circ \boldsymbol{y} = \{x \circ y; x \in \boldsymbol{x}, y \in \boldsymbol{y}\}.$$

Základní aritmetické operace $+, -, *, /$ můžeme vyjádřit:

- $\boldsymbol{x} + \boldsymbol{y} = [\underline{x} + \underline{y}, \overline{x} + \overline{y}]$,
- $\boldsymbol{x} - \boldsymbol{y} = [\underline{x} - \overline{y}, \overline{x} - \underline{y}]$,
- $\boldsymbol{x} * \boldsymbol{y} = [\min(S), \max(S)]$, kde $S = \{\underline{xy}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{xy}\}$,
- $\boldsymbol{x}/\boldsymbol{y} = [\min(S'), \max(S')]$, kde $S' = \{\underline{x}/\underline{y}, \underline{x}/\overline{y}, \overline{x}/\underline{y}, \overline{x}/\overline{y}\}, 0 \notin \boldsymbol{y}$.

Důležité je, že množina $\mathbb{IR}$ s operacemi, které jsme definovali, netvoří těleso. Přesto pro ni některé axiomy tělesa platí. Existuje, zde nulový prvek $\mathbf{0} = [0{,}0]$ a jednotkový prvek $\mathbf{1} = [1{,}1]$.

Věta 1 (Základní vlastnosti intervalové aritmetiky)**.** *Platí následující vlastnosti.*

- *Intervalové sčítaní a násobení je komutativní a asociativní.*
- *Intervalové sčítaní a násobení není distributivní, ale je subdistributivní, což znamená:*

$$\forall\, \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathbb{IR} : \boldsymbol{x}(\boldsymbol{y} + \boldsymbol{z}) \subseteq \boldsymbol{xy} + \boldsymbol{xz}.$$

Důkaz. Komutativita a asociativita intervalového sčítání a násobení se snadno dokáže z definice intervalové aritmetiky. Pro přehlednost dokážeme pouze komutativitu sčítání:

$$\forall\, \boldsymbol{x}, \boldsymbol{y} \in \mathbb{IR} : \boldsymbol{x} + \boldsymbol{y} = [\underline{x} + \underline{y}, \overline{x} + \overline{y}] = [\underline{y} + \underline{x}, \overline{y} + \overline{x}] = \boldsymbol{y} + \boldsymbol{x}.$$

Dále provedeme důkaz subdistributivity:

$$\begin{aligned} \boldsymbol{x}(\boldsymbol{y} + \boldsymbol{z}) &= \{x(y + z); x \in \boldsymbol{x}, y \in \boldsymbol{y}, z \in \boldsymbol{z}\} \\ &= \{xy + xz); x \in \boldsymbol{x}, y \in \boldsymbol{y}, z \in \boldsymbol{z}\} \\ &\subseteq \{xy + x'z); x, x' \in \boldsymbol{x}, y \in \boldsymbol{y}, z \in \boldsymbol{z}\} \\ &= \boldsymbol{xy} + \boldsymbol{xz}. \end{aligned}$$

□

Příklad. Následující příklad ukazuje, že intervalová aritmetika není obecně distributivní. Distributivitu vyvrátíme následujícími intervaly $\boldsymbol{x} = [1, 2], \boldsymbol{y} = 1$, $\boldsymbol{z} = -1$:

$$\begin{aligned} \boldsymbol{x}(\boldsymbol{y} + \boldsymbol{z}) &= [1, 2] \cdot (1 - 1) = [1, 2] \cdot 0 = 0, \\ \boldsymbol{xy} + \boldsymbol{xz} &= [1, 2] \cdot 1 + [1, 2] \cdot (-1) = [1, 2] - [1, 2] = [-1, 1]. \end{aligned}$$

Axiom existence inverzního a opačné prvku selhává pro intervalovou aritmetiku na množině intervalů.

Poznámka. Pro nedegenerovaný interval neexistuje opačný prvek. Kdyby existoval neutrální prvek $\boldsymbol{y}$ k intervalu $\boldsymbol{x}$, tak by muselo platit

$$[0{,}0] = \boldsymbol{x} + \boldsymbol{y} = [\underline{x} + \underline{y}, \overline{x} + \overline{y}].$$

Což by znamenalo, že $\underline{x} + \underline{y} = 0 \land \overline{x} + \overline{y} = 0$, tedy $\boldsymbol{y}$ by vypadalo následovně $\boldsymbol{y} = [-\underline{x}, -\overline{x}]$. Což obecně není interval. Neboť platí $\underline{x} \leq \overline{x}$, pak ale $-\underline{x} \geq -\overline{x}$. Což je spor s definicí intervalu 1, pokud původní interval nebyl degenerovaný.

2. Intervalové vektory, matice a funkce

Tato část čerpá především z knihy Moore a kol. (2009). Reálné vektory značíme netučnými malými písmeny (x, y, z), reálné matice netučnými písmeny velkými (A, B, C). Intervalové vektory a matice značíme tučnými písmeny $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \text{resp. } \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$.

Absolutní hodnotu reálné matice A budeme chápat jako absolutní hodnotu jednotlivých členů $(|A|_{i,j} = |a_{i,j}|)$.

Poznámka. Pro matice $A, B \in \mathbb{R}^{m\times n}$ zavedeme relaci $\leq$:

$$A \leq B \iff a_{i,j} \leq b_{i,j}, \text{pro } 1 \leq i \leq m, 1 \leq j \leq n.$$

2.1 Intervalové matice

Definice 8 (Intervalová matice I)**.** *Nechť $\underline{A}, \overline{A} \in \mathbb{R}^{m\times n}$, takové že $\underline{A} \leq \overline{A}$. Pak matici*

$$\boldsymbol{A} = [\underline{A}, \overline{A}] = \{A;\ \underline{A} \leq A \leq \overline{A}\}$$

nazveme intervalovou maticí. Maticím $\underline{A}$, $\overline{A}$ říkáme horní (resp. dolní) mez.

Intervalovou matici lze chápat jako matici, která místo členů má intervaly. Další ekvivalentní pohled na intervalové matice je takový, že intervalová matice je množina matic, jejíž koeficienty tvoří všechny možné prvky z intervalových koeficientů dané intervalové matice. Jednotlivé prvky však vybíráme z intervalů nezávisle.

Poznámka. Pokud $\underline{A} = \overline{A}$ dostáváme neintervalovou matici.

V předchozí kapitole jsme uváděli alternativní definici intervalu pomocí středu a poloměru, obdobnou definice uvedeme i zde.

Definice 9 (Intervalová matice II)**.** *Intervalovou matici definujeme*

$$\boldsymbol{A} = [A^c - A^\Delta, A^c + A^\Delta] = \{A;\ |A - A^c| \leq A^\Delta\}.$$

Matici $A^c = \frac{1}{2}(\underline{A} + \overline{A})$ říkáme středová a matici $A^\Delta = \frac{1}{2}(\overline{A} - \underline{A})$ nazveme maticí poloměru.

Poznámka. Obě definice (8 a 9) jsou ekvivalentní a vzájemně převoditelné. Z původní definice 8 jsme odvodili definici 9. Podobně můžeme ze znalosti poloměru a středové matice odvodit horní a dolní mez

$$\underline{A} = A^c - A^\Delta,$$
$$\overline{A} = A^c + A^\Delta.$$

Množinu všech intervalových matic rozměru $m\times n$ značíme $\mathbb{IR}^{m\times n}$. *Intervalový vektor* bude chápat jako intervalovou matici s jedním sloupcem.

Poznámka. Intervalovému vektoru budeme občas říkat *box.*

Definice 10 (Obálka)**.** *Buď $S \subseteq \mathbb{R}^n$. Pak* obálka *S je libovolný intervalový vektor $\boldsymbol{v} \in \mathbb{R}^n : S \subseteq \boldsymbol{v}$. Občas říkáme $\boldsymbol{v}$ zapouzdřuje S.*

Definice 11 (Intervalový obal)**.** *Buď $S \subseteq \mathbb{R}^n$.* Intervalový obal *S je obálka $\boldsymbol{v} \in \mathbb{R}^n$ taková, že pro každou obálku $\boldsymbol{w}$ množiny S platí $\boldsymbol{v} \subseteq \boldsymbol{w}$.* Intervalový obal *značíme $\square S$.*

2.2 Základní poznatky o intervalových maticích

Definice 12. *Buď $\boldsymbol{A} \in \mathbb{IR}^{m\times n}$, $\boldsymbol{b} \in \mathbb{IR}^m$, $y \in \{\pm 1\}^m$, $z \in \{\pm 1\}^n$. Pak definujeme matici*

$$A_{yz} = A^c - \mathrm{diag}(y) A^{\Delta} \mathrm{diag}(z)$$

a vektor

$$b_y = b^c + \mathrm{diag}(y) b^{\Delta},$$

kde

$$\mathrm{diag}(y) = \begin{pmatrix} y_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & y_n \end{pmatrix}.$$

Tvrzení 2. *Buď $\boldsymbol{A} \in \mathbb{IR}^{m\times n}$, pak $A_{yz} \in \boldsymbol{A}$.*

Důkaz. Ukážeme, že $|A_{yz} - A^c| \leq A^{\Delta}$.

$$\begin{aligned} |A_{yz} - A^c| &= |A^c - \mathrm{diag}(y) A^{\Delta} \mathrm{diag}(z) - A^c| \\ &= |\mathrm{diag}(y) A^{\Delta} \mathrm{diag}(z)| \\ &\leq |\mathrm{diag}(y)||A^{\Delta}||\mathrm{diag}(z)| = I_n A^{\Delta} I_n = A^{\Delta} \end{aligned}$$

□

V některých problémech se často objevují symetrické intervalové matice. Tato práce není výjimkou a v pozdějších kapitolách se jim budeme věnovat. Proto je třeba se už teď o nich zmínit.

Definice 13 (Symetrická intervalová matice)**.** *Mějme $\boldsymbol{A} \in \mathbb{IR}^{n\times n}$, pak odpovídající* symetrická intervalová matice *je definována jako $\boldsymbol{A}^S = \{A \in \boldsymbol{A};\ A = A^T\}$.*

2.3 Intervalové funkce

Jednou z nedílných částí intervalové analýzy je umět vyhodnotit funkci na intervalu. Následující definice ukazuje, jak budeme chápat vyhodnocení funkce na daném intervalu.

Definice 14. *Buď* $f : \mathbb{R}^n \to \mathbb{R}, \boldsymbol{x} \in \mathbb{IR}^n$. *Pak*

$$f(\boldsymbol{x}) = \{f(x);\, x \in \boldsymbol{x}\}.$$

Vyhodnotit některé funkce na daném intervalu je jednoduché, například do monotónních spojitých funkcí lze jednoduše dosadit. Pokud je f neklesající na $\boldsymbol{x} \in \mathbb{IR}^n$, potom $f(\boldsymbol{x}) = [f(\underline{x}), f(\overline{x})]$. Pokud je funkce nerostoucí, stačí prohodit krajní meze. Další skupinou funkcí, kde vyhodnocení na daném intervalu není složité, jsou po částech monotónní funkce. Obecně však spočítání mezí není snadné.

Příklad. Zde uvádíme několik funkcí a vyhodnocení funkce na daném intervalu.

- $\exp(\boldsymbol{x}) = [\exp(\underline{x}), \exp(\overline{x})]$,
- $\log(\boldsymbol{x}) = [\log(\underline{x}), \log(\overline{x})]$ pro $\underline{x} \geq 0$,
- $\sin(\boldsymbol{x}) = [\sin(\underline{x}), \sin(\overline{x})]$ pro $\boldsymbol{x} \subseteq [-\frac{\pi}{2}, \frac{\pi}{2}]$,
- $\boldsymbol{x}^2 = \begin{cases} [\min(\underline{x}^2, \overline{x}^2), \max(\underline{x}^2, \overline{x}^2)] & \text{pokud } 0 \notin \boldsymbol{x}, \\ [0, \max(\underline{x}^2, \overline{x}^2)] & \text{jinak.} \end{cases}$

V obecném případě nemusí být $f(\boldsymbol{x})$ interval. Proto se v intervalových metodách snažíme raději určit intervalový obal $\square f(\boldsymbol{x})$. Avšak spočítání $\square f(\boldsymbol{x})$ je obecně těžký problém, pravděpodobně ani neexistuje algoritmus na vyčíslení.

Věta 3. *Problém spočítání* $\square f(\boldsymbol{x})$ *je algoritmicky nerozhodnutelný.*

Důkaz. Využijeme toho, že problém dokázaný Matiyasevichem (Matiyasevich, 1970) je nerozhodnutelný. Matiyasevich dokázal, že zjistit existenci celočíselného kořene polynomu $p(x_1, \dots, x_n)$ je algoritmicky nerozhodnutelný. Definujme charakteristickou funkci $f : \mathbb{R}^n \to \mathbb{R}$ jako

$$f(x) = \begin{cases} 1 & \text{pokud } x \in \mathbb{Z} \text{ a } p(x) = 0, \\ 0 & \text{jinak.} \end{cases}$$

Pak

$$\square f(\boldsymbol{x}) = \begin{cases} [0, 1] & \text{pokud } p(x) \text{ má celočíselný kořen}, \\ 0 & \text{jinak.} \end{cases}$$

Pokud bychom uměli spočítat $\square f(\boldsymbol{x})$, pak bychom uměli zjistit, jestli $p(x)$ má nebo nemá celočíselný kořen.

$\square$

Důsledek. Z důkazu plyne, že algoritmicky nerozhodnutelný problém je i aproximace $\square f(\boldsymbol{x})$ s předem danou absolutní nebo relativní přesností.

To je velice důležitý poznatek pro celou intervalovou analýzu. Vzhledem k tomuto výsledku se nebudeme snažit spočítat $\square f(\boldsymbol{x})$, naopak se pokusíme spočítat co nejtěsnější obálku $f(\boldsymbol{x})$. Avšak spočítání těsné obálky je stále výpočetně těžký problém. Tyto poznatky nás vedou k následující definici a především k Základní větě intervalového počítaní, která nám dá určitý nástroj, jak s tímto problémem naložit. Tato věta je velice důležitá v celém intervalovém počítání, budeme ji nadále často používat. V následující definici zavedeme intervalovou funkci $\boldsymbol{f} : \mathbb{IR}^n \to \mathbb{IR}$. Nejdříve je však nutné definovat vlastnosti, které intervalová funkce bude splňovat.

Definice 15 (Izotonie vzhledem k inkluzi)**.** *Funkce* $\boldsymbol{f} : \mathbb{IR}^n \to \mathbb{IR}$ *je* izotonní vzhledem k inkluzi, *pokud pro všechny* $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{IR}^n$ *platí*

$$\boldsymbol{x} \subseteq \boldsymbol{y} \implies \boldsymbol{f}(\boldsymbol{x}) \subseteq \boldsymbol{f}(\boldsymbol{y}).$$

Definice 16 (Intervalové rozšíření funkce)**.** *Funkce* $\boldsymbol{f} : \mathbb{IR}^n \to \mathbb{IR}$ *je* intervalové rozšíření $f : \mathbb{R}^n \to \mathbb{R}$, *pokud pro všechny* $x \in \mathbb{R}^n$ *platí*

$$f(x) \subseteq \boldsymbol{f}(x).$$

Tyto dvě vlastnosti stačí na to, abychom dostali obálku $f(\boldsymbol{x})$. Tento poznatek formulujeme jako následující větu, poprvé uvedenou v knize Moore (1996).

Věta 4 (Základní věta intervalového počítání)**.** *Je-li* $\boldsymbol{f} : \mathbb{IR}^n \to \mathbb{IR}$ *izotonní vzhledem k inkluzi a je-li intervalovým rozšířením* $f : \mathbb{R}^n \to \mathbb{R}$, *pak pro všechny* $\boldsymbol{x} \in \mathbb{IR}^n$ *platí*

$$f(\boldsymbol{x}) \subseteq \boldsymbol{f}(\boldsymbol{x}).$$

Důkaz. Pro všechny $x \in \boldsymbol{x}$ platí $f(x) = \boldsymbol{f}(x) \subseteq \boldsymbol{f}(\boldsymbol{x})$. Rovnost platí, neboť $\boldsymbol{f} : \mathbb{IR}^n \to \mathbb{IR}$ je intervalové rozšíření $f : \mathbb{R}^n \to \mathbb{R}$. Inkluze naopak plyne z izotonie vzhledem k inkluzi. Dostáváme $\forall x \in \boldsymbol{x} : f(x) \subseteq \boldsymbol{f}(\boldsymbol{x})$, jelikož to platí pro všechny $x \in \boldsymbol{x}$, pak $f(\boldsymbol{x}) \subseteq \boldsymbol{f}(\boldsymbol{x})$.

□

Jednoduše lze nahlédnout, že intervalová aritmetika je izotonní vzhledem k inkluzi a je intervalovým rozšířením. Naopak $\mathrm{mag}(x)$ není izotonní vzhledem k inkluzi ($\mathrm{mag}([0,1]) = 1$, $\mathrm{mag}([0,2]) = 2$).

Definice 17 (Přirozené intervalové rozšíření)**.** *Nechť* $f : \mathbb{R}^n \to \mathbb{R}$ *je funkce, jejíž předpis lze napsat pomocí konečného množství aritmetických operací.* Přirozené intervalové rozšíření $\boldsymbol{f}$ *vznikne nahrazením reálné aritmetiky ve funkci* f *intervalovou aritmetikou.*

Tvrzení 5. *Přirozené intervalové rozšíření je izotonní vzhledem k inkluzi a je to intervalové rozšíření.*

Důkaz. Intervalová aritmetika splňuje obě vlastnosti (izotonii vzhledem k inkluzi a intervalové rozšíření). Dále stačí použít matematickou indukci.

□

Příklad. Buď $f : \mathbb{R}^n \to \mathbb{R}$, $\boldsymbol{x} = [-1, 2]$.

- $f(x) = x^2 - x$
  $\boldsymbol{f}(\boldsymbol{x}) = [0, 4] - [-1, 2] = [-2, 5]$
- $f(x) = x\,(x - 1)$
  $\boldsymbol{f}(\boldsymbol{x}) = [-1, 2] \cdot ([-1, 2] - 1) = [-1, 2] \cdot [-2, 1] = [-4, 2]$
- $f(x) = (x - \frac{1}{2})^2 - \frac{1}{4}$
  $\boldsymbol{f}(\boldsymbol{x}) = ([-1, 2] - \frac{1}{2})^2 - \frac{1}{4} = [-\frac{3}{2}, \frac{3}{2}]^2 - \frac{1}{4} = [0, \frac{9}{4}] - \frac{1}{4} = [-\frac{1}{4}, 2]$

Z předchozího příkladu vidíme, že rozdílné předpisy téže funkce vedou k rozdílným intervalovým rozšířením. Což má za následek rozdílné obálky. Tento příklad nás vede k úvaze, který předpis použít. Nebo zda existuje předpis, který by dával ještě těsnější obálku. Vidíme, že třetí předpis $f(x) = (x - \frac{1}{2})^2 - \frac{1}{4}$ vede k nejtěsnější obálce. Důvod je ten, že ve výrazu se proměnná objevuje jednou. V následujícím tvrzení (Moore, 1996) se na tento problém zaměříme.

Tvrzení 6. *Pokud ve výrazu funkce $f : \mathbb{R}^n \to \mathbb{R}$ se každá proměnná $x_1, \ldots, x_n$ vyskytuje nejvýše jednou, pak pro přirozené intervalové rozšíření $\boldsymbol{f}$ platí* $\forall \boldsymbol{x} \in \mathbb{IR}^n : f(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x})$.

Důkaz. Inkluze "$\subseteq$"přímo plyne ze Základní věty intervalového počítaní (věta 4) a tvrzení 5.
Dále dokážeme opačnou inkluzi "$\supseteq$".
Buď $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \in \mathbb{IR}$ a $\circ$ aritmetická operace. Položme $\boldsymbol{c} = \boldsymbol{a} \circ \boldsymbol{b}$. Pro každé $c \in \boldsymbol{c}$ existují $a \in \boldsymbol{a}, b \in \boldsymbol{b}$, takové že $c = a \circ b$. Dle indukce platí, že pro každé $y \in \boldsymbol{f}(x)$ existuje $x \in \boldsymbol{x}$, takové že $y = f(x)$. Jelikož ve výrazu funkce $f(x)$ se každá proměnná $x_1, \ldots, x_n$ vyskytuje nejvýše jednou, můžeme jednoznačně vybrat $x \in \boldsymbol{x}$. $\square$

3. Soustavy intervalových lineárních rovnic

3.1 Množina řešení

Definice 18 (Soustava intervalových lineárních rovnic)**.** *Nechť* $\boldsymbol{A} \in \mathbb{IR}^{m\times n}$, $\boldsymbol{b} \in \mathbb{IR}^{n}$. *Pak* soustava intervalových lineárních rovnic *je*

$$Ax = b, A \in \boldsymbol{A}, b \in \boldsymbol{b},$$

kterou budeme značit

$$\boldsymbol{A}x = \boldsymbol{b}.$$

Definice 19 (Množina řešení soustavy intervalových lineárních rovnic)**.** *Nechť* $\boldsymbol{A}x = \boldsymbol{b}$ *je soustava lineárních intervalových rovnic.* Množina řešení soustavy intervalových lineárních rovnic *je definována*

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

Soustavě intervalových lineárních rovnic budeme občas říkat intervalový lineární systém. Množinu řešení intervalového systému lze chápat jako sjednocení všech možných řešení všech neintervalových lineárních soustav obsažených v intervalové lineární soustavě. Množina řešení nemusí být konvexní, ale tvoří konvexní množinu v každém ortantu. Následující věta (Oettli a Prager, 1964) charakterizuje množinu řešení intervalového lineárního systému.

Věta 7 (Oettli-Prager)**.** $x \in \Sigma \iff |A^c x - b^c| \le A^\Delta |x| + b^\Delta$

Důkaz. "$\Rightarrow$"Buď $x \in \Sigma$, takže dle definice 19 platí $\exists A \in \boldsymbol{A}, b \in \boldsymbol{b} : Ax = b$.

$$\begin{aligned}|A^c x - b^c| &= |(A^c - A)x + (Ax - b) + b - b^c| = |(A^c - A)x + b - b^c| \\ &\le |(A^c - A)x| + |b - b^c| \le |A^c - A||x| + |b - b^c| \le A^\Delta|x| + b^\Delta\end{aligned}$$

"$\Leftarrow$"Buď $|A^c x - b^c| \le A^\Delta|x| + b^\Delta$. Definuje vektor y

$$y_i = \begin{cases} \frac{(A^c x - b^c)_i}{(A^\Delta|x| + b^\Delta)_i} & \text{pokud } (A^\Delta|x| + b^\Delta)_i > 0, \\ 1 & \text{jinak.} \end{cases}$$

Jednoduše nahlédneme, že platí $y \in [-1, 1]^m$.
Dostáváme $(A^c x - b^c)_i = y_i(A^\Delta|x| + b^\Delta)_i$, což nám dává

$$A^c x - b^c = \mathrm{diag}(y)(A^\Delta|x| + b^\Delta).$$

Označíme $z = \mathrm{sgn}(x)$, pak $|x| = \mathrm{diag}(z)x$. Dostáváme

$$A^c x - b^c = \mathrm{diag}(y)A^\Delta \mathrm{diag}(z)x + \mathrm{diag}(y)b^\Delta,$$

což lze také vyjádřit jako

$$(A^c - \mathrm{diag}(y)A^\Delta\mathrm{diag}(z))x = b^c + \mathrm{diag}(y)b^\Delta.$$

Použitím definice 12 lze rovnici upravit $A_{yz}x = b_y$. Z použitím tvrzení 2 dostáváme $x \in \Sigma$.

$\square$

Množina řešení Σ je nekonvexní polyedr, ale je konvexní v každém ortantu (Beeck, 1973).

Věta 8. *Množina řešení soustavy intervalových lineárních rovnic je v každém ortantu prázdná nebo konvexní polyedr.*

Důkaz. Mějme ortant se znaménkovým vektorem $s \in \{\pm 1\}^n$. Pak tento ortant je popsaný $\mathrm{diag}(s)x \geq 0$. Množina řešení omezená pouze na tento ortant lze popsat jako

$$|A^c x - b^c| \leq A^\Delta |x| + b^\Delta, \mathrm{diag}(s)x \geq 0.$$

Jelikož $|x| = \mathrm{diag}(s)x$, můžeme předchozí rovnici upravit

$$|A^c x - b^c| \leq A^\Delta \mathrm{diag}(s)x + b^\Delta, \mathrm{diag}(s)x \geq 0,$$

dále se šikovně zbavíme absolutní hodnoty, dostáváme

$$A^c x - b^c \leq A^\Delta \mathrm{diag}(s)x + b^\Delta,\ -(A^c x - b^c) \leq A^\Delta \mathrm{diag}(s)x + b^\Delta, \mathrm{diag}(s)x \leq 0.$$

Dále přeskupíme předchozí nerovnice a využijeme definici Intervalové matice II 9

$$(A^c - A^\Delta \mathrm{diag}(s))x \leq \overline{b},\ (-A^c - A^\Delta \mathrm{diag}(s))x \leq -\underline{b}, \mathrm{diag}(s)x \leq 0.$$

$\square$

Důsledek. Pokud $\boldsymbol{A}x = \boldsymbol{b}$, $x \geq 0$, pak množinu řešení lze popsat $\mathrm{diag}(s) = I$

$$\underline{A}x \leq \overline{b},\ \overline{A}x \geq \underline{b}, x \geq 0.$$

3.2 Metody

V této kapitole se zaměříme na metody na nalezení obálky Σ. Omezíme se na čtvercové případy intervalových lineárních systémů. Uvedeme jak přímé tak iterativní metody.

Struktura množiny řešení Σ je velice komplikovaná, proto nás více zajímá, jak vypadá intervalový obal $\square\Sigma$. Avšak jak už bylo zmíněno v předchozí sekci spočítání $\square\Sigma$ je NP-těžký problém. Proto naším cílem bude najít co možná nejtěsnější obálku množiny řešení v polynomiálním čase. Nebo mít nějaký rozumný kompromis mezi časovou složitostí metody a těsností obálky. Kreinovich a Lakeyev (1996) ukázali, že spočítání obálky s předem danou relativní nebo absolutní přesností je stále NP-těžký problém.

Stejně jako u reálných systémů, tak i u intervalových systémů řešitelnost záleží na tom, jestli příslušná matice je regulární.

Definice 20 (Regulární intervalová matice)**.** *Intervalová matice $\boldsymbol{A} \in \mathbb{IR}^{n \times n}$ je* regulární, *pokud každá matice $A \in \boldsymbol{A}$ je regulární.*

Důležitou třídou intervalových matic, které se vyskytují v metodách na nalezení obálky množiny řešení intervalových lineárních systémů, jsou M-matice.

Definice 21 (Reálná M-matice)**.** *$A \in \mathbb{R}^{n \times n}$ je* M-matice *pokud $\forall i \neq j : a_{ij} \leq 0$ a platí jedna z ekvivalencí*

1. $A^{-1} \geq 0$
2. $\exists v > 0 : Av > 0$
3. *Existuje rozklad* $A = M - N$; $M^{-1} \geq 0, N \geq 0, \rho(M^{-1}N) < 1$.

Definice 22 (Intervalová M-matice)**.** *Intervalová matice $\boldsymbol{A} \in \mathbb{IR}^{n \times n}$ je* M-matice *pokud každá matice $A \in \boldsymbol{A}$ je M-matice.*

Věta 9. *$\boldsymbol{A} \in \mathbb{IR}^{n \times n}$ je M-matice pokud $\forall i \neq j : \overline{A}_{ij} \leq 0$ a $\underline{A}$ je M-matice.*

Důkaz. Přímá inkluze je zřejmá, plyne přímo z definice reálné M-matice 21 a intervalové M-matice 22. Jelikož $\underline{A}$ je M-matice, pak $\exists v > 0 : \underline{A}v > 0$, $\forall A \in \boldsymbol{A} : Av \geq \underline{A}v > 0$.

□

3.2.1 Předpodmínění

V metodách na nalezení obálky množiny řešení lineárních intervalových rovnic se často používá předpodmínění původní soustavy. Předpodmínění se používá za účelem nalezení těsnější obálky, díky omezení vlivů intervalových operací. Buď $C \in \mathbb{R}^{n \times n}$ regulární matice, pak předpodmíněním lineárního intervalového systému $\boldsymbol{A}x = \boldsymbol{b}$ rozumíme přenásobení obou stran maticí C, tj.

$$C\boldsymbol{A}x = C\boldsymbol{b}.$$

Množina řešení tohoto nově vzniklého systému obsahuje množinu řešení původního systému, neboť matice C je regulární. Díky této vlastnosti máme jistotu, že obálka předpodmíněného intervalového systému je i obálkou původního intervalového systému. Použitím předpodmínění může dojít k nadhodnocení obálky, avšak mnoho metod v praxi funguje dobře s předpodmíněním.

Nejčastěji volíme $C = (A^c)^{-1}$. V praxi stačí spočítat přibližnou aproximaci inverzní matice $C \approx (A^c)^{-1}$. Rovněž středovou matici můžeme volit jako aproximaci středové matice. Důvod, proč takto volíme C, je, že střed matice $C\boldsymbol{A}$ je jednotková matice. Pokud volíme přibližnou inverzi středové matice, dostáváme matici, která má střed blízko jednotkové matice. Díky tomu operace mimo diagonálu mají malý vliv na nepřesnost v množině řešení.

Nevýhodou předpodmínění je, že musíme spočítat matici C a také to že předpodmínění natahuje množinu řešení.

3.2.2 Intervalová Gaussova eliminace

Idea intervalové Gaussovy eliminace je nahrazení reálných operací intervalovými. Místo, abychom popisovali tuto známou metodu, raději ji ukáže na příkladu.

Příklad. Nalezení obálky pomocí Gaussovy eliminace.

$$(\boldsymbol{A} \quad \boldsymbol{b}) \sim \begin{pmatrix} [2,4] & [-2,1] & [-2,2] \\ [-1,2] & [2,4] & [-2,2] \end{pmatrix} \sim \begin{pmatrix} [2,4] & [-2,1] & [-2,2] \\ 0 & [1,6] & [-4,4] \end{pmatrix}$$

Prvek na pozici $(2,1)$ v poslední matici nemusíme vyhodnocovat intervalovou aritmetikou, neboť víme, že je 0. Zbylé kroky jsme provedli pouze nahrazením reálných operací intervalovými operacemi v klasické Gaussově eliminaci. Dále provedeme zpětnou substituci.

$$\boldsymbol{x}_2 = [-4,4]$$
$$\boldsymbol{x}_1 = \frac{[-2,2] - [-2,1] \cdot [-4,4]}{[2,4]} = [-5,5]$$

Výsledná obálka spočítaná Gaussovou eliminací je $\boldsymbol{x} = ([-5,5],[-4,4])^T$.

3.2.3 Iterační metody

Zde uvedeme metody, které hledají obálku Σ pomocí iterativních kroků. Metody mají společné to, že začnou s $\boldsymbol{x}^0$, po konečně mnoho krocích vydají $\boldsymbol{x}^n$ a platí $\square\Sigma \subseteq \boldsymbol{x}^n$. Popíšeme metody Jacobiho, Gauss-Seidelovu a Krawczykovu metodu, které se postupně snaží zpřesňovat $\boldsymbol{x}^0$. Dále popíšeme metodu ϵ-inflace, která funguje na opačném principu než předchozí metody. Metoda ϵ-inflace postupně zvětšuje počáteční box $\boldsymbol{x}^0$, dokud nevydá obálku množiny řešení. Dále popíšeme, jak volit $\boldsymbol{x}^0$. Naše pozornost neunikne ani tomu, jak nalézt kritérium na zastavení těchto iteračních metod.

Nepočítáme-li ϵ-inflace intervalové iterační metody lze vyjádřit intervalovým operátorem $\mathcal{P} : \mathbb{IR}^n \mapsto \mathbb{IR}^n$, který má vlastnost, že jeho použitím neztrácíme žádné řešení

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

Jedna iterace vypadá následovně $\boldsymbol{x} \mapsto \mathcal{P}(\boldsymbol{x}) \cap \boldsymbol{x}$.

Reálné iterační metody

Stejně jako Gaussova eliminace tak i některé iterativní metody mají základ v reálných iterativních metodách. Pro jednoduchost předpokládáme čtvercový případ soustavy. Nyní odvodíme iterativní metody pro reálné případy. Idea těchto metod je věta o pevném bodě

$$Ax = b \iff x = G(x).$$

Pro dané x^0 se řešení hledá jako limita posloupnosti $x^{k+1} = G(x^k)$.

Richardsonova metoda

$$
\begin{aligned}
x &= x + b - Ax \\
x &= \underbrace{(I - A)}_{B_R} x + b \\
x^{k+1} &= B_R x^k + b
\end{aligned}
$$

Jacobiho metoda

$$
\begin{aligned}
A &= E + D + F, \text{kde} \\
&E \,\text{je ostře dolní trojúhelníková,} \\
&D \,\text{je diagonální,} \\
&F \,\text{je ostře horní trojúhelníková.}
\end{aligned}
$$

$$
\begin{aligned}
Ax = b \iff (E + D + F)x &= b, \\
Dx &= -(E + F)x + b, \\
x &= \underbrace{-D^{-1}(E + F)}_{B_J} x + \underbrace{D^{-1}b}_{f_J}, \\
x^{k+1} &= B_J x^k + f_J.
\end{aligned}
$$

Gaussova-Seidelova metoda

$$
\begin{aligned}
Ax = b \iff (D + E)x + Fx &= b, \\
(D + E)x &= -Fx + b, \\
x &= \underbrace{-(D + E)^{-1}F}_{B_{GS}} x + \underbrace{(D + E)^{-1}b}_{f_{GS}}, \\
x^{k+1} &= B_{GS} x^k + f_{GS}.
\end{aligned}
$$

Vyčíslení inverzní matice k $D + E$ se vyhneme následovně

$$
\begin{aligned}
x^{k+1} &= -(D + E)^{-1}Fx^k + (D + E)^{-1}b, \\
(D + E)x^{k+1} &= -Fx^k + b, \\
Dx^{k+1} &= -Ex^{k+1} - Fx^k + b, \\
x^{k+1} &= -D^{-1}Ex^{k+1} - D^{-1}Fx^k + D^{-1}b.
\end{aligned}
$$

Popsané metody se v praktické implementaci používají ve tvaru rozepsaném po složkách bez použití inverzní matice.

Jacobiho metoda

$$
\begin{aligned}
\sum_{j=1}^{n} a_{ij}x_j &= b_i \\
a_{ii}x_i + \sum_{j\neq i} a_{ij}x_j &= b_i
\end{aligned}
$$

$$
x_i^{k+1} = \frac{1}{a_{ii}}\Bigg(- \sum_{j=1}^{i-1} a_{ij}x_j^k - \sum_{j=i+1}^{n} a_{ij}x_j^k \Bigg) + \frac{b_i}{a_{ii}}
$$

Gaussova-Seidelova metoda

$$b_i = \sum_{j=1}^{n} a_{ij} x_j$$

$$b_i = \sum_{j=1}^{i-1} a_{ij} x_j + a_{ii} x_i + \sum_{j=i+1}^{n} a_{ij} x_j$$

$$a_{ii} x_i = -\sum_{j=1}^{i-1} a_{ij} x_j - \sum_{j=i+1}^{n} a_{ij} x_j + b_i$$

$$x_i^{k+1} = \frac{1}{a_{ii}} \left(-\sum_{j=1}^{i-1} a_{ij} x_j^{k+1} - \sum_{j=i+1}^{n} a_{ij} x_j^k \right) + \frac{b_i}{a_{ii}}$$

Intervalová Jacobiho a Gauss-Seidelova metoda

Nyní popíšeme intervalové verze předchozích metod.

Mějme intervalovou lineární čtvercovou soustavu $\boldsymbol{A}x = \boldsymbol{b}$. Abychom dostali formuli pro iterativní zlepšování jedné proměnné, stačí nahradit původní reálné aritmetické operace intervalovými operacemi.

Intervalová Jacobiho metoda

$$\boldsymbol{x}_i^* = \frac{1}{\boldsymbol{a}_{ii}} \left(-\sum_{j=1}^{i-1} \boldsymbol{a}_{ij} \boldsymbol{x}_j^k - \sum_{j=i+1}^{n} \boldsymbol{a}_{ij} \boldsymbol{x}_j^k + \boldsymbol{b}_i \right) \tag{3.1}$$

Pro spočítání $\boldsymbol{x}_i^{k+1}$ uděláme průnik se starým řešení

$$\boldsymbol{x}_i^{k+1} = \boldsymbol{x}_i^k \cap \boldsymbol{x}_i^*. \tag{3.2}$$

Nyní si popíšeme algoritmus intervalové Jacobiho metody.
Mějme intervalový lineární systém $\boldsymbol{A}x = \boldsymbol{b}$. Nejdříve musíme spočítat počáteční obálku $\boldsymbol{x}^0 \supseteq \Sigma$. Dále k-tá iterace Jacobiho metody je

for $i := 1, \ldots, n$ **do**
  $\boldsymbol{x}_i^k = \frac{1}{\boldsymbol{a}_{ii}} \left(\boldsymbol{b}_i - \sum_{j=1}^{i-1} \boldsymbol{a}_{ij} \boldsymbol{x}_j^{k-1} - \sum_{j=i+1}^{n} \boldsymbol{a}_{ij} \boldsymbol{x}_j^{k-1} \right);$
  $\boldsymbol{x}_i^k = \boldsymbol{x}_i^k \cap \boldsymbol{x}_i^{k-1};$
end for.

Výhodou Jacobiho metody je, že jednotlivé $\boldsymbol{x}_i^{k-1}$ lze počítat na základě starých hodnot paralelně.

Intervalová Gauss-Seidlova metoda

$$\boldsymbol{x}_i^* = \frac{1}{\boldsymbol{a}_{ii}} \left(-\sum_{j=1}^{i-1} \boldsymbol{a}_{ij} \boldsymbol{x}_j^{k+1} - \sum_{j=i+1}^{n} \boldsymbol{a}_{ij} \boldsymbol{x}_j^k + \boldsymbol{b}_i \right) \tag{3.3}$$

Pro spočítání $\boldsymbol{x}_i^{k+1}$ opět stačí udělat průnik se starým řešení

$$\boldsymbol{x}_i^{k+1} = \boldsymbol{x}_i^k \cap \boldsymbol{x}_i^*.$$

Nyní si uvedeme algoritmus intervalové Gauss-Seidlovy metody.
Mějme intervalový lineární systém $\boldsymbol{A}x = \boldsymbol{b}$. Stejně jako u Jacobiho metody nejdříve musíme spočítat počáteční obálku $\boldsymbol{x}^0 \supseteq \Sigma$. Dále k-tá iterace Gauss-Seidelovy metody je

for $i := 1, \dots, n$ **do**

$\boldsymbol{x}_i^k = \frac{1}{\boldsymbol{a}_{ii}}\left(\boldsymbol{b}_i - \sum_{j=1}^{i-1} \boldsymbol{a}_{ij}\boldsymbol{x}_j^k - \sum_{j=i+1}^{n} \boldsymbol{a}_{ij}\boldsymbol{x}_j^{k-1}\right);$

$\boldsymbol{x}_i^k = \boldsymbol{x}_i^k \cap \boldsymbol{x}_i^{k-1};$

end for.

Gauss-Seidelova metoda je určitým vylepšením Jacobiho metody. Výhodou této metody je, že nově spočítané koeficienty vektoru obálky počítáme rovnou. Přičemž u Jacobiho metody čekáme až na další iteraci. Nevýhodou Gauss-Seidelovy metody ale je, že ji nelze paralyzovat.

Pokud bychom v nějaké iteraci Jacobiho metody (3.2) dostali prázdný průnik, znamená to, že systém nemá řešení. Nebo se také může stát, že jsme nepočítali $\boldsymbol{x}^0$, ale odhadli jsme ho nějakou metodou. Pak prázdný průnik může znamenat, že jsme odhad určili špatně. Totéž platí o Gauss-Seidelově metodě.

Jacobiho a Gauss-Seidelova metoda konverguje ke stejné limitě (Neumaier, 1990, Thm. 4.4.10).

Důležité je ukázat, že použitím Jacobiho či Gauss-Seidelovy metody neztratíme žádné řešení. Zaměříme se pouze na Jacobiho metodu, u Gauss-Seidelovy metody je to podobné. Opět rozložíme $\boldsymbol{A} = \boldsymbol{E} + \boldsymbol{D} + \boldsymbol{F}$ na ostře dolní trojúhelníkovou $\boldsymbol{E}$, diagonální $\boldsymbol{D}$ a ostře horní trojúhelníkovou $\boldsymbol{F}$. Využijeme maticového tvaru Jacobiho metody. Pak definujeme Jacobiho operátor

$$J(\boldsymbol{x}) = \boldsymbol{D}^{-1}\big(\boldsymbol{b} - (\boldsymbol{E} + \boldsymbol{F})\boldsymbol{x}\big).$$

Věta 10. $\boldsymbol{x} \cap \Sigma \subseteq J(\boldsymbol{x})$

Důkaz. Pro každé řešení platí $Ax = b$, pro určité $A \in \boldsymbol{A}$, $b \in \boldsymbol{b}$. Z maticového tvaru vyjádříme $x = D^{-1}\big(\ b - (E + \ F)x\big)$. Což nám dává

$$x \subseteq \boldsymbol{D}^{-1}\big(\boldsymbol{b} - (\boldsymbol{E} + \boldsymbol{F})\boldsymbol{x}\big) = J(\boldsymbol{x}).$$

□

Obdobné tvrzení platí i o Gauss-Seidelově operátoru $GS(\boldsymbol{x})$.

Pokud $\boldsymbol{A}$ je M-matice, pak intervalová Gauss-Seidelova metoda konverguje k $\square\Sigma$. Dokázáno ve článku Barth a Nuding (1974).

Problém těchto metod, je pokud $0 \in \boldsymbol{a}_{ii}$, neboť ve výrazech (3.2) a (3.3) se vyskytuje $1/\boldsymbol{a}_{ii}$. Tento problém lze vyřešit přerovnáním rovnic tak, aby diagonální prvky matice $\boldsymbol{A}$ neobsahovali 0. Nebo můžeme použít zobecněnou intervalovou aritmetiku.

Poznámka. Zobecněná intervalová aritmetika:

$$\frac{\boldsymbol{a}}{\boldsymbol{b}} = \left\{\frac{a}{b};\, a \in \boldsymbol{a}, b \in \boldsymbol{b}, a \neq 0\right\}$$

Krawczykova metoda

Krawczykova metoda vychází z Richardsonovy metody pro reálné lineární systémy. Definujme Krawczykův operátor

$$K(\boldsymbol{x}) = \boldsymbol{b} + (I_n - \boldsymbol{A})\boldsymbol{x}.$$

V následující větě ukážeme, že použitím Krawczykova operátoru neztrácíme žádné řešení.

Věta 11. $\boldsymbol{x} \cap \Sigma \subseteq K(\boldsymbol{x})$

Důkaz. Buď $x \in \boldsymbol{x} \cap \Sigma$, pak $\exists A \in \boldsymbol{A}, b \in \boldsymbol{b} : Ax = b$.
Jelikož $Ax = b$, dostáváme $x = b - Ax + x = b + (I - A)x \subseteq \boldsymbol{b} + (I_n - \boldsymbol{A})\boldsymbol{x} = K(\boldsymbol{x})$. □

U Krawczykovy metody můžeme určit, jak moc přehodnocujeme intervalový obal po jedné iteraci.

Věta 12. *Buď $\Sigma \subseteq \boldsymbol{x}$. Pak po jedné iteraci Krawczykova operátoru nepřehodnocujeme $\square\Sigma$ o více než $2\,\mathrm{mag}(I - \boldsymbol{A})x^{\Delta}$.*

Důkaz. Označme

$$\alpha = \boldsymbol{b} + (I - \boldsymbol{A})x^c, \beta = \mathrm{mag}(I - \boldsymbol{A})x^{\Delta}.$$

Buď $x \in \Sigma$ to znamená, že $\exists A \in \boldsymbol{A}, b \in \boldsymbol{b} : Ax = b$,
$x = b + (I - A)x = b + (I - A)(x^c - x^c + x) = b + (I - A)x^c + (I - A)(x - x^c)$.
Pak

$$\overline{\square\Sigma} \leq \overline{\underbrace{\boldsymbol{b} + (I - \boldsymbol{A})x^c}_{\alpha} + \underbrace{(I - \boldsymbol{A})(x - x^c)}_{\leq \mathrm{mag}(I - \boldsymbol{x})x^{\Delta} = \beta}} \leq \alpha + \beta.$$

Obdobně vyjádříme

$$\begin{aligned}\overline{\square\Sigma} &\geq b + (I - A)x^c + (I - A)(x - x^c) \\ &\geq \underbrace{\overline{\boldsymbol{b} + (I - \boldsymbol{A})x^c}}_{\alpha} - \underbrace{\max_{A \in \boldsymbol{A},\, b \in \boldsymbol{b}} (I - A)(x - x^c)}_{\mathrm{mag}(I - \boldsymbol{x})x^{\Delta} = \beta} = \alpha - \beta.\end{aligned}$$

Dostáváme

$$\alpha - \beta \leq \overline{\square\Sigma} \leq \overline{\boldsymbol{b} + (I - \boldsymbol{A})\boldsymbol{x}} \leq \alpha + \beta,$$

pak nadhodnocení je nejvýše 2β.

□

Někdy se Krawczykův operátor uvádí v předpodmíněném tvaru

$$K(\boldsymbol{x}) = C\boldsymbol{b} + (I_n - C\boldsymbol{A})\boldsymbol{x},$$

kde $C \in \mathbb{R}^{n \times n}$ je volena jako $C \approx (A^c)^{-1}$. Tento tvar je využíván v praktických implementacích, neboť častěji vede k těsnější obálce.

Pokud použijeme předpodmínění v Gauss-Seidlově metodě, pak se dá ukázat (Neumaier, 1990, Thm. 4.3.5), že platí $K(\boldsymbol{x}) \subseteq GS(\boldsymbol{x})$.

Určení počáteční obálky

Předchozí iterační metody fungovaly na principu zlepšování počáteční obálky $\boldsymbol{x}^0$. Zde ukážeme, jak volit $\boldsymbol{x}^0$. Existuje několik způsobů, jak volit počáteční obálku $\boldsymbol{x}^0$. Jednoduchou možností je vzít obálku spočítanou jinou metodou a dát ji na vstup iterační metodě. Další jednoduchou možností je vyřešit středovou soustavu $A^c x = b^c$ a výsledek obalit intervalem. Problém nastává s volbou poloměru intervalu. Následující věta dává explicitní vzorec na výpočet počáteční obálky.

Věta 13. *Buď* $C \in \mathbb{R}^{n\times n} : \|I - C\boldsymbol{A}\|_\infty < 1$, *pak*

$$\Sigma \subseteq \frac{\|C\boldsymbol{b}\|_\infty}{1 - \|I - C\boldsymbol{A}\|_\infty}[-1, 1]^n.$$

Poznámka. Intervalová maticová norma je definována

$$\|\boldsymbol{A}\|_\infty = \max_i \sum_j \operatorname{mag}(\boldsymbol{a}_{ij}).$$

Důkaz. Buď $x \in \Sigma$, což znamená $Ax = b$ pro nějaké $A \in \boldsymbol{A}, b \in \boldsymbol{b}$, tedy $CAx = Cb$. Vyjádříme

$$x = (I - CA)x + CAx = (I - CA)x + Cb,$$

pak pro všechny maticové normy indukované vektorovou normou platí

$$\|x\| = \|(I - CA)x + Cb\| \le \|(I - CA)x\| + \|Cb\| \le \|I - CA\|\|x\| + \|Cb\|.$$

Předchozí nerovnost ekvivalentně upravíme $\big(1 - \|I - CA\|\big)\|x\| \le \|Cb\|$. Pro maticovou normu $\|\cdot\|_\infty$ dostáváme

$$\|x\|_\infty \le \frac{\|Cb\|_\infty}{1 - \|I - CA\|_\infty} \le \frac{\|C\boldsymbol{b}\|_\infty}{1 - \|I - C\boldsymbol{A}\|_\infty}.$$

□

ϵ-inflace

Předchozí iterační metody fungovaly na stejném principu. Začaly s nějakou obálkou Σ a iterativně ji ztenčovaly. Naopak metoda ϵ-inflace postupuje opačným směrem. Začne s malým intervalovým vektorem a postupně ho zvětšuje, dokud není obálkou Σ.

Buď $\mathcal{O}_{\boldsymbol{A},\boldsymbol{b}}(\boldsymbol{x}) : \mathbb{IR}^n \mapsto \mathbb{IR}^n$ operátor splňující

- $\mathcal{O}_{A,b}(x)$ je spojitý vzhledem k x pro každé $A \in \boldsymbol{A}, b \in \boldsymbol{b}$,
- $\mathcal{O}_{A,b}(x) = x$ pro každé $A \in \boldsymbol{A}, b \in \boldsymbol{b}$, $x \in \mathbb{R}^n$ takové, že $Ax = b$.

Příklad. Krawczykův operátor splňuje vlastnosti operátoru $\mathcal{O}_{\boldsymbol{A},\boldsymbol{b}}(\boldsymbol{x})$.

Věta 14. *Je-li* $\mathcal{O}_{\boldsymbol{A},\boldsymbol{b}}(\boldsymbol{x}) \subseteq \boldsymbol{x}$, *pak* $\forall A \in \boldsymbol{A}, b \in \boldsymbol{b}, \exists x \in \boldsymbol{x} : Ax = b$.

Důkaz. Pro každé $A \in \boldsymbol{A}, b \in \boldsymbol{b}$ platí $\mathcal{O}_{A,b}(\boldsymbol{x}) \subseteq \boldsymbol{x}$. Podle věty o pevném bodě $\exists x \in \boldsymbol{x}$ takové, že $\mathcal{O}_{A,b}(x) = x$. Pak x je řešení $Ax = b$.

□

Připomeňme větu o pevném bodě. Mějme konvexní a kompaktní množinu $\mathcal{M} \subseteq \mathbb{R}^n$ a spojité zobrazení $f : \mathcal{M} \mapsto \mathcal{M}$. Pak existuje $x \in \mathcal{M}$ takové, že $f(x) = x$. Avšak podmínka $\mathcal{O}_{\boldsymbol{A},\boldsymbol{b}}(\boldsymbol{x}) \subseteq \boldsymbol{x}$ není postačující proto, aby platilo $\Sigma \subseteq \boldsymbol{x}$ a $\boldsymbol{A}$ je regulární. Následující příklad ilustruje toto tvrzení.

Příklad. Nechť

$$\boldsymbol{A} = \begin{pmatrix} [0,2] & 0 \\ 0 & [0,2] \end{pmatrix}, \boldsymbol{b} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \boldsymbol{x} = \begin{pmatrix} [-1,1] \\ [-1,1] \end{pmatrix}.$$

Pro Krawczykův operátor dostáváme

$$K(\boldsymbol{x}) = 0 + \begin{pmatrix} [-1,1] & 0 \\ 0 & [-1,1] \end{pmatrix} \cdot \begin{pmatrix} [-1,1] \\ [-1,1] \end{pmatrix} = \begin{pmatrix} [-1,1] \\ [-1,1] \end{pmatrix} = \boldsymbol{x},$$

ale $\Sigma = \mathbb{R}^n$ a $\boldsymbol{A}$ je singulární.

Potřebujeme tedy silnější poznatky o Krawczykově operátoru.

Věta 15. *Buď $\boldsymbol{x}, \boldsymbol{v} \in \mathbb{IR}^n, \boldsymbol{V} \in \mathbb{IR}^{n\times n}$. Je-li $\boldsymbol{v} + \boldsymbol{V}\boldsymbol{x} \subseteq \operatorname{int} \boldsymbol{x}$, pak $\rho(\boldsymbol{V}) < 1$.*

Poznámka. int $\boldsymbol{x}$ označuje vnitřek intervalu (z anglického interior).

Důkaz. Buď $V \in \boldsymbol{V}$, pak můžeme přepsat

$$\boldsymbol{v} + V\boldsymbol{x} = \boldsymbol{v} + V[x^c - x^\Delta, x^c + x^\Delta] = \boldsymbol{v} + Vx^c + [-|V|x^\Delta, |V|x^\Delta] \subseteq \operatorname{int} \boldsymbol{x}.$$

Jelikož $|V|x^\Delta < x^\Delta$, pak podle Perronovy teorie nezáporných matic platí $\rho(V) \leq \rho(|V|) < 1$. Když každá matice $V \in \boldsymbol{V}$ má $\rho(V) < 1$, pak $\rho(\boldsymbol{V}) < 1$.

□

Věta 16. *Je-li $K(\boldsymbol{x}) \subseteq \operatorname{int} \boldsymbol{x}$, pak $\Sigma \subseteq \boldsymbol{x}$ a $\boldsymbol{A}$ je regulární.*

Důkaz. Podle předchozí věty 14 je $\rho(I - \boldsymbol{A}) < 1$. To znamená, že pro každou matici $A \in \boldsymbol{A}$ je $\rho(I - A) < 1$, což implikuje regularitu $\boldsymbol{A}$. Použitím téže věty 14 dostáváme $\Sigma \subseteq \boldsymbol{x}$.

□

Tato věta nám dává možnost formulovat algoritmus ϵ-inflace.
Nejdříve zvolme počáteční vektor $\boldsymbol{x}^0 = (\boldsymbol{b} - \boldsymbol{A}x^*)$, kde x^* je řešení $A^c x = b^c$ a pak iterujeme.
k-tá iterace:

$\boldsymbol{x}_k = \boldsymbol{x}^{k-1}[1 - \epsilon,\ 1 + \epsilon]$;
if $K(\boldsymbol{x}^k) \subseteq \operatorname{int} \boldsymbol{x}^k$ **then**
  konec
end if

Konstanta $\epsilon > 0$ je inflační konstanta, pomocí které postupně zvětšujeme původní vektor, dokud nemáme jistotu, že aktuální vektor je obálkou množiny řešení. Výhodou této metody je rychlost, neboť stačí malý počet iterací.

Poznámka. Metoda ϵ-inflace je implementována v Intlabu jako funkce **verifylss**. V Intlabu (Rump, 2010) je tato metoda trochu pozměněná, používá se předpodmínění $C \approx A^{-1}$ a také reziduální tvar Krawczykova operátoru. Inflační konstanta je volena $\epsilon = 0.1$. V Intalabu je maximální počet iterací omezen na 15.

3.2.4 Metoda Hansen-Bliek-Rohn

Hansen-Bliek-Rohnova metoda je přímá metoda na nalezení zapouzdření množiny všech řešení intervalového lineárního systému. Pokud $A^c = I_n$, metoda nalezne intervalový obal $\Box\Sigma$. Což není zas tak moc omezující podmínka, neboť můžeme předpodmínit původní soustavu. Jestliže tak učiníme, pak metoda dává intervalový obal předpodmíněného systému, což stále dává těsnou obálku množiny řešení původního systému. Metoda využívá předpodmínění implicitně. Existují další vylepšení této metody jako například Ning a Kearfott 1997. Pro praktické implementace je jednodušší použít upravenou metodu Ning-Kearfott. Ukážeme původní metodu Hansen-Bliek-Rohn, která je uvedena v pracích Hansen (1992), Bliek (1992) a dokázána v práci Rohn (1993).

Věta 17 (Hansen-Bliek-Rohn, 1993)**.** *Buď A^c regulární a $\rho(|(A^c)^{-1}|A^\Delta) < 1$. Označme*

$$\begin{aligned} x^c &= (A^c)^{-1}b^c, \\ M &= \big(I_n - |(A^c)^{-1}|A^\Delta\big)^{-1}, \\ x^* &= M\big(|x^c| + |(A^c)^{-1}|b^\Delta\big). \end{aligned}$$

Pak pro každé $i = 1,\dots,n$ platí

$$\overline{\Box\Sigma} \leq \max\Big\{x_i^* + M_{ii}(x_i^c - |x_i^c|), \frac{1}{2M_{ii}-1}\big(x_i^* + M_{ii}(x_i^c - |x_i^c|)\big)\Big\} \tag{3.4}$$

a

$$\underline{\Box\Sigma} \geq \min\Big\{ - x_i^* + M_{ii}(x_i^c + |x_i^c|), \frac{1}{2M_{ii}-1}\big(- x_i^* + M_{ii}(x_i^c + |x_i^c|)\big)\Big\}. \tag{3.5}$$

Důkaz. Nejdříve dokážeme odhad pro horní mez (3.4), odhad na dolní mez (3.5) se dokáže převedením na (3.4).
Použijeme Oettli-Pragerovu větu 7. Pro každé $x \in \Sigma$ platí

$$|A^c x - b^c| \leq A^\Delta|x| + b^\Delta. \tag{3.6}$$

Dále budeme tuto nerovnici (3.6) upravovat

$$\begin{aligned} |(A^c)^{-1}||A^c x - b^c| &\leq |(A^c)^{-1}|(A^\Delta|x| + b^\Delta), \\ |(A^c)^{-1}(A^c x - b^c)| &\leq |(A^c)^{-1}|(A^\Delta|x| + b^\Delta), \\ |(A^c)^{-1}A^c x - (A^c)^{-1}b^c| &\leq |(A^c)^{-1}|(A^\Delta|x| + b^\Delta), \\ |x - x^c| &\leq |(A^c)^{-1}|(A^\Delta|x| + b^\Delta). \end{aligned}$$

Z čehož dostáváme dvě nerovnice

$$x - x^c \leq |(A^c)^{-1}|(A^\Delta|x| + b^\Delta) \tag{3.7}$$

a

$$|x| - |x^c| \leq |(A^c)^{-1}|(A^\Delta|x| + b^\Delta). \tag{3.8}$$

Sestavíme i-tou nerovnost z nerovnice (3.7) a se zbytkem nerovnice (3.8), dostáváme

$$|x| + (x_i - |x|_i)e_i - |x^c| - (x_i^c - |x^c|_i)e_i) \leq |(A^c)^{-1}|(A^\Delta|x| + b^\Delta),$$

členy s $|x|$ převedeme na jednu stranu a členy s $|x^c|$ na druhou stranu nerovnice.

$$(I_n - |(A^c)^{-1}|A^\Delta)|x| + (x_i - |x|_i)e_i \leq |x^c| + (x_i^c - |x^c|_i)e_i + |(A^c)^{-1}|b^\Delta. \quad (3.9)$$

Podle Neumannových řad lze matici M vyjádřit jako

$$M = \left(I_n - |(A^c)^{-1}|A^\Delta\right)^{-1} = \sum_{k=0}^{\infty}(|(A^c)^{-1}|A^\Delta)^k \geq I_n \geq 0.$$

Předchozí nerovnost (3.9) přenásobíme zleva M a dostaneme

$$|x| + (x_i - |x|_i)Me_i \leq (x_i^c - |x^c|_i)Me_i + x^*.$$

Vyjádříme i-tou nerovnici

$$|x|_i + (x_i - |x|_i)M_{ii} \leq (x_i^c - |x^c|_i)M_{ii} + x_i^*.$$

Rozlišíme dva případy

1. $x_i \geq 0$
$$x_i \leq (x_i^c - |x^c|_i)M_{ii} + x_i^*, \quad (3.10)$$

2. $x_i < 0$
$$(2M_{ii} - 1)x_i \leq (x_i^c - |x^c|_i)M_{ii} + x_i^*.$$
Jelikož $M \geq I_n$, pak $2M_{ii} > 1$ a díky tomu výraz můžeme upravit
$$x_i \leq \frac{(2M_{ii} - 1)}{(x_i^c - |x^c|_i)M_{ii} + x_i^*}. \quad (3.11)$$

Z přechozích nerovnosti (3.10) a (3.11) vezmeme maximum a dostáváme (3.4).

Dolní odhad (3.5) převedeme na předchozí případ nahrazením $\boldsymbol{A}(-x) = -\boldsymbol{b}$. Pro tento případ x^c má opačné znaménko. Pro všechny $x \in \Sigma$ dostáváme

$$-x_i \leq \max\left\{x_i^* + M_{ii}(-x_i^c - |x_i^c|), \frac{1}{2M_{ii} - 1}\left(x_i^* + M_{ii}(-x_i^c - |x_i^c|)\right)\right\},$$

což dokazuje (3.5).

$\square$

Poznámka. Připomeňme, co říkají Neumannovy řady.
Pro $A \in \mathbb{R}^{n\times n}$ platí $\rho(A) < 1 \iff (I - A)^{-1} = \sum_{k=0}^{\infty} A^k$.
Poznámka. Pokud $A^c = I_n$, Hansen-Bliek-Rohnova metoda nalezne přesně intervalový obal množiny řešení $\square\Sigma$.

Jak jsme již zmínili, existují další alternativní formule vycházející z Hansen-Bliek-Rohnovy formule. Ukážeme formuli Ning a Kearfott (1997). Označme

$$u = \langle \boldsymbol{A} \rangle^{-1}\mathrm{mag}(\boldsymbol{b}),$$
$$d_i = (\langle \boldsymbol{A} \rangle^{-1})_{ii}, \text{ pro } i = 1,\ldots,n,$$
$$\alpha_i = \langle \boldsymbol{a}_{ii} \rangle - \frac{1}{d_i}, \text{ pro } i = 1,\ldots,n,$$

kde $\langle \boldsymbol{A} \rangle = I_n - A^\Delta$.
Pokud $A^c = I_n$, pak Ning-Kearfottova formule explicitně dává intervalový obal

$$\square\Sigma_i = \frac{\boldsymbol{b}_i + (u_i/d_i - \mathrm{mag}(\boldsymbol{b}_i))[-1,1]}{\boldsymbol{a}_{ii} + \alpha_i[-1,1]}, \text{ pro } i = 1,\ldots,n.$$

3.2.5 Porovnání metod

Předpokládejme $A^c = I_n$, což není příliš velké omezení, neboť můžeme původní systém předpodmínit $(A^c)^{-1}$. Označme $\boldsymbol{x}^K, \boldsymbol{x}^{GS}, \boldsymbol{x}^G, \boldsymbol{x}^{HBR}$ limitní obálku spočítanou Krawczykovou a Gauss-Seidelovou metodou, obálku nalezenou Gaussovou eliminací a Hansen-Bliek-Rohnovou metodou.

Tvrzení 18. *Platí*

$$\square\Sigma = \boldsymbol{x}^{HBR} \subseteq \boldsymbol{x}^G \subseteq \boldsymbol{x}^{GS} \subseteq \boldsymbol{x}^K.$$

Důkaz. Porovnání s Gaussovou eliminací je ukázáno v knize Neumaier (1990, Thm. 4.5.11). Ostatní inkluze jsme už ukázali nebo zmínili v předchozích sekcích.

□

3.2.6 Nalezení intervalového obalu

Předchozí metody vždy nalezly obálku množiny řešení systému $\boldsymbol{A}x = \boldsymbol{b}$. Zde se zmíníme o metodách, které (až na numerické chyby při vyhodnocování na počítači) najdou intervalový obal množiny řešení. Tyto algoritmy mají v nejhorším případě exponenciální časovou složitost. Avšak nemusí být exponenciální ve všech případech.

První algoritmus se opírá o větu 8, která říká, že množina řešení je v každém ortantu prázdná nebo konvexní polyedr. Postupně projdeme všechny ortanty a stanovíme meze dané množiny, která je tvořena konvexním polyedrem (pokud není prázdná) a výstupem budou maximální hranice. Celkem musíme projít 2^n ortantů. Na tuto metodu můžeme použít lineární programování. Abychom dostali podmínky lineárního programu, upravíme Oettli-Pragerovu větu 7. Mějme dán ortant $s \in \{\pm 1\}^n$, pak $\mathrm{diag}(s)x$ určuje daný ortant. Upravíme Oettli-Pragerovu podmínku

$$|A^c x - b^c| \leq A^\Delta |x| + b^\Delta$$

pro daný ortant s, kde pro každé řešení v tomto ortantu platí $|x| = \mathrm{diag}(s)x$, pak dostáváme

$$|A^c x - b^c| \leq A^\Delta \mathrm{diag}(s)x + b^\Delta.$$

Následně se zbavíme absolutní hodnoty

$$\begin{aligned} A^c x - b^c &\leq A^\Delta \mathrm{diag}(s)x + b^\Delta, \\ -A^c x + b^c &\leq A^\Delta \mathrm{diag}(s)x + b^\Delta. \end{aligned}$$

Poté stačí dát prvky obsahující x na jednu stranu nerovnice. Lineární program, jehož vyřešením nalezneme $\square\Sigma$ v ortantu s, je

pro každé $i = 1, \ldots, n$

$$\begin{aligned} \underline{\square\Sigma}_i^s &= \min x_i, \\ \overline{\square\Sigma}_i^s &= \max x_i, \end{aligned}$$

za podmínek

$$
\begin{aligned}
0 &\le \operatorname{diag}(s)x, \\
\underline{b} &\le \big(A^c + A^\Delta \operatorname{diag}(s)\big)x, \\
\overline{b} &\ge \big(A^c - A^\Delta \operatorname{diag}(s)\big)x.
\end{aligned}
$$

Celkem máme $2n$ lineárních programů pro každý z 2^n ortantů.

Tuto metodu lze vylepšit nalezením obálky $\boldsymbol{x} \subseteq \square\Sigma$. Následně jenom procházíme ortanty, který mají neprázdný průnik s $\boldsymbol{x}$.

Vylepšením těchto metod je metoda Jansson (1997), která je založena na topologických vlastnostech množiny řešení.

4. Metoda nejmenších čtverců

V této kapitole se budeme zabývat řešením intervalových soustav metodou nejmenších čtverců. Uvedeme metody na zapouzdření řešení přeurčených intervalových soustav lineárních rovnic metodou nejmenších čtverců. Dále ukážeme metody na zapouzdření řešení symetrických intervalových soustav lineárních rovnic. Neboť jak později ukážeme, problém zapouzdření řešení intervalových soustav lineárních rovnic metodou nejmenších čtverců lze redukovat na zapouzdření řešení symetrických intervalových soustav lineárních rovnic.

Nejdříve připomeneme, jak vypadá metoda nejmenších čtverců pro reálné soustavy. Buď $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$, pak $x \in \mathbb{R}^n$ je řešení metodou nejmenších čtverců soustavy $Ax = b$ (typicky $m \gg n$), pokud platí

$$\|Ax - b\|_2 = \min_{y \in \mathbb{R}^n} \|Ay - b\|_2.$$

Věta 19 (Metoda nejmenších čtverců)**.** *Buď $A \in \mathbb{R}^{m \times n}$ hodnosti n. Pak řešení soustavy $Ax = b$ metodou nejmenších čtverců je $x = (A^\top A)^{-1} A^\top b$ a je jednoznačné.*

Důkaz. Věta je dokázána ve skriptech Hladík (2016, věta 8.40).

□

V praxi se řešení nehledá podle vzorce $x = (A^\top A)^{-1} A^\top b$, ale mnohem efektivnější je řešení hledat jako řešení soustavy

$$A^\top A x = A^\top b.$$

Dále už se budeme výhradně zabývat pouze intervalovými maticemi a systémy. V této práci nás bude převážně zajímat množina řešení intervalové soustavy lineárních rovnic řešené metodou nejmenších čtverců.

Definice 23 (Množina řešení metodou nejmenších čtverců)**.** *Buď $\boldsymbol{A} \in \mathbb{IR}^{m \times n}$, $\boldsymbol{b} \in \mathbb{IR}^n$, pak* množina řešení intervalových soustav lineárních rovnic metodou nejmenších čtverců *je definována*

$$\Sigma_{LSQ} = \{x \in \mathbb{R}^n;\ \exists A \in \boldsymbol{A}, b \in \boldsymbol{b} : A^\top A x = A^\top b\}.$$

Soustavu budeme značit jako $\boldsymbol{A}^\top \boldsymbol{A} x = \boldsymbol{A}^\top \boldsymbol{b}$.

Poznámka. Zkratka LSQ je z anglického výrazu least squares.

V článku Černý a kol. (2013) je dokázáno, že rozhodnutí, zda pro dané $\boldsymbol{A} \in \mathbb{IR}^{m \times n}$ a $\boldsymbol{b} \in \mathbb{IR}^n$ je množina Σ_{LSQ} omezená, je co-NP-těžký problém.

Díky těmto faktům bude naším cílem najít zapouzdření množiny všech řešení systému

$$A^\top A x = A^\top b,\ \ A \in \boldsymbol{A}, b \in \boldsymbol{b}. \tag{4.1}$$

4.1 Jednoduché metody

Nejjednodušší možností, jak hledat obálku množiny řešení Σ_{LSQ}, je najít zapouzdření všech řešení soustavy intervalových lineárních rovnic

$$\tilde{\boldsymbol{A}}x = \tilde{\boldsymbol{b}}, \text{kde } \tilde{\boldsymbol{A}} = \boldsymbol{A}^\top \boldsymbol{A},\ \tilde{\boldsymbol{b}} = \boldsymbol{A}^\top \boldsymbol{b}. \tag{4.2}$$

Výsledek $\boldsymbol{A}^\top \boldsymbol{A}$ a $\boldsymbol{A}^\top \boldsymbol{b}$ budeme počítat pomocí intervalové aritmetiky. Jelikož $\boldsymbol{A}$ má rozměry $m \times n$, matice $\boldsymbol{A}^\top \boldsymbol{A}$ má rozměry $n \times n$. Dostáváme tedy čtvercový systém, který lze řešit libovolnou metodou na nalezení obálky množiny Σ. Tento postup má řadu nevýhod. V další kapitole pomocí numerického testování ukážeme, že tato metoda velice často nadhodnocuje množinu řešení Σ_{LSQ}. Intervalový systém (4.2) obsahuje všechny systémy z (4.1) a většinou některé navíc. Což je důvodem nadhodnocení obálky. Některé rovnice se objevují navíc v (4.2), což je způsobeno závislostmi. Jiný pohled na tutéž věc je, že provádíme násobení $\boldsymbol{A}^\top \boldsymbol{A}$, kde díky velkému počtu násobení intervalů dochází k velkému nárůstu intervalů ve výsledné matici.

Možné vylepšení této metody je předpodmínit systém (4.2). K předpodmínění tohoto systému nejčastěji volíme inverzní matici k $\tilde{\boldsymbol{A}}^c$. I když předpodmínění vede k nafukování množiny řešení, i tak je výsledná obálka většinou těsnější.

4.2 Převedení na symetrický případ

V této sekci ukážeme, jak lze ekvivalentně použít metodu nejmenších čtverců. Problém zapouzdření množiny všech řešení intervalových soustav lineárních rovnic metodou nejmenších čtverců zredukujeme na problém zapouzdření všech řešení systému

$$\begin{pmatrix} 0_n & \boldsymbol{A}^T \\ \boldsymbol{A} & I_m \end{pmatrix} \begin{pmatrix} x \\ \hat{x} \end{pmatrix} = \begin{pmatrix} 0 \\ \boldsymbol{b} \end{pmatrix}. \tag{4.3}$$

Řešíme tedy čtvercový intervalový systém, avšak nově vzniklý systém se zvětší. Dostáváme matici velikosti $(m+n) \times (m+n)$.

Věta 20. *Množina řešení* (4.3) *je zahrnuta v množině řešení* (4.2).

Důkaz. Volme libovolný reálný systém obsažený v intervalovém systému (4.3)

$$\begin{pmatrix} 0_n & A^T \\ A' & I_m \end{pmatrix} \begin{pmatrix} x \\ \hat{x} \end{pmatrix} = \begin{pmatrix} 0 \\ b \end{pmatrix},$$

kde $A, A' \in \boldsymbol{A}, b \in \boldsymbol{b}$. Ekvivalentně lze přepsat pomocí blokového násobní matic na dvě soustavy

$$A^\top \hat{x} = 0$$
$$A'x + \hat{x} = b.$$

Druhou rovnici vyjádříme ve tvaru $\hat{x} = b - A'x$ a dosadíme do první. Dostáváme $A^\top(b - A'x) = 0$, což je ekvivalentní

$$A^\top A'x = A^\top b.$$

Jelikož $A^\top A' \in \boldsymbol{A}^\top \boldsymbol{A} = \tilde{\boldsymbol{A}}$ a $A^\top b \in \boldsymbol{A}^\top \boldsymbol{b} = \tilde{\boldsymbol{b}}$, dostáváme rovnici obsaženou v systému (4.2).

□

Důsledek. Buď $\boldsymbol{A} \in \mathbb{IR}^{m\times n}, \boldsymbol{b} \in \mathbb{IR}^n$. Označme $\Sigma_{LSQ}(\boldsymbol{A},\boldsymbol{b})$ jako množinu řešení $\boldsymbol{A}x = \boldsymbol{b}$ metodou nejmenších čtverců. Pak platí

$$\Sigma_{LSQ}(\boldsymbol{A},\boldsymbol{b}) \subseteq \boldsymbol{x},$$

kde $\boldsymbol{x}$ je částí řešení

$$\begin{pmatrix} 0_n & \boldsymbol{A}^T \\ \boldsymbol{A} & I_m \end{pmatrix} \begin{pmatrix} x \\ \hat{x} \end{pmatrix} = \begin{pmatrix} 0 \\ \boldsymbol{b} \end{pmatrix}.$$

Tento důsledek nám dává algoritmus, jak najít obálku množiny řešení Σ_{LSQ}. Pro dané $\boldsymbol{A} \in \mathbb{IR}^{m\times n}, \boldsymbol{b} \in \mathbb{IR}^n$, vyřešíme systém (4.3) některou z metod na nalezení obálky množiny Σ.

Jak je zřejmé z důkazu předchozí věty 20, při řešení intervalové soustavy (4.3) volíme reálné matice z intervalové matice nezávisle na sobě. Proto může docházet k nadhodnocení.

Nevýhodou této metody je, že se zvětšuje původní systém. Avšak tato metoda často dává těsnější obálky. Více se porovnáním metod budeme zabývat v následující kapitole.

4.3 Symetrické soustavy

Předchozí metoda vůbec nebrala v potaz vlastnosti, které se objevují v (4.3). Tato metoda přehlížela závislosti. Důležitým poznatkem je, že matice

$$\begin{pmatrix} 0_n & \boldsymbol{A}^T \\ \boldsymbol{A} & I_m \end{pmatrix},$$

která se objevuje v systému (4.3), je symetrická. Metoda, kterou ukážeme v této sekci, využije této vlastnosti.

4.3.1 Řešení symetrických intervalových soustav lineárních rovnic

Definice 24 (Množina řešení symetrických matic)**.** *Buď $\boldsymbol{A} \in \mathbb{IR}^{n\times n}, \boldsymbol{b} \in \mathbb{IR}^n$ a navíc $\boldsymbol{A}_{ij} = \boldsymbol{A}_{ji}$ pro všechny i, j.* Množina řešení symetrických matic *je definována*

$$\Sigma_{sym} = \{x \in \mathbb{R}^n;\ \exists A \in \boldsymbol{A}, b \in \boldsymbol{b} : A^\top = A, Ax = b\}.$$

Nyní ukážeme, jak zapouzdřit Σ_{sym}. Vektor $\boldsymbol{x}$, který bude obálkou Σ_{sym}, budeme hledat ve tvaru $\boldsymbol{x} = x^* + \boldsymbol{y}$, kde x^* je libovolný reálný vektor a $\boldsymbol{y}$ zapouzdřuje množinu řešení systému

$$C\boldsymbol{A}y = C(\boldsymbol{b} - \boldsymbol{A}x^*).$$

Myšlenka této metody je efektivně vyhodnotit vektor na pravé straně. Pokud bychom tento vektor vyhodnocovali přímo intervalovou aritmetikou, ignorovali bychom tím závislosti. Proto využijeme symetrie $\boldsymbol{A}$, pak l-tá složka vektoru na pravé straně lze vyjádřit jako

$$\boldsymbol{z}_l = \sum_{j=1}^{n} C_{lj}(\boldsymbol{b}_j - \boldsymbol{A}_{jj}x_j^*) - \sum_{j=1}^{n}\sum_{i=1}^{j-1}(C_{li}x_j^* + C_{lj}x_i^*)\boldsymbol{A}_{ij}. \tag{4.4}$$

K nalezení obálky $\boldsymbol{y}$ intervalového lineárního systému $C\boldsymbol{A}y = \boldsymbol{z}$ použijeme libovolný řešič intervalových lineárních soustav rovnic.

Věta 21. *Platí* $C(\boldsymbol{b} - \boldsymbol{A}x^*) \subseteq \boldsymbol{z}$. *Navíc* $\boldsymbol{z}$ *je optimální interval, což znamená*

$$\boldsymbol{z} = \square\{C(b - Ax^*); A \in \boldsymbol{A}, b \in \boldsymbol{b}, A = A^\top\}.$$

Důkaz. S využitím symetrie matice $\boldsymbol{A}$ a subdistributivity intervalové aritmetiky dostáváme

$$\begin{aligned}
\big(C(\boldsymbol{b} - \boldsymbol{A}x^*)\big)_l &= \sum_{j=1}^{n} C_{lj}\Big(b_j - \sum_{i=1}^{n} \boldsymbol{A}_{ji}x_i^*\Big) = \sum_{j=1}^{n} C_{lj}\Big(b_j - \boldsymbol{A}_{jj}x_i^* - \sum_{\substack{i=1\\ i\neq j}}^{n} \boldsymbol{A}_{ij}x_i^*\Big)\\
&\subseteq \sum_{j=1}^{n} C_{lj}(b_j - \boldsymbol{A}_{jj}x_i^*) - \sum_{j=1}^{n} C_{lj} \sum_{\substack{i=1\\ i\neq j}}^{n} \boldsymbol{A}_{ij}x_i^*\\
&\subseteq \sum_{j=1}^{n} C_{lj}(\boldsymbol{b}_j - \boldsymbol{A}_{jj}x_j^*) - \sum_{j=1}^{n}\sum_{i=1}^{j-1}(C_{li}x_j^* + C_{lj}x_i^*)\boldsymbol{A}_{ij} = \boldsymbol{z}_l.
\end{aligned}$$

Jelikož každá proměnná v (4.4) se vyskytuje nejvýše jednou, tak podle tvrzení (6) $\boldsymbol{z}$ je optimální.

$\square$

4.3.2 Metoda využívající symetrie

Zde budeme prezentovat další metodu na zapouzdření řešení přeurčených intervalových soustav lineárních rovnic metodou nejmenších čtverců. Tato metoda bude využívat řešení symetrických intervalových soustav lineárních rovnic. Předchozí poznatky nám dávají možnost formulovat korektní metodu nalezení obálky množiny řešení Σ_{LSQ}.

function METODA VYUŽÍVAJÍCÍ SYMETRIE($\boldsymbol{A}, \boldsymbol{b}$)
  $\tilde{\boldsymbol{A}} := \begin{pmatrix} 0_n & \boldsymbol{A}^T \\ \boldsymbol{A} & I_m \end{pmatrix}$; $\tilde{\boldsymbol{b}} := \begin{pmatrix} 0 \\ \boldsymbol{b} \end{pmatrix}$; $\boldsymbol{z} = 0_{m+n}$;
  $C := (\tilde{A}^c)^{-1}$;
  $x^* :=$ řešení $\tilde{A}^c x = \tilde{b}^c$;
  for $l := 1, \ldots, m+n$ **do**
    $\boldsymbol{z}_l := \sum_{j=1}^{n} C_{lj}(\tilde{\boldsymbol{b}}_j - \tilde{\boldsymbol{A}}_{jj}x_j^*) - \sum_{j=1}^{n}\sum_{i=1}^{j-1}(C_{li}x_j^* + C_{lj}x_i^*)\tilde{\boldsymbol{A}}_{ij}$;
  end for
  $\boldsymbol{y} :=$ zapouzdření řešení $C\tilde{\boldsymbol{A}}y = \boldsymbol{z}$;
  $\boldsymbol{x} := x^* + \boldsymbol{y}$;
  return $(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)^T$;
end function

K řešení $\tilde{A}^c x = \tilde{b}^c$ použijeme standartní algoritmy lineární algebry (například Gaussovu eliminaci). Pro zapouzdření množiny řešení systému $C\tilde{\boldsymbol{A}}y = \boldsymbol{z}$ můžeme zvolit libovolný řešič lineárních intervalových rovnic. Volba metody je důležitá, neboť ovlivňuje těsnost výsledné obálky. Podrobně se tomu budeme věnovat v další kapitole.

4.4 Další metody

Existují další metody na nalezení obálky množiny řešení lineárních intervalových soustav metodou nejmenších čtverců. Jednou z metod je intervalová Choleskyho metoda (Alefeld a Mayer, 1993), která je založena na řešení symetrického intervalového systému pomocí intervalového Choleskyho rozkladu.

4.4.1 Householderova metoda

Další metodou je intervalová Householderova metoda, která je založena na intervalové Householderově transformaci. Metoda je prezentována v Bentbib (2002). Tato metoda se omezuje pouze na intervalové matice rozměru $m \times n$ s lineárně nezávislými sloupci.

Tato metoda využívá intervalové Householderovy transformace, která rozloží $\boldsymbol{A} \subseteq \boldsymbol{QR}$, kde $\boldsymbol{Q} \in \mathbb{IR}^{m\times n}$ a $\boldsymbol{R} \in \mathbb{IR}^{n\times n}$ je horní trojúhelníková matice. Navíc $\boldsymbol{Q}$ je ortogonální. Pak dosadíme rozklad matice $\boldsymbol{A}$ do $\boldsymbol{A}^T\boldsymbol{A}x = \boldsymbol{A}^T x$ a jednoduchými úpravami dostaneme horní trojúhelníkový intervalový systém $\boldsymbol{R}x = \boldsymbol{Q}^T\boldsymbol{b}$. K vyřešení tohoto systému použijeme zpětnou substituci.

Dále uvedeme Houselhoderovu transformaci.

function Intervalová Householderova transformace($\boldsymbol{A}, \boldsymbol{b}$)
  $m, n :=$ rozměry $\boldsymbol{A}$;
  $m_1 := m$;
  $\boldsymbol{R}^* := 0 \in \mathbb{IR}^{m\times n}$, $\boldsymbol{Q}^* := I_m$;
  for $i := 1, \dots, \min(m,n)$ **do**
    $\boldsymbol{a}^* := \boldsymbol{A}(1 : m - i + 1, 1)$;
    $\boldsymbol{A}^* := 0 \in \mathbb{IR}^{(m-i)\times(n-i)}$;
    $\boldsymbol{v} := \boldsymbol{a}^* + \big(\operatorname{sgn}(A^c_{11}) \cdot \|\boldsymbol{a}^*\|\big) \cdot e(1,m - i + 1)$;
    $\boldsymbol{H} :=$ Householderova matice, která vznikne z vektoru UNIV($\boldsymbol{v}$);
    $\boldsymbol{R}^*_{11} := -\operatorname{sgn}(A^c_{11}) \cdot \|\boldsymbol{a}^*\|$;
    for $j := 2, \dots ,n - i + 1$ **do**
      $\boldsymbol{w} := \boldsymbol{H} \cdot \boldsymbol{A}(1 : m_1, j)$;
      $\boldsymbol{A}^*(1 : m_1 - 1, j - 1) := \boldsymbol{w}(2 : m_1)$;
      $\boldsymbol{R}^*_{ij} = \boldsymbol{w}_1$;
    end for
    if $i = 1$ **then**
      $\boldsymbol{Q}^* := \boldsymbol{H}$;
    else
      $\boldsymbol{Q}^* := \boldsymbol{Q}^* \cdot \begin{pmatrix} 1 & 0 \\ 0 & \boldsymbol{H} \end{pmatrix}$;
    end if
    $\boldsymbol{A} := \boldsymbol{A}^*$;
    $m_1 := m_1 - 1$;
  end for
  $\boldsymbol{Q} = \boldsymbol{Q}^*(1 : m, 1 : n)$;
  $\boldsymbol{R} = \boldsymbol{R}^*(1 : n, 1 : n)$;
  return $\boldsymbol{Q}, \boldsymbol{R}$;
end function

Normu intervalového vektoru budeme počítat

$$\|\boldsymbol{v}\| = \sqrt{\sum_{i=1}^{n} \boldsymbol{v}_i^2}.$$

Funkce UNIV je definována

$$\big(\mathrm{UNIV}(\boldsymbol{v})\big)_i = \begin{cases} \frac{\boldsymbol{v}_i}{\|\boldsymbol{v}\|} & \text{pokud } 0 \in \boldsymbol{v}_i, \\ \frac{\mathrm{sgn}(v_i^c)}{\sqrt{1+(\sum_{j\neq i}^{n} \boldsymbol{v}_j^2)/\boldsymbol{v}_i^2}} & \text{pokud } 0 \notin \boldsymbol{v}_i. \end{cases}$$

Intervalová Householderova matice asociována s vektorem $\boldsymbol{w} \in \mathbb{IR}^n$ je definována jako

$$\boldsymbol{H} = I - 2 \cdot (\boldsymbol{w}\boldsymbol{w}^T),$$

kde diagonála $\boldsymbol{w}\boldsymbol{w}^T$ je počítána $(\boldsymbol{w}\boldsymbol{w}^T)_{ii} = \boldsymbol{w}_i^2$. Matice $\boldsymbol{H}$ je symetrická a obsahuje reálné symetrické Householderovy matice H asociované s vektory $v \in \boldsymbol{v}$.

V článku Bentbib (2002) jsou také ukázány různé volby předpodmínění. V následující kapitole otestujeme různé volby předpodmínění.

5. Porovnání metod

V této kapitole porovnáme metody na zapouzdření řešení intervalových soustav metodou nejmenších čtverců. Nejdříve na několika konkrétních systémech ukážeme, jak se metody chovají. Poté provedeme numerické porovnání metod. Metody budeme srovnávat na stejných náhodně vygenerovaných datech. Metody budeme testovat zejména ze dvou hledisek. Zajímat nás bude těsnost obálek a rychlost metod.

5.1 Testovací prostředí

Metody budeme testovat na konkrétních implementacích. Metody jsou implementovány v jazyce Matlab s využitím intervalového toolboxu Intlab. Rychlost metod budeme porovnávat podle výpočetního času. Výpočetní čas záleží na cílové platformě, optimalizaci kódu v jazyce Matlab a překladači. Překladač jazyka Matlab využívá vektorové instrukce, proto je vhodné zdrojové kódy optimalizovat pro překladač Matlabu. Z těchto důvodů se výpočetní čas metod může měnit.

Všechny testy jsme provedli na počítači s těmito parametry:

- PC (x86-64)
- Intel Core i7 6700 (4x 3.4 GHz + HyperThreading)
- 16 GB RAM
- Gentoo Linux

Veškeré testy byly provedeny v prostředí Matlab R2015b a Intlab V6.

5.2 Testované metody

V následujících testech budeme testovat řadu metod, proto zde uvádíme značení a stručný popis metod. Metody jsou podrobně vysvětleny v předchozích kapitolách. V závorce uvádíme označení metody.

- Vestavěná metoda Intlabu verifylss, založená na ϵ-inflaci (verifylss).
- Krawczykova iterační metoda (Krawczyk).
- Intervalová Jacobiho iterační metoda (Jacobi).
- Intervalová Gauss-Seidelova iterační metoda s předpodmínením (G-S, G-S predp).
- Intervalová Gauss-Seidelova iterační metoda bez předpodmínení (G-S bez).
- Metoda Ning-Kearfott (NK).
- Metoda využívající symetrie matice (sym).

- Metoda, kde obálka spočítaná pomocí metody sym se vezme jako počáteční vektor pro iteraci Gauss-Seidelovy metody (sym + G-S).
- Průnik obálek spočítané metodou NK a metodou sym + G-S (intersect).
- Intervalová Householderova metoda (Householder).

5.3 Příklady

Zde uvádíme typické uměle vytvořené příklady na demonstraci metod. Metody otestujeme na třech konkrétních intervalových systémech. Tyto systémy jsou specifické a rozhodně se podle nich nedají metody obecně porovnávat. Ale určitou informaci o metodách nám to zprostředkuje.

Budeme testovat metody na zapouzdření řešení systémů typu

$$\boldsymbol{A}x = \boldsymbol{b}, \quad \text{kde } \boldsymbol{A} \in \mathbb{IR}^{m\times n}, \boldsymbol{b} \in \mathbb{IR}^{n},\ m > n,$$

metodou nejmenších čtverců. První skupinou budou metody, které hledají obálku množiny řešení Σ_{LSQ} jako zapouzdření všech řešení soustavy intervalových lineárních rovnic

$$\tilde{\boldsymbol{A}}x = \tilde{\boldsymbol{b}}, \text{kde } \tilde{\boldsymbol{A}} = \boldsymbol{A}^\top \boldsymbol{A},\ \tilde{\boldsymbol{b}} = \boldsymbol{A}^\top \boldsymbol{b}. \tag{5.1}$$

Další skupina metod bude hledat řešení jako zapouzdření všech řešení systému

$$\begin{pmatrix} 0_n & \boldsymbol{A}^T \\ \boldsymbol{A} & I_m \end{pmatrix} \begin{pmatrix} x \\ \hat{x} \end{pmatrix} = \begin{pmatrix} 0 \\ \boldsymbol{b} \end{pmatrix}. \tag{5.2}$$

Poslední skupinu tvoří přímé metody. Tyto metody řeší přímo $\boldsymbol{A}x = \boldsymbol{b}$ metodou nejmenších čtverců.

U metod budeme měřit výpočetní čas a těsnost obálky. Obálky budeme porovnávat podle relativní šířky obálek. Relativní šířka obálky je definovaná

$$\frac{1}{n}\sum_{i=1}^{n} \frac{w(\boldsymbol{x}_i)}{w(\boldsymbol{v}_i)},$$

kde $\boldsymbol{v}$ je obálka spočítaná metodu **verifylss** a $\boldsymbol{x}$ obálka spočítaná nějakou testovací metodou.

V následujících příkladech budeme mít vždy přeurčený intervalový lineární systém $\boldsymbol{A}x = \boldsymbol{b}$, který bude uveden vždy na začátku. Spočítaná řešení toho systému metodami nejmenších čtverců budou uvedena v tabulkách pod sebou. Nejdříve budou uvedeny metody, které hledají řešení pomocí systému typu (5.1). Dále budou uvedeny metody, které využívají rozšířený systém (5.2). Na konci je vždy uvedena přímá metoda.

5.3.1 První příklad

$$\boldsymbol{A} = \begin{pmatrix} [1,1] & [0,2] \\ [1,1] & [2,2] \\ [1,1] & [3,3] \\ [1,1] & [4,4] \\ [1,1] & [5,5] \end{pmatrix}, \ \boldsymbol{b} = \begin{pmatrix} [2,2] \\ [3,3] \\ [4,4] \\ [5,5] \\ [6,6] \end{pmatrix} \tag{5.3}$$

metoda	**rel. šířka**	**čas (s)**	**obálka**
verifylss	1.000000	0.003909	$([-36, 38.0001], [-10.5, 12.5001])$
G-S bez	0.573532	0.069727	$([-31.6848, 10.8462], [-2.0295, 11.1339])$
G-S predp.	0.928992	0.022697	$([-35.1428, 38.0001], [-7.5, 12.5001])$
Krawczyk	1.000000	0.005210	$([-36, 38.0001], [-10.5, 12.5001])$
NK	0.711098	0.002807	$([-28, 38.0001], [0.30304, 12.5001])$

Tabulka 5.1: Metody, které hledají obálku množiny řešení systému (5.3) pomocí řešení systému typu (5.1).

metoda	**rel. šířka**	**čas (s)**	**obálka**
verifylss	0.031610	0.003694	$([-0.26666, 2.2667], [0.66667, 1.3334])$
G-S	0.029195	0.020380	$([-0.26666, 2.2667], [0.77778, 1.3334])$
Krawczyk	0.031610	0.005490	$([-0.26666, 2.2667], [0.66667, 1.3334])$
sym	0.031610	0.017205	$([-0.26666, 2.2667], [0.66667, 1.3334])$
NK	0.028712	0.003328	$([-0.26666, 2.2667], [0.8, 1.3334])$

Tabulka 5.2: Metody, které hledají obálku množiny řešení systému (5.3) pomocí řešení systému typu (5.2).

metoda	**rel. šířka**	**čas (s)**	**obálka**
Householder	0.477823	0.033543	$([-18.8366, 16.4264], [-3.8832, 7.1365])$

Tabulka 5.3: Přímé metody na řešení systému (5.3).

.

5.3.2 Druhý příklad

$$
\boldsymbol{A} = \begin{pmatrix} [1,1] & [-2.5, -1.5] \\ [1,1] & [-1.5, -0.5] \\ [1,1] & [-0.5, 0.5] \\ [1,1] & [0.5, 1.5] \\ [1,1] & [1.5, 2.5] \\ [1,1] & [2.5, 3.5] \\ [1,1] & [3.5, 4.5] \\ [1,1] & [4.5, 5.5] \\ [1,1] & [5.5, 6.5] \\ [1,1] & [6.5, 7.5] \\ [1,1] & [7.5, 8.5] \end{pmatrix}, \ \boldsymbol{b} = \begin{pmatrix} [1.5, 2.5] \\ [2.5, 3.5] \\ [3.5, 4.5] \\ [4.5, 5.5] \\ [5.5, 6.5] \\ [6.5, 7.5] \\ [7.5, 8.5] \\ [8.5, 9.5] \\ [9.5, 10.5] \\ [10.5, 11.5] \\ [11.5, 12.5] \end{pmatrix} \tag{5.4}
$$

metoda	**rel. šířka**	**čas (s)**	**obálka**
verifylss	1.000000	0.003695	$([-14.5538, 22.5539], [-4.0153, 6.0154])$
G-S bez	0.491055	0.064554	$([-9.8154, 8.4157], [-0.26159, 4.6616])$
G-S predp.	0.868763	0.011413	$([-13.142, 22.5539], [-1.7641, 6.0154])$
Krawczyk	1.000000	0.004266	$([-14.5538, 22.5539], [-4.0153, 6.0154])$
NK	0.749037	0.002699	$([-9.8153, 22.5539], [-0.26153, 6.0154])$

Tabulka 5.4: Metody, které hledají obálku množiny řešení systému (5.4) pomocí řešení systému typu (5.1).

metoda	**rel. šířka**	**čas (s)**	**obálka**
verifylss	0.096403	0.001862	$([2.1859, 5.8142], [0.52342, 1.4766])$
G-S	0.088840	0.123325	$([2.2207, 5.7794], [0.64579, 1.4661])$
Krawczyk	0.094416	0.010387	$([2.2207, 5.7794], [0.53393, 1.4661])$
sym	0.094416	0.050463	$([2.2207, 5.7794], [0.53393, 1.4661])$
NK	0.087030	0.004201	$([2.2207, 5.7794], [0.6821, 1.4661])$

Tabulka 5.5: Metody, které hledají obálku množiny řešení systému (5.4) pomocí řešení systému typu (5.2).

metoda	**rel. šířka**	**čas (s)**	**obálka**
Householder	0.972412	0.104128	$([-16.5557, 19.0428], [-3.2979, 6.5874])$

Tabulka 5.6: Přímé metody na řešení systému (5.4).

5.3.3 Třetí příklad

$$
\boldsymbol{A} = \begin{pmatrix} [0.10001, 0.30001] & [0.90001, 1.1001] \\ [8.9001, 9.1001] & [0.40001, 0.60001] \\ [0.90001, 1.1001] & [6.9001, 7.1001] \end{pmatrix}, \ \boldsymbol{b} = \begin{pmatrix} [0.80001, 1.2001] \\ [-0.2, 0.20001] \\ [1.8001, 2.2001] \end{pmatrix} \tag{5.5}
$$

metoda	rel. šířka	čas (s)	obálka
verifylss	1.000000	0.001159	$([-0.065484, 0.03289], [0.23457, 0.37005])$
G-S bez	0.947837	0.017055	$([-0.065414, 0.02791], [0.24164, 0.36995])$
G-S predp.	0.973396	0.010060	$([-0.065414, 0.03176], [0.24002, 0.36995])$
Krawczyk	0.998565	0.004524	$([-0.065414, 0.03282], [0.23467, 0.36995])$
NK	0.972531	0.002673	$([-0.065414, 0.03171], [0.24019, 0.36995])$

Tabulka 5.7: Metody, které hledají obálku množiny řešení systému (5.5) pomocí řešení systému typu (5.1).

metoda	rel. šířka	čas (s)	obálka
verifylss	0.647312	0.001478	$([-0.04704, 0.014449], [0.25695, 0.34766])$
G-S	0.636521	0.024646	$([-0.0469, 0.013911], [0.25873, 0.34746])$
Krawczyk	0.644344	0.004926	$([-0.0469, 0.014308], [0.25716, 0.34746])$
sym	0.633658	0.010509	$([-0.046826, 0.014234], [0.2585, 0.34611])$
NK	0.636275	0.003058	$([-0.0469, 0.013897], [0.25878, 0.34746])$

Tabulka 5.8: Metody, které hledají obálku množiny řešení systému (5.5) pomocí řešení systému typu (5.2).

metoda	rel. šířka	čas (s)	obálka
Householder	1.301411	0.018972	$([-0.077802, 0.03784], [0.2087, 0.40206])$

Tabulka 5.9: Přímé metody na řešení systému (5.5).

5.3.4 Shrnutí příkladů

První věc, která vyplývá z předchozích příkladů, je, že pro těsnost obálek je vždy lepší řešit systém typu (5.2). Avšak pro obdélníkové matice ($n \ll m$) se velmi výrazně zvyšuje výpočetní čas. Tyto dvě kritéria jdou často proti sobě. Záleží na tom, jaké máme požadavky na metodu. Zda chceme co nejrychleji výsledek nebo nám jde o co nejtěsnější obálku.

Příklady také demonstrují, že při předpodmínění může dojít k zvětšení množiny řešení. Výsledná obálka je při předpodmínění nadhodnocená. Naopak ve většině případů díky předpodmínění dosáhneme těsnější obálky. Tyto příklady jsou velice specifické a uměle vytvořené. Například matice v prvním příkladu obsahuje pouze jeden interval s nenulovou šířkou.

Intervalovou Householderovu metodu lze jen zřídka v praxi použít. Tato metoda sice často vydá těsnější obálku než metody, které řeší systém typu (5.1), ale je časově náročnější než metody, které řeší systém typu (5.2). Metodám řešící systém (5.2) nemůže konkurovat v těsnosti obálky. Tato metoda by mohla mít jedině uplatnění, pokud by poměr rovnic a proměnných byl hodně velký.

Metoda **verifylss** nedává nejtěsnější obálku, ale je velice rychlá.

Další výsledky se stěží dají vyvozovat na základě těchto příkladů, proto musíme provést větší a komplexnější sadu testů.

5.4 Numerické testování

Zde budeme testovat metody na náhodných datech. Postupně budeme generovat různé typy náhodných intervalových lineárních soustav.

5.4.1 Systém testování

Generují se náhodné intervalové přeurčené systémy a na nich se testují metody pro nalezení obálky metodou nejmenších čtverců. U metod se měří výpočetní čas a těsnost obálky.

Systémy generujeme podle následujících kritérií.

- Náhodně se generují intervalové matice a vektory s rozdílnými rozměry a rozdílnými pevnými poloměry.
- Postupně se generují větší systémy.
- Středové matice A^c a středové vektory b^c jsou generovány rovnoměrným rozdělením v intervalu $[-10, 10]$.
- A^Δ a b^Δ se negenerují. Volí se různé pevné poloměry $\delta > 0$.

Metody budeme testovat podle následujících kritérií.

- Pro každé rozměry matice m, n a poloměr δ se generuje počet # náhodných systémů s těmito parametry.
- Měří se průměrný čas pro daný druh systému.
- Měří se průměrná šířka obálky.
- Pro těsnost obálky se používá relativní poměr velikosti obálek. Těsnost testujeme vůči metodě **verifylss**. Čím je tento poměr menší, tím je obálka těsnější.
- Místo systému

  $$\boldsymbol{A}^T\boldsymbol{A}x = \boldsymbol{A}^T\boldsymbol{b},$$

  budeme řešit pouze systém

  $$\begin{pmatrix} 0_n & \boldsymbol{A}^T \\ \boldsymbol{A} & I_m \end{pmatrix} \begin{pmatrix} x \\ \hat{x} \end{pmatrix} = \begin{pmatrix} 0 \\ \boldsymbol{b} \end{pmatrix}. \tag{5.6}$$

- Do času se nezapočítává sestavení systému (5.6), pouze se počítá čas na jeho vyřešení.

Tučně budeme značit metodu s nejtěsnější obálkou, resp. nejrychlejším výpočetním časem. Metody jsou sice rozděleny do dvou tabulek, ale vždy se metody testovaly na stejných datech v rámci jedné kategorie.

5.4.2 Výsledky numerického testování metod

Těsnost obálek

m	n	δ	#	Krawczyk	Jacobi	Gauss-Seidel
5	2	1	50	0.991971	**0.945672**	0.945672
5	2	0.1	50	0.995586	**0.978327**	0.978327
5	2	0.01	50	0.999306	**0.997534**	0.997534
6	3	1	50	1	**0.981721**	0.981721
6	3	0.1	50	0.990262	**0.971465**	0.971465
6	3	0.01	50	0.998929	**0.996879**	0.996879
8	4	0.1	50	0.989053	**0.971355**	0.971355
8	4	0.01	50	0.998605	**0.996467**	0.996467
8	4	0.001	50	0.999846	**0.999653**	0.999653
15	7	0.1	50	0.990296	**0.974512**	0.974512
15	7	0.01	50	0.998088	**0.996175**	0.996175
15	7	0.001	50	0.999725	**0.99953**	0.99953
20	10	0.1	50	0.986614	**0.9726**	0.9726
20	10	0.01	50	0.997784	**0.995822**	0.995822
20	10	0.001	50	0.999612	**0.999412**	0.999412
30	14	0.1	20	0.98777	**0.977428**	0.977428
30	14	0.01	20	0.997824	**0.995926**	0.995926
30	14	0.001	20	0.999485	**0.99929**	0.99929

Tabulka 5.10: Numerické testování těsnosti obálek iterativních metod na menších systémech.

m	n	δ	#	NK	sym	sym + G-S	intersect
5	2	1	50	0.860649	0.85409	0.831959	**0.785665**
5	2	0.1	50	0.977081	0.856014	**0.856009**	0.856009
5	2	0.01	50	0.997522	**0.85555**	0.85555	0.85555
6	3	1	50	0.896233	0.877031	0.869194	**0.826785**
6	3	0.1	50	0.969006	0.838095	**0.838055**	0.838055
6	3	0.01	50	0.996851	**0.849309**	0.849309	0.849309
8	4	0.1	50	0.967662	0.835539	0.835539	**0.835459**
8	4	0.01	50	0.996425	**0.842352**	0.842352	0.842352
8	4	0.001	50	0.999653	**0.851807**	0.851807	0.851807
15	7	0.1	50	0.96881	**0.832023**	0.832023	0.832023
15	7	0.01	50	0.996122	**0.833708**	0.833708	0.833708
15	7	0.001	50	0.999529	**0.83333**	0.83333	0.83333
20	10	0.1	50	0.963241	**0.818506**	0.818506	0.818506
20	10	0.01	50	0.995741	**0.826665**	0.826665	0.826665
20	10	0.001	50	0.999411	**0.828692**	0.828692	0.828692
30	14	0.1	20	0.964234	**0.814359**	0.814359	0.814359
30	14	0.01	20	0.995823	**0.819963**	0.819963	0.819963
30	14	0.001	20	0.999289	**0.825749**	0.825749	0.825749

Tabulka 5.11: Numerické testování těsnosti obálek zbylých metod na menších systémech.

m	n	δ	#	Krawczyk	Jacobi	Gauss-Seidel
50	15	0.01	10	0.997772	**0.996067**	0.996067
50	15	0.001	10	0.999511	**0.999337**	0.999337
50	20	0.01	10	0.998443	**0.996648**	0.996648
50	20	0.001	10	0.999319	**0.999133**	0.999133
100	30	0.01	10	0.991326	**0.989712**	0.989712
100	30	0.001	10	0.999126	**0.998955**	0.998955

Tabulka 5.12: Relativní těsnost obálek iterativních metod na větších systémech.

m	n	δ	#	NK	sym	sym + G-S	intersect
50	15	0.01	10	0.99598	**0.828979**	0.828979	0.828979
50	15	0.001	10	0.999336	**0.833805**	0.833805	0.833805
50	20	0.01	10	0.996519	**0.819833**	0.819833	0.819833
50	20	0.001	10	0.999132	**0.82126**	0.82126	0.82126
100	30	0.01	10	0.98955	**0.822795**	0.822795	0.822795
100	30	0.001	10	0.998953	**0.829305**	0.829305	0.829305

Tabulka 5.13: Relativní těsnost obálek zbylých metod na větších systémech.

Výpočetní čas

m	n	δ	#	verifylss	Krawczyk	Jacobi	Gauss-Seidel
5	2	1	50	**0.006665**	0.014122	0.1563	0.107503
5	2	0.1	50	**0.001607**	0.005709	0.060714	0.042291
5	2	0.01	50	**0.001444**	0.003766	0.033559	0.027212
6	3	1	50	**0.00386**	0.00413	0.05888	0.038968
6	3	0.1	50	**0.001839**	0.006945	0.097559	0.065859
6	3	0.01	50	**0.001464**	0.004026	0.046951	0.038164
8	4	0.1	50	**0.001859**	0.007638	0.145708	0.096051
8	4	0.01	50	**0.001477**	0.004302	0.069145	0.053609
8	4	0.001	50	**0.001478**	0.003326	0.043333	0.040177
15	7	0.1	50	**0.002**	0.010009	0.372883	0.22944
15	7	0.01	50	**0.001518**	0.00468	0.140839	0.1067
15	7	0.001	50	**0.001517**	0.003543	0.085637	0.072374
20	10	0.1	50	**0.002408**	0.013515	0.712965	0.424545
20	10	0.01	50	**0.001545**	0.005015	0.209625	0.157156
20	10	0.001	50	**0.001551**	0.003788	0.128545	0.11119
30	14	0.1	20	**0.003368**	0.017165	1.33433	0.895031
30	14	0.01	20	**0.001722**	0.005676	0.340636	0.25172
30	14	0.001	20	**0.00176**	0.004219	0.201168	0.170688
50	15	0.01	10	**0.002089**	0.006536	0.527824	0.383281
50	15	0.001	10	**0.002112**	0.004838	0.297588	0.253839
50	20	0.01	10	**0.002104**	0.006918	0.60809	0.439351
50	20	0.001	10	**0.002188**	0.005112	0.350956	0.267994
100	30	0.01	10	**0.003621**	0.011513	1.42909	1.00285
100	30	0.001	10	**0.002872**	0.007285	0.672516	0.573072

Tabulka 5.14: Výpočetní čas v sekundách jednotlivých iterativních metod.

m	n	δ	#	NK	sym	sym + G-S
5	2	1	50	**0.004912**	0.019259	0.028577
5	2	0.1	50	**0.004186**	0.019841	0.026819
5	2	0.01	50	**0.004129**	0.019792	0.026105
6	3	1	50	**0.00131**	0.008602	0.012557
6	3	0.1	50	**0.00451**	0.030024	0.03842
6	3	0.01	50	**0.004491**	0.029923	0.037154
8	4	0.1	50	**0.004909**	0.0485	0.058092
8	4	0.01	50	**0.004907**	0.048942	0.058559
8	4	0.001	50	**0.004951**	0.04915	0.058847
15	7	0.1	50	**0.006493**	0.15072	0.168106
15	7	0.01	50	**0.006525**	0.151281	0.168598
15	7	0.001	50	**0.006459**	0.150533	0.167727
20	10	0.1	50	**0.007732**	0.275226	0.298027
20	10	0.01	50	**0.00779**	0.277732	0.296839
20	10	0.001	50	**0.007701**	0.276066	0.29856
30	14	0.1	20	**0.01068**	0.588737	0.633309
30	14	0.01	20	**0.010502**	0.59232	0.627172
30	14	0.001	20	**0.010682**	0.601771	0.633966
50	15	0.01	10	**0.015057**	1.28397	1.33495
50	15	0.001	10	**0.016224**	1.31203	1.36626
50	20	0.01	10	**0.015647**	1.49273	1.51676
50	20	0.001	10	**0.016982**	1.46167	1.55126
100	30	0.01	10	**0.028548**	5.13954	5.19927
100	30	0.001	10	**0.026467**	5.01417	5.13243

Tabulka 5.15: Výpočetní čas v sekundách zbylých metod.

5.4.3 Shrnutí numerického testování

Gauss-Seidelova iterační metoda je zrychlením Jacobiho metody. Tyto dvě metody dávají stejně těsnou obálku, akorát Gauss-Seidelova metoda udělá menší počet iterací. To je způsobeno tím, že Gauss-Seidelova metoda používá ihned aktualizované hodnoty a nečeká až do další iterace.

Metoda **verifylss** je velice rychlá, avšak nedává příliš těsnou obálku. Tato metoda byla navržena na systémy, ve kterých intervaly mají malé šířky. Všimněme si, že rychlost této metody je ovlivněna šířkou intervalů. Velikost systému se projevuje na rychlosti velice málo. Výpočetní čas této metody se stále pohybuje v řádech milisekund. Podobné pozorování se dá říci i o Krawczykově metodě. Naopak rychlost Gauss-Seidelovy a Jacobiho metody závisí na velikosti systému. Rychlost metody **NK** je do značné míry ovlivněna pouze rozměry systému, neboť tato metoda využívá explicitní formule k nalezení obálky.

Testování ukázalo, že metoda využívající symetrie dává těsnější obálku než ostatními metody, avšak pro velké systémy je časově náročná. Ukázalo se, že tuto obálku lze zmenšit tím, že jí dáme na vstup Gauss-Seidelově iterační metodě. Pokud by po předpodmínění platilo $(C\boldsymbol{A})^c = I$, tak by metoda **sym + G-S** dávala stejnou obálku jako metoda **sym**. U malých systémů se vyplatilo udělat průnik obálek spočtené metodami **sym + G-S** a **NK**.

Metodu **sym** lze zrychlit pomocí paralelního počítání 4.4. Neboť vyhodnocení toho vzorce trvá převážnou část výpočtu.

Obecně lze těžko posoudit, která metoda je nejlepší. Neboť záleží na výběru kritéria, zda chceme rychlou metodu nebo metodu, která vydá těsnou obálku. Rozhodně, ale můžeme konstatovat, že metoda **verifylss** je až na malé výjimky nejrychlejší. Dále můžeme tvrdit, že využitím symetrie lze dosáhnout těsné obálky. U větších systémů metoda **sym** dává výrazně těsnější obálku než ostatní metody.

5.5 Speciální případy

Zde budeme testovat Householderovu metodu. Jelikož tato metoda se ukázala jako méně použitelná v porovnání s jinými, budeme ji testovat na systémech s velkým počtem rovnic a malým počtem proměnných. Těsnost obálek testujeme vůči metodě **NK**, která řeší $\boldsymbol{A}^T\boldsymbol{A}x = \boldsymbol{A}^T x$.

Mějme intervalový lineární systém $\boldsymbol{A}x = \boldsymbol{b}$, kde $\boldsymbol{A} \in \mathbb{IR}^{m\times n}, \boldsymbol{b} \in \mathbb{IR}^n$. Označme:

- $QR = A^c$ je reálný QR rozklad matice A^c
- $R_1 \in \mathbb{R}^{n\times n}$ vznikne z R odstraněním posledních $(m-n)$ řádků
- $\boldsymbol{B} = Q^\top \boldsymbol{A} R_1^{-1}$
- $\boldsymbol{c} = Q^\top \boldsymbol{b}$
- x_0 vznikne z vektoru $\boldsymbol{c}^c$ odebráním posledních $(m-n)$ prvků
- d^c vznikne s c^c nahrazením prvních n prvků nulami
- $d^\Delta = B^\Delta \cdot |x_0| + c^\Delta$
- $\boldsymbol{h} = [d^c - B^\Delta \cdot |x_0|, d^c + B^\Delta \cdot |x_0|]$

Dále označme $x^H_{\boldsymbol{A},\boldsymbol{b}}$ jako obálku množiny řešení $\boldsymbol{A}x = \boldsymbol{b}$ metodou nejmenších čtverců spočítanou intervalovou Householderovou metodou.

Pak budeme porovnávat následující obálky:

- $\boldsymbol{v}_1 = x^H_{\boldsymbol{A},\boldsymbol{b}}$
- $\boldsymbol{v}_2 = R_1^{-1} \cdot x^H_{\boldsymbol{B},\boldsymbol{c}}$
- $\boldsymbol{v}_3 = R_1^{-1} \cdot (x_0 + x^H_{\boldsymbol{A},\boldsymbol{d}})$
- $\boldsymbol{v}_4 = R_1^{-1} \cdot (x^H_{\boldsymbol{B},c^c} + x^H_{\boldsymbol{B},\boldsymbol{c}})$
- $\boldsymbol{v}_5 = R_1^{-1} \cdot (x_0 + x^H_{\boldsymbol{B},\boldsymbol{h}} + x^H_{\boldsymbol{B},\boldsymbol{c}})$
- $\boldsymbol{x}^{NK}$ obálku systému $\boldsymbol{A}^T\boldsymbol{A}x = \boldsymbol{A}^T\boldsymbol{b}$ spočítanou metodu Ning-Kearfott

Tyto volby předpodmínění Householderovy metody jsou dokázány v článku Bentbib (2002).

Počet opakování testů je 20 pro všechny typy matic.

Těsnost obálek

m	n	δ	v_1	v_2	v_3	v_4	v_5	x^{NK}
2	1	1	**0.948417**	0.969268	1.07007	1.56379	1.61895	1
2	1	0.1	1.18327	**0.752774**	0.766266	1.13497	1.14209	1
2	1	0.01	1.0853	**0.770412**	0.771789	1.14749	1.14817	1
2	1	0.001	1.05354	**0.724673**	0.724785	1.0875	1.08756	1
5	1	1	1.41818	1.14763	1.3264	1.80358	1.9341	**1**
5	1	0.1	1.42138	**0.991399**	1.00851	1.62277	1.63418	1
5	1	0.01	1.41615	**0.984391**	0.986	1.59241	1.59343	1
5	1	0.001	1.33636	**0.930422**	0.930578	1.47212	1.47222	1
8	2	0.1	3.90488	0.94604	**0.92378**	1.62622	1.59935	1
8	2	0.01	3.94022	1.05055	**0.99162**	1.79138	1.73192	1
8	2	0.001	3.73807	0.89755	**0.842421**	1.53906	1.48389	1
15	2	0.1	5.46347	1.1969	1.1702	2.02679	1.99543	**1**
15	2	0.01	4.3145	1.12051	1.06563	1.91203	1.8566	**1**
15	2	0.001	4.23822	1.0942	1.04249	1.87452	1.82276	**1**
50	1	1	1.64636	1.68298	1.76223	2.64135	2.72061	**1**
50	1	0.1	1.67534	1.28562	1.30385	2.09947	2.11143	**1**
50	1	0.01	1.66389	1.26635	1.26827	2.08405	2.08532	**1**
50	1	0.001	1.69318	1.27025	1.27045	2.10378	2.10391	**1**
60	2	0.1	6.49125	1.41397	1.39512	2.39961	2.37601	**1**
60	2	0.01	5.8955	1.39961	1.36687	2.4051	2.37173	**1**
60	2	0.001	5.60604	1.36263	1.33159	2.32892	2.29782	**1**

Tabulka 5.16: Numerické testování těsnosti obálek Householderovy metody.

Výpočetní čas

m	n	δ	v_1	v_2	v_3	v_4	v_5	x^{NK}
2	1	1	0.00450	0.004077	0.004108	0.00815	0.00807	**0.003229**
2	1	0.1	0.00451	0.004104	0.004134	0.00832	0.00833	**0.003171**
2	1	0.01	0.00449	0.004095	0.004118	0.00817	0.00812	**0.003176**
2	1	0.001	0.00447	0.004115	0.00412	0.00816	0.00810	**0.003152**
5	1	1	0.01037	0.008863	0.008987	0.01793	0.01759	**0.003156**
5	1	0.1	0.01055	0.008837	0.008824	0.01759	0.01755	**0.003157**
5	1	0.01	0.01058	0.008831	0.008822	0.01758	0.01756	**0.003142**
5	1	0.001	0.01055	0.008845	0.008874	0.01764	0.01758	**0.003134**
8	2	0.1	0.03652	0.030861	0.030946	0.06169	0.06172	**0.003354**
8	2	0.01	0.03666	0.030842	0.030877	0.06177	0.06172	**0.003355**
8	2	0.001	0.03664	0.030910	0.030926	0.06178	0.06162	**0.003337**
15	2	0.1	0.10137	0.089624	0.089882	0.17967	0.17898	**0.003328**
15	2	0.01	0.10240	0.089804	0.089839	0.17973	0.18039	**0.003334**
15	2	0.001	0.10187	0.090141	0.090013	0.18046	0.18014	**0.003356**
50	1	1	0.48217	0.460458	0.459201	0.92963	0.93428	**0.003260**
50	1	0.1	0.49451	0.494648	0.494625	0.98847	0.96879	**0.003217**
50	1	0.01	0.48149	0.465811	0.465553	0.93011	0.93448	**0.003251**
50	1	0.001	0.50522	0.479161	0.473391	0.94607	0.95543	**0.003384**
60	2	0.1	1.36310	1.297960	1.292140	2.59472	2.58988	**0.003418**
60	2	0.01	1.33629	1.311430	1.316670	2.60552	2.65999	**0.003365**
60	2	0.001	1.29722	1.265000	1.281400	2.51909	2.49757	**0.003229**

Tabulka 5.17: Výpočetní čas v sekundách Householderovi metody.

5.5.1 Shrnutí testování speciálních případů

Householderova metoda využívá intervalovou QR faktorizaci. Tedy rozkládá původní matici $\boldsymbol{A} \in \mathbb{IR}^{m\times n}$ na horní trojúhelníkovou matici $\boldsymbol{R} \in \mathbb{IR}^{n\times n}$ a ortogonální matici $\boldsymbol{Q} \in \mathbb{IR}^{m\times n}$. Vždy platí $\boldsymbol{A} \subseteq \boldsymbol{QR}$. Máme tedy jistotu, že všechny matice obsažené v $\boldsymbol{A}$ jsou obsažené i v $\boldsymbol{QR}$. Často $\boldsymbol{QR}$ obsahuje spoustu matic navíc, což je způsobené velkým počtem intervalových operací v Householderově transformaci.

Další nevýhodou Householderovy metody je, že Householderova transformace je časově náročná. Některé obálky byly dokonce spočteny pomocí dvou Householderových metod, což se projevilo na výpočetním čase.

Obálky $\boldsymbol{v}_4$ a $\boldsymbol{v}_5$ jsou vždy horší než $\boldsymbol{x}^{NK}$, $\boldsymbol{x}^{NK}$ je pokaždé těsnější a navíc výpočet $\boldsymbol{v}_4$ a $\boldsymbol{v}_5$ trvá dlouho.

Se vzrůstajícími rozměry systému se zvětšuje i obálka $\boldsymbol{v}_1$. Konkrétně se obálka zvětšuje, když se zvětšuje poměr m/n.

Pokud bychom chtěli zvolit Householderovu metodu, tak nejlepší volbou je obálky počítat pomocí $\boldsymbol{v}_2$ nebo $\boldsymbol{v}_3$. Výpočetní čas těchto obálek je skoro stejný jako výpočetní čas $\boldsymbol{v}_1$, neboť v těchto případech je použita pouze jedna intervalová Householderova transformace. Všimněme si, že při malých rozměrech systému jsou obálky $\boldsymbol{v}_2$ a $\boldsymbol{v}_3$ těsnější než $\boldsymbol{x}^{NK}$ a při větších rozměrech systému se obálky příliš nenadhodnocují.

Využití Houselderovy metody v praxi je malé, neboť ve většině případů nadhodnocuje obálku a vždy je pomalá. Avšak jak jsme ukázali, existují systémy, ve kterých by tato metoda mohla najít využití.

6. Uživatelská dokumentace

V této kapitole popíšeme návod, jak zprovoznit a obsluhovat přiloženou implementaci řešiče. Všechny metody jsou napsány v jazyce Matlab s využitím intervalového toolboxu Intlab. Popíšeme, jak zprovoznit Intlab. Také vysvětlíme, jak pracovat s intervalovými daty v Intlabu. Poté ukážeme, jak používat přiložené metody. Programovacímu jazyku Matlab se věnovat nebudeme. Pro pochopení stačí ovládat základ Matlabu.

6.1 Intlab

Intlab (zkratka z anglického Interval Laboratory) je knihovna určená zejména pro Matlab, která umožňuje práci s intervalovými daty.

6.1.1 Instalace Intlabu

Budeme se zabývat pouze instalací Intlabu pro Matlab. Dříve Intlab fungoval pouze v Matlabu. Ale od nové verze Intlabu lze Intlab používat i v Octave. Poté co si obstaráme knihovnu Intlab, je nutné přidat Intlab do cesty Matlabu. Nejdříve musíme upravit soubor *startup.m*. Tento soubor se nachází v adresáři Intalabu. Hned na začátku tohoto souboru přepíšeme řádek typu

```
cd [adresář]
```

tím, že do uvozovek napíšeme plnou cestu k Intlabu. Pak máme dvě možnosti, jak spustit Intlab.

První možnost je spustit Intlab dočasně. To provedeme tím, že v příkazovém řádku Matlabu se přepneme do adresáře s knihovnu Intlab, poté provedeme příkaz

```
startintlab
```

a pak už můžeme používat Intlab. Tento postup je vhodný zejména, pokud používáme pouze konzoli v Matlabu. Například pokud jsme se připojili na vzdálený počítač s Matlabem.

Druhou možností je přidat do seznamů cest Matlabu adresář s knihovnou Intlab. To nám zajistí to, že při každém spuštění Matlabu bude Intlab automaticky připraven k použití.

6.1.2 Intervaly v Intlabu

Abychom mohli používat intervalové matice, musíme nejdříve umět definovat interval. V Intlabu jsou tři možnosti na to, jak vytvořit interval. První možností je funkce **intval**.

```
>> x = intval(1)
intval x =
[    1.0000,    1.0000]
>> x = intval(1/3)
intval x =
[    0.3333,    0.3334]
```

Všimněme si, že Intlab automaticky provádí zaokrouhlování, pokud meze intervalu nelze reprezentovat. Pokud bychom chtěli měnit zaokrouhlování, lze použít funkce **setround**. Další možností je definovat interval pomocí dolní a horní meze, k tomu slouží funkce **infsup**. Nebo můžeme interval definovat pomocí středu a poloměru, na to nám poslouží funkce **midrad**.

```
>> x = infsup(1,2)
intval x =
[    1.0000,    2.0000]
>> x = midrad(1.5,0.5)
intval x =
[    1.0000,    2.0000]
```

Intervalové matice se v Intlabu definují analogicky a opět třemi stejnými funkcemi.

```
>> A = intval([1 2; 3 4])
intval A =
[    1.0000,    1.0000] [    2.0000,    2.0000]
[    3.0000,    3.0000] [    4.0000,    4.0000]
>> A = infsup([1 2; 3 4], [4/3 3; 4 5])
intval A =
[    1.0000,    1.3334] [    2.0000,    3.0000]
[    3.0000,    4.0000] [    4.0000,    5.0000]
>> A = midrad([1 2; 3 4], 1/100 )
intval A =
[    0.9899,    1.0101] [    1.9899,    2.0101]
[    2.9899,    3.0101] [    3.9899,    4.0101]
>> A = midrad([1 2; 3 4], [2 1; 0 1] )
intval A =
[   -1.0000,    3.0000] [    1.0000,    3.0000]
[    3.0000,    3.0000] [    3.0000,    5.0000]
```

Jestliže bychom měli už definovaný interval, nebo intervalovou matici a zajímal by nás poloměr, střed nebo meze intervalu, tak k tomu slouží následující funkce. Poloměr a střed získáme pomocí funkcí **rad** resp. **mid**. Dolní a horní mez vrací funkce **inf** resp. **sup**. Všechny tyto funkce mají jeden stejný parametr a to interval nebo intervalovou matici.

```
>> xc = mid(x)
xc =
    1.5000
>> Ac = mid(A)
Ac =
     1     2
     3     4
```

Základní operace a funkce jsou přetížené. Tedy aritmetické operace fungují i pro intervalová data. Rovněž můžeme použít funkce (sin, cos, exp, ...), které jsou standardně definované pro reálná čísla. Rovněž v Intalabu fungují standartní maticové funkce jako inverze nebo transpozice matice.

6.1.3 Lineární soustavy rovnic

Intlab obsahuje funkci **verifylss** na výpočet obálky množiny řešení intervalových soustav lineárních rovnic. Funkce **verifylss**$(\mathbf{A}, \mathbf{b})$ má dva parametry - reálnou nebo intervalovou matici a reálný nebo intervalový vektor. V Matlabu je možné využít zápisu $\mathbf{A} \setminus \mathbf{b}$.

6.2 Volání řešiče

Součástí této práce je balíček funkcí implementovaných v prostředí Matlab s využitím Intlabu. Jsou zde metody na nalezení obálky množiny řešení lineárních intervalových rovnic metodou nejmenších čtverců, také řešič symetrických soustav. Dále můžeme najít řešiče intervalových lineárních soustav a řadu dalších pomocných funkcí. Není potřeba nic instalovat, stačí pouze volat v Matlabu příslušné funkce. Všechny funkce jsou v samostatných M-souborech a podrobná dokumentace těchto funkcí je umístěna na začátku každého souboru. Dokumentace obsahuje podrobný popis funkcí a všech parametrů funkce.

6.2.1 Názvy funkcí

Všechny názvy souborů jsou značeny malými písmeny a název je zřetězením několika zkratek. Zkratky udávají typ problému, metodu řešení a typ výsledku. V následující tabulce uvádíme některé důležité zkratky. Zkratka pro metodu řešení je většinou jméno autorů metod nebo jejich iniciály.

značení	umístění	význam
i	na začátku	funkce pracující s intervaly
ils	na začátku	soustava lineárních intervalových rovnic
iols	na začátku	přeurčená soustava lineárních intervalových rovnic
sym	uprostřed	funkce pracuje nějakým způsobem se symetrií matice
enc	na konci	obálka zapouzdřující do intervalu všechna řešení

Tabulka 6.1: Seznam zkratek v názvu souboru v přiloženém balíčku funkcí.

Pokud na první pohled není vidět, co by funkce měla dělat, stačí otevřít soubor této funkce, kde je vše popsáno.

6.2.2 Parametry funkcí

Všechny funkce začínající zkratkou **ils** nebo **iols** většinou obsahují dva povinné vstupní parametry (intervalovou matici a intervalový vektor). Další parametry jsou většinou nepovinné. Intervalová data mají prefix **i**. Matice jsou značeny velkými písmeny ze začátku abecedy, vektory naopak malými písmeny. Pokud název funkce končí zkratkou **enc** znamená to, že první výstupní parametry bude intervalový vektor.

6.2.3 Metoda nejmenších čtverců

Balíček metod obsahuje funkci na řešení intervalových soustav metodou nejmenších čtverců. Tato funkce zapouzdřuje do intervalu všechna řešení lineárního intervalového systému metodou nejmenších čtverců a její název je **iolslsqenc**.

Funkce má tři vstupní parametry, z toho dva povinné, a dva výstupní parametry.

parametr	povinné	význam
iA	ano	intervalová matice koeficientů soustavy $\boldsymbol{iA} \in \mathbb{IR}^{m\times n}$
ib	ano	intervalový vektor pravých stran $\boldsymbol{ib} \in \mathbb{IR}^{n}$
tradeoff	ne	udává kompromis mezi časem řešení a těsností obálky

Tabulka 6.2: Seznam vstupních parametrů funkce **iolslsqenc**.

parametr	význam
x	obálka zapouzdřující do vektoru množinu řešení
status	výstupní příznaky

Tabulka 6.3: Seznam výstupních parametrů funkce **iolslsqenc**.

Jak jsme již ukázali, pro řešení intervalových soustav metodou nejmenších čtverců existuje řada metod. Některé metody jsou rychlé, jiné naopak dávají těsnější obálku. V předchozí kapitole jsme ukázali, že tyto dvě kritéria jdou často proti sobě. Proto poslední parametr funkce udává kompromis mezi časem a těsností. Možné hodnoty toho parametru jsou:

- 'FASTEST': nejrychlejší možná metoda
- 'FASTER': spíše rychlá metoda
- 'EFFECTIVE': efektivní kompromis (standartní hodnota, pokud tento parametr není nastaven)
- 'TIGHTER': těsnější obálka
- 'TIGHTEST': nejtěsnější obálka
- 'DEFAULT': standartní metoda (='EFFECTIVE')

Poslední výstupní parametr udává, zda zvolená metoda našla úspěšně obálku.

6.2.4 Příklad volání funkce

Pro ilustraci zde uvádíme příklad, jak pracovat s některými předloženými funkcemi. Práce s ostatními funkcemi je velice podobná, proto zde ukážeme jen jednu jednoduchou funkci. Vysvětlíme, jak pracovat s funkcí **ilskrawczykenc**, která hledá obálku řešení lineárních intervalových rovnic Krawczykovou metodou. Pro jednoduchost popíšeme jen základní možnost volání, neboť tato funkce obsahuje celkem osm parametrů. Tato funkce očekává dva povinné vstupní parametry.

Uvažme následující intervalovou soustavu

$$\begin{pmatrix} [2,4] & [-1,0] \\ [0,1] & [1,2] \end{pmatrix} x = \begin{pmatrix} [-2,5] \\ [-3,5] \end{pmatrix}.$$

Následující kód ukazuje použití metody **ilskrawczykenc**.

```
>> A = infsup ( [ 2 -1; 0 1], [4 0; 1 2])
intval A =
[    2.0000,    4.0000] [   -1.0000,    0.0000]
[    0.0000,    1.0000] [    1.0000,    2.0000]
>> b = infsup([-2 -3], [5 5])'
intval b =
[   -2.0000,    5.0000]
[   -3.0000,    5.0000]
>> ilskrawczykenc(A,b)
intval ans =
[   -5.4136,    6.5715]
[   -8.7670,    9.7143]
```

Metoda nám vrátila obálku množiny řešení. Avšak intervalový obal množiny řešení je $([-2,5],[-5.5,6.0])^T$.

7. Programátorská dokumentace

V této kapitole popíšeme implementační detaily programu. Všechny algoritmy jsou popsány v pseudokódu v předchozích kapitolách. Zde uvedeme technické detaily a odlišnosti, nebudeme uvádět všechny použité metody. Ukážeme, jak optimalizovat kód tak, aby překladač Matlabu vygeneroval co nejrychlejší kód.

Pro efektivní programování v Matlabu je velice vhodné používat funkce vestavěné funkce pro matice a vektory. Obecně platí, pokud data mají vektorovou nebo maticovou charakteristiku, je vhodné použít vestavěné funkce na práci s nimi. Neboť použitím těchto funkcí Matlab dokáže optimalizovat cílový kód. Překladač Matlabu dokáže generovat vektorové instrukce. Navíc většina funkcí Matlabu je navržena velice efektivně. Nahrazením části kódu vhodnými funkcemi Matlabu můžeme dosáhnout poměrně velkého zrychlení.

7.1 Předpodmínění

Mějme dán intervalový systém $\boldsymbol{A}x = \boldsymbol{b}$, kde $\boldsymbol{A} \in \mathbb{IR}^{m\times n}$, $\boldsymbol{b} \in \mathbb{IR}^{n}$. Pak předpomiňovací matici volíme jako přibližnou inverzi k

$$\begin{pmatrix} A^{c1} & 0 \\ A^{c2} & I_{m-n} \end{pmatrix}$$

kde $A^{c1} \in \mathbb{R}^{n\times n}$ obsahuje prvních n řádků matice A^c a $A^{c2} \in \mathbb{R}^{(m-n)\times n}$ zbytek matice A^c.

7.2 Iterační metody

Iterační metody fungují na iterativním zlepšování počátečního řešení. Iterativní zlepšování jsme vyjádřili pomocí vzorce pro $\boldsymbol{x}_i$. Jednotlivé složky v k-té iteraci Gauss-Seidelova jsme vyjádřili

$$\boldsymbol{x}_i^k = \frac{1}{\boldsymbol{a}_{ii}}\left(\boldsymbol{b}_i - \sum_{j=1}^{i-1} \boldsymbol{a}_{ij}\boldsymbol{x}_j^k - \sum_{j=i+1}^{n} \boldsymbol{a}_{ij}\boldsymbol{x}_j^{k-1}\right),$$

$$\boldsymbol{x}_i^k = \boldsymbol{x}_i^k \cap \boldsymbol{x}_i^{k-1}.$$

V praktické implementaci v Matlabu $\boldsymbol{x}_i^k$ počítáme

```
sloupce = [ 1:(i-1) (i+1):n ];
xk(i) = ( b(i) - A(i, sloupce) * xk(sloupce) ) / A(i,i).
```

Ještě předtím než začneme počítat $\boldsymbol{x}_i^k$, musíme nastavit $\boldsymbol{x}_i^k = \boldsymbol{x}_i^{k-1}$. Staré řešení si musíme také uchovávat, abychom potom mohli udělat průnik. Obdobně tento postup volíme i u Jacobiho metody.

U Gauss-Seidelovy metody děláme průnik řešení pokaždé, když aktualizujeme $\boldsymbol{x}_i^k$. U Jacobiho metody stačí dělat průnik řešení až po konci jedné iterace.

U iteračních metod je nastavený maximální počet iterací, maximální počet iterací lze měnit pomocí parametru funkce. Iterační metody svůj výpočet skončí, pokud jsme přesáhli maximální počet iterací, nebo platí

$$|\underline{x}^k - \underline{x}^{k-1}| < \epsilon \wedge |\overline{x}^k - \overline{x}^{k-1}| < \epsilon, \text{pro}\, \epsilon > 0.$$

7.3 Hansen–Bliek–Rohn, Ning-Kearfott

Pokud bychom použili metodu Hansen–Bliek–Rohn tak, jak jsme ji ukázali ve větě 17, nemuseli bychom dostat rigorózní výsledek. To je způsobeno zaokrouhlovacími chybami na počítači a také tím, že některé reálná čísla se nedají přesně reprezentovat na počítači (např. $0.1, 1/3$). Pokud bychom chtěli dostat rigorózní výsledek, tak bychom metodu museli poupravit. Buď bychom museli pracovat pouze s intervaly v průběhu výpočtu, nebo bychom museli různě přepínat zaokrouhlování v Matlabu.

Proto je jednodušší implementovat metodu Ning-Kearfott. Pokud všechny operace v této metodě budeme provádět pomocí intervalové aritmetiky, tak dostaneme rigorózní obálku.

7.4 Metoda využívající symetrie

Tato funkce je obecně popsána v 4.3.2, zde uvedeme pouze konkrétní specifikace naší implementace. Aby tato metoda fungovala rychle, je důležité efektivně vyhodnotit

$$\boldsymbol{z}_l = \sum_{j=1}^{n} C_{lj}(\tilde{\boldsymbol{b}}_j - \tilde{\boldsymbol{A}}_{jj}x_j^*) - \sum_{j=1}^{n}\sum_{i=1}^{j-1}(C_{li}x_j^* + C_{lj}x_i^*)\tilde{\boldsymbol{A}}_{ij}.$$

Vzorec rozdělíme na dvě části. První část $\sum_{j=1}^{n} C_{lj}(\tilde{\boldsymbol{b}}_j - \tilde{\boldsymbol{A}}_{jj}x_j^*)$ spočítáme pro všechny l jako $C(\boldsymbol{b} - \boldsymbol{d} \odot x^*)$, kde $\boldsymbol{d}$ je vektor sestavený s diagonálních prvků matice $\boldsymbol{A}$ a oprace $\odot$ je násobení vektorů po složkách. Druhou část spočítáme pro každé l jako $\hat{\boldsymbol{z}}$

$\hat{\boldsymbol{z}}_l := 0;$
for $j := 1, \ldots, n$ **do**
  $C_x := \left(C(l, 1:j-1) \cdot x_j^*\right)^T + C_{lj} \cdot x^*(1:j-1);$
  $\hat{\boldsymbol{z}}_l := \hat{\boldsymbol{z}}_l + \left\langle C_x, \boldsymbol{A}(1:j-1, j)\right\rangle;$
end for

kde $\langle \cdot, \cdot \rangle$ značí skalární součin dvou vektorů. Výpočet $\boldsymbol{z}$ jsme provedli podle ekvivalentního vzorce

$$\boldsymbol{z} = C(\boldsymbol{b} - \boldsymbol{d} \odot x^*) - \hat{\boldsymbol{z}},$$

kde

$$\hat{\boldsymbol{z}}_l = \sum_{j=1}^{n}\sum_{i=1}^{j-1}(C_{li}x_j^* + C_{lj}x_i^*)\tilde{\boldsymbol{A}}_{ij}.$$

Akorát $\hat{\boldsymbol{z}}_l$ jsme počítali efektivně, tak aby Matlab mohl kód optimalizovat.

Na nalezení obálky množiny řešení lineárního intervalového systému

$$C\tilde{\boldsymbol{A}}y = \boldsymbol{z}$$

používáme metodu Ning-Kearfott. Tuto metodu jsme volili, protože je rychlá a dává těsnou obálku.

7.5 Metoda nejmenších čtverců

Funkce **iolslsqenc** má jako poslední parametr možnost určit použitou metodu. Obecně je obtížné určit, která metoda vyhovuje danému parametru nejlépe. Metody jsme rozdělili do kategorií na základě numerického testování. Zkratky metod volíme stejně jako v sekci 5.2. Ve funkci **iolslsqenc** volíme metody na základě parametru **tradeoff** následovně:

- 'FASTEST': metoda **verifylss** řešící systém (5.1)
- 'FASTER': metoda **verifylss** řešící systém (5.2)
- 'EFFECTIVE': metoda Ning-Kearfott (**NK**) využívající alternativní systém (5.2)
- 'TIGHTER': metoda využívající symetrie (**sym**)
- 'TIGHTEST': metoda **intersect**

Závěr

V této práci jsme uvedli přehled metod zapouzdření všech řešení intervalových lineárních rovnic metodou nejmenších čtverců. Tento systém je tvořen symetrickou intervalovou maticí, proto jsou součástí práce také metody na symetrické soustavy. Výsledkem naší práce je také implementace těchto metod v prostředí Matlab za využití intervalové knihovny Intlab. Dále jsme numericky porovnali různé přístupy.

V prvních kapitolách jsme představili základy intervalové analýzy, jako například intervalovou aritmetiku, intervalové funkce a matice. Také jsme se věnovali intervalové lineární algebře. Pak naše pozornost směřovala k soustavám intervalových lineárních rovnic. Následně jsme prezentovali metody na nalezení obálky množiny řešení intervalových lineárních rovnic metodou nejmenších čtverců.

Poté jsme jednotlivé metody porovnali jak z hlediska rychlosti, tak z hlediska těsnosti obálek. Metody jsme porovnávali na náhodně vygenerovaných přeurčených intervalových systémech. Na konci porovnání jsme rozebrali výhody a nevýhody jednotlivých metod. Uvedli jsme případy, kdy je vhodné použít konkrétní metody.

Poslední kapitoly jsme věnovali přiloženému balíčku metod pro Matlab. Ukázali jsme jak tyto metody používat. Všechny metody jsme podrobně zdokumentovali v jednotlivých M-souborech. Nakonec jsme se zabývali implementačními detaily těchto metod.

Seznam použité literatury

Alefeld, G. a Mayer, G. (1993). The cholesky method for interval data. *Linear Algebra Appl.*, **194**, 161–182.

Barth, W. a Nuding, E. (1974). Optimale losung von intervallgleichungssystemen. *Comput.*, **12**, 117–125.

Beeck, H. (1973). Charakterisierung der losungsmenge von intervallgleichungssystemen. *Z. Angew. Math. Mech.*, **53**, T181–T182.

Bentbib, A. H. (2002). Solving the full rank interval least square problem. *Applied Numerical Mathematic*, **41**, 283–294.

Bliek, C. (1992). *Computer Methods for Design Automation.* PhD thesis, Massachusetts Institute of Technology.

Černý, M., Antoch, J. a Hladík, M. (2013). On the possibilistic approach to linear regression models involving uncertain, indeterminate or interval data. *Inf. Sci.*, **244**, 26–47. ISSN 0020-0255.

Hansen, E. R. (1992). Bounding the solution of interval linear equations. *SIAM J. Numer. Anal.*, **29(5)**, 1499–1503.

Hladík, M. (2016). Lineární algebra (nejen) pro informatiky. Technical report, Karlova Univerzita, Matematicko-fyzikální fakulta, Katedry aplikované matematiky.

Jansson, C. (1997). Calculation of exact bounds for the solution set of linear interval systems. *Linear Algebra Appl.*, **251**, 321–340.

Kreinovich, V. a Lakeyev, A. V. (1996). Linear interval equations: Computing enclosures with bounded relative or absolute overestimation is np-hard. *Reliab. Comput.*, **2(4)**, 341–350.

Matiyasevich, Y. V. (1970). Enumerable sets are diophantine. *Sov. Math., Dokl.*, **11(2)**, 354–357.

Moore, R. E. (1996). *Interval Analysis.* Prentice-Hall, Englewood Cliffs, NJ.

Moore, R. E., Kearfott, R. B. a Cloud, M. J. (2009). *Introduction to interval analysis.* SIAM. ISBN 978-0-898716-69-6.

Neumaier, A. (1990). *Interval Methods for Systems of Equations.* Cambridge University Press, Cambridge. ISBN 0-521-33196-X.

Ning, S. a Kearfott, R. B. (1997). A comparison of some methods for solving linear interval equations. *SIAM J. Numer. Anal.*, **34(4)**, 1289–1305.

Oettli, W. a Prager, W. (1964). Compatibility of approximate solution of linear equations with given error bounds for coefficients and right-hand sides. *Numer. Math.*, **6**, 405–409.

Rohn, J. (1993). Cheap and tight bounds: The recent result by E. Hansen can be made more efficient. *Interval Comput.*, **4**, 13–21.

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

Seznam tabulek