MATEMATICKO-FYZIKÁLNÍ FAKULTA
Univerzita Karlova

BAKALÁŘSKÁ PRÁCE

Pavel Vigilev

Předpodmínění a regularita symetrických intervalových matic

Katedra aplikované matematiky

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

Praha 2021

Prohlašuji, že jsem tuto bakalářskou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů. Tato práce nebyla využita k získání jiného nebo stejného titulu.

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

Děkuji především svému pánu vědoucímu doc. Mgr. Milanu Hladíkovi Ph.D za trpělivost, cenné rady, jež mi dodával v průběhu psaní této práce, a hlavně za čas, který mi věnoval. Bez něho by tato práce nevznikla. Chtěl bych také poděkovat všem profesorům Matematicko-fyzikální fakulty, kteří mě vyučovali, za jejich dobrou práci. Děkují také své rodině a přítelkyní za podporu po celou dobu mého studia.

Název práce: Předpodmínění a regularita symetrických intervalových matic

Autor: Pavel Vigilev

Katedra: Katedra aplikované matematiky

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

Abstrakt: V práci jsme se podívali na způsoby předpodmínění symetrických intervalových matic, použili jsme táto předpodmíněni pro popis a implementaci algoritmů testování regularity symetrických intervalových matic. Dále jsme porovnali efektivitu popsaných algoritmů. Podívali jsme se na metody odhadu vlastních čísel této třídy matic. Zkonstruovali jsme také metodu odhadu vlastních čísel, která používá testování regularity pro filtrování vstupního intervalu. Pak jsme numericky porovnali efektivitu těchto metod na různých typech matic. Implementovali jsme všechny algoritmy v MATLABu s využitím knihovny IntLab. Zjistili jsme, že testování postačující podmínky se standardním předpodmíněním je nejefektivnější metoda testování regularity ze všech implementovaných námi metod. Zkonstruována námi metoda odhadu vlastních čísel dává přesné výsledky díky své iterativitě, ale je mnohém pomalejší než jiné metody se stejnou přesností.

Klíčová slova: předpodmínění, intervalová analýza, intervalová matice, regularita intervalových matic, symetrická intervalová matice, Gaussova-Seidelova metoda

Title: Preconditioning and regularity of symmetric interval matrices

Author: Pavel Vigilev

Department: Department of Applied Mathematics

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

Abstract: In this thesis we observed different approaches to construct a preconditioner for symmetric interval matrices. Using these preconditioners we described and implemented methods for testing regularity of these matrices and compared efficiency of described algorithms. After that we observed different methods for estimating eigenvalues of this class of matrices. Also we constructed such method that uses any testing regularity method for filtering an input interval. After that we compared efficency of these methods on different classes of matrices. All algorithms we implemented using MATLAB with the IntLab library. Comparing numerical results we concluded that the way to test regularity of symmetric interval matrix based on the sufficient condition for regularity with standard preconditioner is the most efficient one among the algorithms we implemented. The constructed method for estimating eigenvalues based on testing regularity gives very accurate result because of its iterativity, but it seems to be very slow comparing to other methods which give similar accuracy.

Keywords: preconditioning, interval analysis, interval matrix, regularity of interval matrices, symmetric interval matrix, Gauss-Seidel method

Obsah

Úvod

Intervalová aritmetika je aritmetika definovaná na množině reálných intervalů místo množiny reálných čísel. Jinými slovy, v intervalové aritmetice veličina se reprezentuje ne jako konkretní číslo, ale jako interval ve kterém leží hodnota této veličiny. Reprezentace veličin pomocí intervalů je přirozená a dovoluje pracovat s intervaly i když používáme funkce nebo operátory. Například, v kapitole 1 jsou definovány základní operace jako sčítaní a násobení pro práci s intervaly. Jde definovat i složitější funkce, výsledkem aplikování kterých na interval dostaneme znovu interval, nebo nejakou jinou podmnožinu $\mathbb{R}$. Mnoho pojmů z lineární algebry, matematické analýzy a dalších oblasti matematiky jsou předefinovány pro práci s intervaly.

Intervalové metody čím dal tím častěji se používají v počítačové grafice, robotice, ekonomii.

Předpodmínění se široko používá pro numerické řešení lineárních systémů. Předpodmínění zmenšuje podmíněnost matice, což zvětšuje přesnost řešení lineárních systémů iterativními metodami. V této práci zajímají nás symetrické intervalové matice a jejich předpodmínění.

Cíl práce

Cílem této práce je prohlédnout různé způsoby předpodmínění symetrických intervalových matic v kontextu testování regularity. Dále uvést způsoby odhadu vlastních čísel symetrických intervalových matic a aplikovat testování regularity na odhadování vlastních čísel. Implementovat uvedené algoritmy v MATLABu s využitím knihovny IntLab a numericky porovnat uvedené metody vzhledem k jejich efektivitě.

Struktura práce

Na začátku práce zavedeme základní pojmy intervalové analýzy. Definujeme intervalovou matici, symetrickou intervalovou matici, řekneme, co je regulární intervalová matice a jak vypadají vlastní čísla intervalových matic. Podíváme se na Gaussovu-Seidelovu metodu a na její aplikací na parametrický intervalový systém.

V následujících dvou kapitolách se podíváme na vlastnosti intervalových matic a jejich vlastních čísel. Z těchto vlastnosti vznikají již popsané algoritmy, které uvedeme v těchto dvou kapitolách. Pak se podíváme na to, jak lze použit testování regularity na odhadování vlastních čísel intervalových matic.

Praktickou části této práce je balíček funkcí a tříd implementujících v systému MATLAB algoritmy z kapitol 2 a 3 s využitím knihovny IntLab. Dokumentace k praktické částí je uvedená jako komentář v zdrojovém kódu v balíčku a v kapitole 4. V kapitole 5 uvedeme výsledky numerického porovnání metod testování regularity a odhadu vlastních čísel pro různé druhy generovaných matic.

1. Intervalová analýza

1.1 Intervalová aritmetika

Definice 1 (Reálný interval)**.** *Nechť* $\underline{x}$,$\overline{x} \in \mathbb{R}$, $\underline{x} \leq \overline{x}$. *Reálný interval je množina* $\mathbf{x} = [\underline{x}, \overline{x}] = \{a | a \in \mathbb{R}, \underline{x} \leq a \leq \overline{x}\}$.

Zaveďme základní značení a definujme operaci pro práci s intervaly.
Značení:

- poloměr intervalu je $x^\Delta = \frac{\overline{x} - \underline{x}}{2}$
- střed intervalu je $x^c = \frac{\overline{x} + \underline{x}}{2}$
- magnituda intervalu je $mag(\mathbf{x}) = \max(|\underline{x}|, |\overline{x}|)$
- mignituda intervalu je

$$mig(\mathbf{x}) = \begin{cases} \min(|\underline{x}|, |\overline{x}|), & \text{pokud } 0 \notin \mathbf{x} \\ 0, & \text{pokud } 0 \in \mathbf{x} \end{cases} \tag{1.1}$$

Aritmetické operace pro reálné intervaly x, y jsou definované následujícím způsobem:

$$\mathbf{x} + \mathbf{y} = [\underline{x} + \underline{y}, \overline{x} + \overline{y}]$$
$$\mathbf{x} - \mathbf{y} = [\underline{x} - \overline{y}, \overline{x} - \underline{y}]$$
$$\mathbf{xy} = [\min(\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y}), \max(\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y})]$$
$$\mathbf{x}/\mathbf{y} = [\min(\underline{x}/\underline{y}, \underline{x}/\overline{y}, \overline{x}/\underline{y}, \overline{x}/\overline{y}), \max(\underline{x}/\underline{y}, \underline{x}/\overline{y}, \overline{x}/\underline{y}, \overline{x}/\overline{y})].$$

Dělení intervalu **x**,**y** t.ž $0 \in \mathbf{y}$ je definované jinak.

$\mathbf{a}$	$\mathbf{b} = 0$	$\underline{b} < \overline{b} = 0$	$\underline{b} < 0 < \overline{b}$	$0 = \underline{b} < \overline{b}$
$\overline{a} < 0$	$\emptyset$	$[\overline{a}/\underline{b}, \infty)$	$(-\infty, \overline{a}/\overline{b}] \cup [\overline{a}/\underline{b}, \infty)$	$(-\infty, \overline{a}/\overline{b}]$
$\underline{a} \leq 0 \leq \overline{a}$	$\mathbb{R}$	$\mathbb{R}$	$\mathbb{R}$	$\mathbb{R}$
$\underline{a} > 0$	$\emptyset$	$(-\infty, \underline{a}/\underline{b}]$	$(-\infty, \underline{a}/\underline{b}] \cup [\underline{a}/\overline{b}, \infty)$	$[\underline{a}/\overline{b}, \infty)$

Tabulka 1.1: Dělení intervalu $\mathbf{a}/\mathbf{b}$, kde $0 \in \mathbf{b}$

Důležité je, že množina všech reálných intervalů $\mathbb{IR}$ s operacemi výš netvoří těleso.

Formálně množina všech intervalu je definovaná jako $\mathbb{IR} = \{[a, b] | a,b \in \mathbb{R}, a \leq b\}$.

1.2 Intervalové matice a jejich regularita

Definice 2 (Intervalová matice)**.** *Intervalová matice* $\mathbf{A} \in \mathbb{IR}^{m\times n}$ *velikosti* $m \times n$ *je třída intervalových matic* $\mathbf{A} = [\underline{A}, \overline{A}] = \{A | A \in \mathbb{R}^{m\times n}, \underline{A} \leq A \leq \overline{A}\}$ *pro nějaká daná* $\underline{A}, \overline{A} \in \mathbb{R}^{m\times n}$.

Zde a dál "$\leq$" použitý na matice znamená méně nebo rovno po prvcích. Obdobně je pro "=". Další značení která potřebujeme jsou $\sqrt{D}$ a $|D|$ pro $D \in \mathbb{R}^{n\times n}$. Aplikujme ty operace na každý prvkek matice D: $(\sqrt{D})_{ij} = \sqrt{D_{ij}}$ a $|D|_{ij} = |D_{ij}|$.

Symetrické intervalové matice na rozdíl od reálných symetrických matic nestačí definovat jen pomocí rovnosti transponovaných matic. Pro intervalové matice mohou vznikat závislosti mezi prvky, proto symetrické intervalové matice jsou definované následujícím způsobem.

Definice 3 (Symetrická intervalová matice)**.** *Nechť* $\mathbf{A} \in \mathbb{IR}^{n\times n}$ *je intervalová matice pro nějaké* n*, pak symetrická intervalová matice je množina* $\mathbf{A}^S = \{A | A \in \mathbf{A}, A^T = A\}$.

Definice 4 (Regularita intervalové matice)**.** *Intervalová matice* $A \in \mathbb{IR}^{n\times n}$ *je regulární, pokud každá* $A \in \mathbf{A}$ *je regulární.*

Definice 5 (Spektrální poloměr)**.** *Nechť* $A \in \mathbb{C}^{n\times n}$ *je matice,* $\lambda_1, \ldots, \lambda_n$ *jsou vlastní čísla matice* A*. Pak* $\rho(A) = \max\{|\lambda_i| | i \in \{1, \ldots, n\}\}$ *je spektrální poloměr matice* A.

Podle ((Neumaier, 1991), věta 4.1.1) $\mathbf{A} \in \mathbb{IR}^{n\times n}$ je regulární, pokud platí

$$\rho(|(A^c)^{-1}|A^\Delta) < 1. \tag{1.2}$$

To je postačující podmínka regularity.

Definice 6 (Silně regulární matice)**.** *Intervalová matice* $\mathbf{A} \in \mathbb{IR}^{n\times n}$ *je silně regulární, pokud* $(A^c)^{-1}\mathbf{A}$ *je regulární.*

Podmínka (1.2) je ekvivalentní předchozí definici podle ((Neumaier, 1991), věta 4.1.1).

1.3 Vlastní čísla intervalových matic

Definice 7 (Množina vlastních čísel)**.** *Nechť* $\mathbf{A} \in \mathbb{IR}^{n\times n}$ *je symetrická intervalová matice. Množina vlastních čísel je množina*

$$\Lambda(\mathbf{A}^S) = \{\lambda \in \mathbb{R} | (\exists A \in \mathbf{A}^S)(\exists x \in \mathbb{R}^{n\times n})(x \neq 0 \wedge Ax = \lambda x)\}.$$

Množina $\Lambda(\mathbf{A}^S)$ je dobře definovaná, protože každá $A \in \mathbf{A}^S$ má pouze reálná vlastní čísla.

Nechť $A \in \mathbb{R}^{n\times n}$ je symetrická matice. A má n vlastních čísel, která jsou všechna reálná čísla, proto můžeme je uspořádat v nerostoucím pořadí:

$$\lambda_1(A) \geq \lambda_2(A) \geq \cdots \geq \lambda_n(A).$$

Definice 8 (i-té vlastní číslo symetrické intervalové matice))**.** *Nechť* $\mathbf{A} \in \mathbb{IR}^{n\times n}$. $\lambda_i(\mathbf{A}^S) = \{\lambda_i(A) | A \in \mathbf{A}^S\}$ *je i-té vlastní číslo symetrické intervalové matice, kde* $i \in \{1, \ldots, n\}$.

$\mathbf{A}^S$ obsahuje pouze symetrické matice, proto $\lambda_i(\mathbf{A}^S)$ je uzavřený interval například podle (Hladík a kol., 2010).

1.4 Gaussova-Seidelova metoda pro intervalové matice

Nechť $\mathbf{A} \in \mathbb{IR}^{m\times n}, \mathbf{b} \in \mathbb{IR}^{m}$. Intervalový lineární systém je třída $\{Ax = b | A \in \mathbf{A}, b \in \mathbf{b}\}$ lineárních systémů. Řešením takového intervalového systému je množina $\Sigma = \{x \in \mathbb{R}^n | (\exists A \in \mathbf{A})(\exists b \in \mathbf{b})(Ax = b)\}$.

Pro lineární intervalový systém $\mathbf{Ax} = \mathbf{b}$, $\mathbf{A} \in \mathbb{IR}^{n\times n}, \mathbf{x}, \mathbf{b} \in \mathbb{IR}^{n}$ lze použít iterativní metody. Gaussova-Seidelova metoda pro takovou soustavu je definována následujícím způsobem.

Máme nějakou obálku počátečního řešení $\mathbf{x}^0 \in \mathbb{IR}^n$. Pro $i = 1, \dots, n$

$$\mathbf{x}_i^k := \Big(\frac{1}{\mathbf{a}_{ii}}\Big(\mathbf{b}_i - \sum_{j=1}^{i-1} \mathbf{a}_{ij}\mathbf{x}_j^k - \sum_{j=i+1}^{n} \mathbf{a}_{ij}\mathbf{x}_j^{k-1}\Big)\Big) \cap \mathbf{x}_i^{k-1}. \tag{1.3}$$

V praktické části Gaussova-Seidelova metoda je námi implementovaná pro parametrický systém.

1.5 Parametrický intervalový systém a jeho řešení

Definice 9 (Parametrický systém)**.** *Nechť $A \in \mathbb{IR}^{n\times n}$, $b \in \mathbb{IR}^n$. Parametrický lineární intervalový systém je třída reálných lineárních systémů*

$$A(p)x = b(p),\ p \in \mathbf{p},$$

kde $\mathbf{p} \in \mathbb{IR}^K$ je daný intervalový vektor parametru. $A(p)$ a $b(p)$ lineárně závisí na parametrech $p_1, \dots, p_K$ s danými maticemi $A^1, \dots, A^K \in \mathbb{R}^{n\times n}$ a vektory $b^1, \dots, b^K \in \mathbb{R}^n$, t.j. platí

$$A(p) = \sum_{k=1}^{K} A^k p_k,\ b(p) = \sum_{k=1}^{K} b^k p_k.$$

Symetrickou intervalovou matici můžeme reprezentovat parametrickým systémem tímto způsobem pomocí $K = \frac{n^2+n}{2} + n$ parametru a sparse-matic $A^1, \dots,$ $A^{\frac{n^2-n}{2}}$ pro reprezentaci koeficienty kolem mimodiagonalových prvků, matic $A^{\frac{n^2-n}{2}+1}, \dots, A^{\frac{n^2+n}{2}}$ pro koeficienty prvků na diagonále a vektory $b^{\frac{n^2+n}{2}+1}, \dots,$ $b^{\frac{n^2+n}{2}+n}$ pro reprezentaci složky vektoru $b \in \mathbb{IR}^n$.

Řešením takového systému je množina

$$\Sigma_p = \{x \in \mathbb{R}^n | (\exists p \in \mathbf{p})(A(p)x = b(p))\}.$$

Definice 10 (Relaxace parametrického systému)**.** *Nechť $K \in \mathbb{N}$, $\mathbf{p} \in \mathbb{IR}^K$ je vektor parametrů a třída $\{A(p)x = b(p) \mid p \in \mathbf{p}\}$ je nějaký parametrický systém. Relaxovaný systém je intervalový lineární systém $A(\mathbf{p})\mathbf{x} = b(\mathbf{p})$*

V MATLABu tato reprezentace je námi implementována třídou `parametric_system` a funkcí `parametric_from_symmetric`.

1.6 Intervalová H-matice

Popišme krátce koncept H-matic, na kterém jsou založené některé námi implementované algoritmy v praktické části.

Definice 11 (Comparison matice)**.** *Nechť* $\mathbf{A} \in \mathbb{IR}^{n\times n}$*. Matice* $\langle A\rangle \in \mathbb{R}^{n\times n}$ *se nazývá comparison matice, pokud* $\langle A\rangle_{ii} = mig(\mathbf{A}_{ii})$ *pro* $i \in \{1,\dots,n\}$*,* $\langle A\rangle_{ik} = -mag(A_{ik})$*, pro* $i,k \in \{1,\dots,n\}, i \neq k$*.*

Definice 12 (H-matice, podle věty 3.7.3 v (Neumaier, 1991))**.** *Nechť* $\mathbf{A} \in \mathbb{IR}^{n\times n}$ *a* $\langle A\rangle$ *je comparison matice matici* $\mathbf{A}$*.* $\mathbf{A}$ *je H-matice, pokud* $\langle A\rangle$ *je regulární a* $\langle A\rangle^{-1}e > 0$ *pro* $e = (1,\dots,1)^T$*.*

Věta 1 (důsledek 4.3.10 (Neumaier, 1991))**.** *Pokud* $\mathbf{A} \in \mathbb{IR}^{n\times n}$ *není H-matice a každá iterace Gaussovy-Seidelovy metody (1.3) vrátí neprázdný interval* $\mathbf{x}$ *pro počáteční interval* $\mathbf{x}^0$*, potom platí* $\overline{x} = \overline{x^0}$ *nebo* $\underline{x} = \underline{x^0}$*.*

Kontrapozitivně táto věta říká, že pokud Gaussova-Seidelova metoda pro intervalový lineární systém $\mathbf{Ax}^0 = 0$ vrátí neprázdný topologický vnitřek $\mathbf{x}$ počátečního boxu $\mathbf{x}^0$ pro každou iteraci, pak intervalová matice $\mathbf{A}$ je H-matice. Ve skutečnosti to znamená, že taková matice je regulární, o čemž říká následující věta.

Věta 2 (Regularita H-matic, věta 3.7.5 (Neumaier, 1991))**.** *Každá H-matice* $\mathbf{A} \in \mathbb{IR}^{n\times n}$ *je regulární.*

Následující věta uvede vztah mezí dvoustranným předpodmíněním a regularitou.

Věta 3 (Regularita předpodmíněné H-matice, veta 4.1.2 (Neumaier, 1991))**.** *Nechť* $\mathbf{A} \in \mathbb{IR}^{n\times n}$*,* $C, C' \in \mathbb{R}^{n\times n}$ *a matice* $C\mathbf{A}C'$ *je H-matice. Pak* $\mathbf{A}$ *je silně regulární.*

2. Algoritmy testující regularitu

V této kapitole se podíváme na metody testování regularity.

2.1 Postačující podmínka a standardní předpodmínění

Předpodmínění intervalových matic pochází z Hansen (1965); Hansen a Smith (1967), kde se používá pro řešení lineárních intervalových systémů. V této sekci se podíváme na předpodmínění maticí inverzní k středové.

Podle ((Neumaier, 1991), tvrzení 4.1.1) pokud $(A^c)^{-1}\mathbf{A}$ je regulární, pak $\mathbf{A}$ je regulární. Označme $\mathbf{B} = (A^c)^{-1}\mathbf{A}$, tedy $\rho(|(B^c)^{-1}|B^\Delta) = \rho(|((A^c)^{-1}A^c)^{-1}|B^\Delta) = \rho(|I^{-1}|B^\Delta) = \rho(B^\Delta)$. $\mathbf{B}$ je regulární, pokud $\rho(B^\Delta) < 1$.

V MATLABu toto předpodmínění jsme implementovali přes předpodmínění s zachováním lineárních závislostí pomocí reprezentace intervalové matice $\mathbf{A}$ jako parametrického systému $\{A(p)x = 0 \mid p \in \mathbf{p}\}$ pro nějaké $\mathbf{p} \in \mathbb{IR}^K$ a $K \in \mathbb{N}$ s využitím třídy `parametric_system`.

Asymptotická složitost takového testování je $O(n^3 + n^2) = O(n^3)$, protože hledání inverze A^c má složitost $O(n^3)$ a násobení dvou matic je $O(n^3)$.

2.2 Postačující podmínka a LDL-rozklad

Předpodmínění v tomto a následujícím algoritmu jsou dvoustranná a pracujeme se symetrickými maticemi, proto potřebujeme zachovat všechny závislosti. Obecně předpodmínění symetrické intervalové matice z dvou stran je definované následujícím způsobem.

Nechť $\mathbf{A} \in \mathbb{IR}^{n\times n}$, $T, P \in \mathbb{R}^{n\times n}$. Označme $\mathbf{B} = T\mathbf{A}^S P$.

$$\begin{aligned} \mathbf{B}_{ij} = T_{i*}(\mathbf{A}^S P)_{*j} = \sum_{k=1}^{n} T_{ik}(\mathbf{A}^S P)_{kj} = \\ = \sum_{t=1}^{n} \Big(T_{it}\mathbf{A}_{tt}P_{tj} + \sum_{k=1}^{t-1} \mathbf{A}_{tk}(T_{it}P_{kj} + T_{ik}P_{tj}) \Big). \end{aligned} \tag{2.1}$$

Asymptotická složitost spočítaní tohoto výrazu je $O(n^2)$. Prvků je n^2, proto celou operaci předpodmínění lze udělat za $O(n^4)$.

Předpodmíníme takovým způsobem symetrickou matici $\mathbf{A}^S$ maticemi $T = (LD)^{-1}$ a $P = (L^T)^{-1}$, kde L a D jsou matice z LDL-rozkladu matice A^c. Předpodmíněnou matici označme $\mathbf{B}$ a B^c tedy je jednotková matice.

Použitím kriteria regularity (1.2) dostaneme, že $\mathbf{A}^S$ je regulární,
pokud $\rho(B^\Delta) < 1$.

Podobně platí i pro předpodmínění $\mathbf{B} = P\mathbf{A}^S P^T$, kde $P = (L\sqrt{|D|})^{-1} = \sqrt{|D|}^{-1} L^{-1}$. Tedy

$$P^T = ((L\sqrt{|D|})^{-1})^T = (\sqrt{|D|}^T L^T)^{-1} = (\sqrt{|D|} L^T)^{-1} = (L^T)^{-1}\sqrt{|D|}^{-1}$$

Pro středovou matici B^c platí:

$$B^c = PA^cP^T = \sqrt{|D|}^{-1} L^{-1}LDL^T(L^T)^{-1}\sqrt{|D|}^{-1} = I_n \tag{2.2}$$

Asymptotická složitost LDL-rozkladu je $O(n^3)$, složitost testování regularity uvedeným způsobem je $O(n^4)$.

2.3 Postačující podmínka a Choleského rozklad

Kdyby v (2.2) matice A^c byla positivně definitní, diagonální matice z LDL-rozkladu by byla jednotkovou matici a dostali bychom Choleského rozklad, asymptotická složitost kterého je také $O(n^3)$.

Nechť znovu máme symetrickou intervalovou matici $\mathbf{A}^S$ velikosti $n \times n$ a nechť L je dolní trojúhelníková matice z Choleského rozkladu matici A^c.

Stejně jako v (2.2) předpodmíněním matice $\mathbf{A}^S$ dostaneme $\mathbf{B} = L^{-1}\mathbf{A}^S(L^T)^{-1}$. Podmínka regularity je $\rho(B^\Delta) < 1$.

Asymptotická složitost je znovu $O(n^4)$.

2.4 Gaussova-Seidelova metoda pro $C\mathbf{A}\mathbf{x} = 0$ a $\mathbf{x}^0 = [-1,1]^n$

Nechť $\mathbf{A} \in \mathbb{IR}^{n\times n}$, $\mathbf{x}^0 = [-1,1]^n$ a nechť A^c je regulární. Uvažme lineární systém $\mathbf{A}^S\mathbf{x} = 0$. Standardním předpodmíněním $C = (A^c)^{-1}$ dostaneme systém $C\mathbf{A}^S\mathbf{x} = 0$. Podle věty 1 pokud iterace Gaussovy-Seidelovy metody vrátí neprázdný topologický vnitřek počátečního boxu $\mathbf{x}^0$, pak matice $C\mathbf{A}$ je regulární a tedy i $\mathbf{A}^S$ je regulární.

Testování regularity touto metodou jsme implementovali v MATLABu s využitím knihovny IntLab funkcí `sym_regular_gs`. Symetrická matice je reprezentovaná jako parametrický systém pomocí třídy `parametric_system`. Konstrukci parametrického systému provádí funkce `parametric_from_symmetric`.

Spočítejme asymptotickou složitost této metody testování regularity. Výpočet jednotlivé i-té složky výsledného intervalového vektoru $\mathbf{x}$ v jediné iteraci je

$$\mathbf{z}_i \leftarrow \frac{1}{\sum_{k=1}^{K}(CA^k)_{ii}\mathbf{p}_k}\left(\sum_{k=1}^{K}(Cb^k)_i\mathbf{p}_k - \sum_{j\neq i}\left(\sum_{k=1}^{K}(CA^k)_{ij}\mathbf{p}_k\right)\mathbf{x}_j\right) \tag{2.3}$$

$$\mathbf{x}_i \leftarrow \mathbf{x}_i \cap \mathbf{z}_i.$$

Takový výpočet bychom měli provést n-krát, tedy počet operací je

$$T_{gs} = n(T_{denom} + T_{left} + T_{right} + T_\cap),$$

kde

- T_{denom} je počet operací počítaní jmenovatelů v Gaussove-Seidelove metodě t.j. $\sum_{k=1}^{K}(CA^k)_{ii}\mathbf{p}_k$, což trvá $O(K)$ kvůli tomu, že násobíme sparse-matici A^k s 1 až 2 nenulovými prvky v konstantním čase;

- T_{left} je počet operací počítaní levé části výrazu v Gaussove-Seidelove metodě t.j. $\sum_{k=1}^{K}(Cb^k)_i\mathbf{p}_k$, což trvá $O(Kn)$;
- T_{right} je počet operací počítaní pravé části výrazu v Gaussove-Seidelove metodě t.j. $\sum_{j\neq i}(\sum_{k=1}^{K}(CA^k)_{ij}\mathbf{p}_k)\mathbf{x}_j$, trvá $O(nK)$, znovu protože A^k jsou sparse-matice pro každé k;
- $T_\cap$ je složitost průniku dvou intervalu a je $O(1)$.

Celkem počet operací testování regularity timto způsobem:

$$T = T_{gs} + T_{check} = n(T_{denom} + T_{left} + T_{right} + T_\cap) + T_{check};$$

zde $T_{check} = O(n)$ je počet operací kontroly zúžení výsledného vektoru ve všech směrech.

Po substituci $K = \frac{n^2+n}{2}$ dostaneme: $T_{denom} \in O(n^2)$, $T_{left} \in O(n^3)$, $T_{right} \in O(n^3)$. Tím pádem $T \in O(n^4)$.

2.5 Gaussova-Seidelova metoda pro $\mathbf{A}\mathbf{x} = 0$ a $\mathbf{x}^0 = [-1,1]^n$ s optimálním předpodmíněním

Optimální předpodmínění neparametrického intervalového lineárního systému pro Gaussovu-Seidelovu metodu bylo zkoumáno u (Kearfott a kol., 1991; Kearfott, 1996). Případ pro parametrický intervalový systém byl zkoumán u (Hladík, 2016). V této sekci aplikujme optimální předpodmínění parametrického lineárního intervalového systému na testování regularity symetrické intervalové matice.

Podle (Hladík, 2016), lze předpodmínit systém tak, aby výsledek iterace Gaussovy-Seidelovy metody byl optimální podle nějakého parametru. Máme tři parametry, které můžeme optimalizovat: šířku, infimum a supremum výsledného intervalu, a každý z nich lze optimalizovat v každé iteraci řešením lineárního programu.

Označme j-tý sloupec matice A jako A_{*j}, i-tý řádek jako A_{i*}. Všimneme si, že výraz i-té iterace v (2.3) záleží jen na i-tém řádku C. Označme $c = C_{i*}$. Normalizujme jmenovatel tak, aby se rovnal intervalu $[1, r]$ pro nějaké $r \geq 1$ a přepišme (2.3) jako

$$\mathbf{z}_i \leftarrow \left(\sum_{k=1}^{K}(cb^k)_i\mathbf{p}_k - \sum_{j\neq i}\left(\sum_{k=1}^{K}(cA_{*j}^k)\mathbf{p}_k\right)\mathbf{x}_j\right) \tag{2.4}$$
$$\mathbf{x}_i \leftarrow \mathbf{x}_i \cap \mathbf{z}_i.$$

Uveďme následující značení:

$$\beta_k = |cb^k|,\ k = 1,\ldots,K$$
$$\alpha_{jk} = |cA_{*j}^k|,\ j = 1,\ldots,n, k = 1,\ldots,K$$
$$\eta_j = \overline{\Big(\sum_{k=1}^{K}(cA_{*j}^k)\mathbf{p}_k\Big)\mathbf{x}_j},\ j \neq i$$
$$\psi_j = \underline{\Big(\sum_{k=1}^{K}(cA_{*j}^k)\mathbf{p}_k\Big)\mathbf{x}_j},\ j \neq i$$

2.5.1 Optimalizace délky intervalu

Pro optimalizaci délky intervalu $\mathbf{z}_i$ potřebujeme řešit lineární program s $Kn + K + 3n - 2$ proměnných a $2Kn + 2K + 4n - 3$ podmínek.

Účelová funkce je

$$\min \sum_{k=1}^{K} 2p_k^{\Delta}\beta_k + \sum_{j \neq i} \eta_j - \psi_j.$$

Podmínky

$$\beta_k \geq cb^k, \beta_k \geq -cb^k, k = 1, \ldots, K \tag{2.5}$$

lze zkonstruovat za $O(K + n)$;

$$\alpha_{jk} \geq cA_{*j}^k, \alpha_{jk} \geq -cA_{*j}^k, j = 1, \ldots, n, k = 1, \ldots, K \tag{2.6}$$

lze zkonstruovat za $O(Kn)$;

$$\eta_j \geq c\sum_{k=1}^{K} A_{*j}p_k^c\underline{x}_j - \sum_{k=1}^{K} p_k^{\Delta}\underline{x}_j\alpha_{jk} \tag{2.7}$$

$$\eta_j \geq c\sum_{k=1}^{K} A_{*j}p_k^c\overline{x}_j + \sum_{k=1}^{K} p_k^{\Delta}\overline{x}_j\alpha_{jk} \tag{2.8}$$

$$\psi_j \leq c\sum_{k=1}^{K} A_{*j}p_k^c\underline{x}_j + \sum_{k=1}^{K} p_k^{\Delta}\underline{x}_j\alpha_{jk} \tag{2.9}$$

$$\psi_j \leq c\sum_{k=1}^{K} A_{*j}p_k^c\overline{x}_j - \sum_{k=1}^{K} p_k^{\Delta}\overline{x}_j\alpha_{jk} \tag{2.10}$$

$$c\sum_{k=1}^{K} A_{*i}^k p_k^c - \sum_{k=1}^{K} p_k^{\Delta}\alpha_{ik} = 1 \tag{2.11}$$

lze zkonstruovat za $O(Kn^2)$ každou. Podmínka (2.11) normalizuje jmenovatel pravé strany (2.3) tak, aby se rovnal $[1, r]$ pro nějaké $r \in \mathbb{R}, r \geq 1$.

2.5.2 Minimalizace suprema

Minimalizaci suprema intervalu $\mathbf{z}_i$ lze udělat řešením lineárního programu s účelovou funkci

$$\min c\sum_{k=1}^{K} b^k p_k^c + \sum_{k=1}^{K} p_k^{\Delta}\beta_k - \sum_{j \neq i} \psi_j.$$

Lze tuto účelovou funkci zkonstruovat za $O(Kn)$.

Použijme podmínky (2.5), (2.6), (2.9), (2.10), (2.11) a přidejme podmínky

$$\psi_j \leq c\sum_{k=1}^{K} A_{*j}p_k^c\underline{x}_j - \sum_{k=1}^{K} p_k^{\Delta}\underline{x}_j\alpha_{jk} \tag{2.12}$$

$$\psi_j \leq c\sum_{k=1}^{K} A_{*j}p_k^c\overline{x}_j + \sum_{k=1}^{K} p_k^{\Delta}\overline{x}_j\alpha_{jk} \tag{2.13}$$

2.5.3 Maximalizace infima

Účelová funkce je

$$\max c \sum_{k=1}^{K} b^k p_k^c - \sum_{k=1}^{K} p_k^{\Delta} \beta_k - \sum_{j \neq i} \eta_j$$

Podmínky jsou (2.5), (2.6), (2.7), (2.8), (2.11) a také

$$\eta_j \geq c \sum_{k=1}^{K} A_{*j} p_k^c \underline{x}_j + \sum_{k=1}^{K} p_k^{\Delta} \underline{x}_j \alpha_{jk} \tag{2.14}$$

$$\eta_j \geq c \sum_{k=1}^{K} A_{*j} p_k^c \overline{x}_j - \sum_{k=1}^{K} p_k^{\Delta} \overline{x}_j \alpha_{jk} \tag{2.15}$$

3. Odhad vlastních čísel

3.1 Základní odhad vlastního čísla

Věta 4 (Rohnova věta, (Hladík a kol., 2010), 3.1)**.** *Pro symetrické intervalové matice* $\mathbf{A}^S \in \mathbb{IR}^{n \times n}$ *platí*

$$\lambda_i(\mathbf{A}^S) \subseteq [\lambda_i(A^c) - \rho(A^\Delta), \lambda_i(A^c) + \rho(A^\Delta)]. \tag{3.1}$$

Důkaz je například v (Hladík a kol., 2010).

Tento odhad vlastního čísla jsme implementovali v MATLABu funkcí `basic_bounds`. Složitost této metody odhadu je $O(n^4)$, protože spočítaní n vlastních čísel reálné matice A^c má asymptotickou složitost $O(n^4)$.

3.2 Odhad vlastních čísel pomocí přímé a nepřímé interlace metody

Definice 13. *Nechť* $\mathbf{A} \in \mathbb{IR}^{n \times n}$*. Pro každé* $i \in \{1, \ldots, n\}$ *definujme* $\mathbf{A}_i$ *vyškrtnutím i-tého řádku a i-tého sloupce z* $\mathbf{A}$*. Obdobně označme* A_i *pro* $A \in \mathbb{R}^{n \times n}$*.*

Věta 5 (Interlace vlastnost)**.** *Pro reálnou symetrickou matici* $A \in \mathbb{R}^{n \times n}$ *a* $i \in \{1, \ldots, n\}$ *platí*

$$\lambda_1(A) \geq \lambda_1(A_i) \geq \lambda_2(A) \geq \lambda_2(A_i) \geq \cdots \geq \lambda_{n-1}(A_i) \geq \lambda_n(A). \tag{3.2}$$

Tvrzení 1. *Pro symetrickou intervalovou matici* $\mathbf{A} \in \mathbb{IR}^{n \times n}$ *platí*

$$\overline{\lambda}_1(\mathbf{A}^S) \leq \lambda_1(mag(\mathbf{A})). \tag{3.3}$$

Důkaz je uveden v (Hladík a kol., 2010), ale uveďme ho také v této práci.

Důkaz. Podle Courantovy-Fischerovy věty platí pro každou $A \in \mathbf{A}^S$: $\lambda_1(A) = \max\{x^T A x | x \in \mathbb{R}^n\}$. Pak platí:

$$\begin{aligned}
\overline{\lambda}_1(A) &= \max\{x^T A x | x \in \mathbb{R}^n, x^T x = 1\} \\
&\leq \max\{|x^T A x| | x \in \mathbb{R}^n, x^T x = 1\} \\
&\leq \max\{|x^T||A||x| | x \in \mathbb{R}^n, x^T x = 1\} \\
&\leq \max\{|x^T| mag(\mathbf{A})|x| | x \in \mathbb{R}^n, x^T x = 1\} \\
&\leq \max\{x^T mag(\mathbf{A}) x | x \in \mathbb{R}^n, x^T x = 1\} = \lambda_1(mag(\mathbf{A}))
\end{aligned} \tag{3.4}$$

□

Definice 14. *Nechť* $\mathbf{A} \in \mathbb{IR}^{n \times n}$*. Horní odhad i-tého vlastního čísla symetrické matice* $\mathbf{A}^S$ *je libovolné* $\lambda_i^u(\mathbf{A}^S) \in \mathbb{R}$ *takové, že* $\lambda_i^u(\mathbf{A}^S) \geq \overline{\lambda_i}(\mathbf{A}^S)$*.*

Na základe této vlastnosti v (Hladík a kol., 2010) jsou rozvíjeny následující dva algoritmy.

Algoritmus 1 Přímá interlace metoda

Require: Symetrická intervalová matice $\mathbf{A}^S$

1: $\mathbf{B}^S \leftarrow \mathbf{A}^S$
2: **for** $k = 1, \ldots, n$ **do**
3: $\quad \lambda_1^u(\mathbf{B}^S) \leftarrow \min\{\lambda_i(B^c) + \rho(B^\Delta), \lambda_1(mag(\mathbf{B}))\}$
4: $\quad \lambda_k^u(\mathbf{A}^S) \leftarrow \lambda_1^u(\mathbf{B}^S)$
5: $\quad i \leftarrow \arg\min_{j=1,\ldots,n-k+1} \lambda_1^u(\mathbf{B}_j^S)$
6: $\quad \mathbf{B}^S \leftarrow \mathbf{B}_i^S$
7: **end for**
8: $I \leftarrow \emptyset$
9: **for** $k = 1, \ldots, n$ **do**
10: $\quad i \leftarrow \arg\min_{j\in\{1,\ldots,n\}\setminus I} \lambda_1^u(\mathbf{B}_j^S)$
11: $\quad I \leftarrow I \cup \{i\}$
12: $\quad \lambda_1^u(\mathbf{B}^S) \leftarrow \min\{\lambda_i(B^c) + \rho(B^\Delta), \lambda_1(mag(\mathbf{B}))\}$
13: $\quad \lambda_{n-k+1}^u(\mathbf{A}^S) \leftarrow \min\{\lambda_{n-k+1}^u(\mathbf{A}^S), \lambda_1^u(\mathbf{B}^S)\}$
14: **end for**
15: **return** $\lambda_k^u(\mathbf{A}^S), k = 1, \ldots, n$

V první iteraci prvního cyklu v 3. a 4. krocích algoritmus odhadne $\lambda_1^u(\mathbf{A}^S)$. Pak v k-té iteraci již máme minor $\mathbf{B}^S$. Ten minor algoritmus zkonstruoval postupným výběrem nějakého indexu v prvních $k-1$ iteracích. Označme posloupnost těchto indexů $P = (i_1, \ldots, i_{k-1})$ takovou, že na začátku k-té iterace platí

$$\mathbf{B}^S = ((\ldots(\mathbf{A}_{i_1}^S)_{i_2}\ldots)_{i_{k-2}})_{i_{k-1}}.$$

Platí tedy podle (3.2)

$$\begin{aligned}\lambda_1^u(\mathbf{B}^S) = \lambda_1^u(((\ldots(\mathbf{A}_{i_1}^S)_{i_2}\ldots)_{i_{k-2}})_{i_{k-1}}) \geq \lambda_2^u((\ldots(\mathbf{A}_{i_1}^S)_{i_2}\ldots)_{i_{k-2}}) \geq \ldots \\ \cdots \geq \lambda_{k-2}^u((\mathbf{A}_{i_1}^S)_{i_2}) \geq \lambda_{k-1}^u(\mathbf{A}_{i_1}^S) \geq \overline{\lambda}_{k-1}(\mathbf{A}_{i_1}^S) \geq \overline{\lambda}_k(\mathbf{A}^S).\end{aligned} \tag{3.5}$$

Pak v k-té iteraci v 5. a 6. krocích algoritmus volí takový index i_{k+1}, aby v další iteraci $\lambda_1^u(\mathbf{B}^S)$ byl co nejmenší, což zaručuje přesnost odhadu. Ukázali jsme tím korektnost algoritmu.

Druhý cyklus algoritmu dělá to samé, ale v jiném pořadí. Konstruuje původní matici z nějakého prvku na diagonále přidáním nějakého řádku a sloupce z původní matici. Ten postup dává přesnější nebo stejný odhad, protože v 13. kroku algoritmus vybírá nejmenší horní odhad.

Pro zvolení i v (5) a (10) lze využít

$$i := \arg\min_{j=1,\ldots,n-k+1} \sum_{r,x\neq j} |\mathbf{B}_{rs}|^2. \tag{3.6}$$

Nepřímá verze algoritmu používá následující větu.

Věta 6 (Weylova věta)**.** *Nechť $A,B \in \mathbb{R}^{n\times n}$ jsou symetrické matice. Pak platí*

$$\forall r,s \in \{1,\ldots,n\}, r+s \leq n+1 : \lambda_{r+s-1}(A+B) \leq \lambda_r(A) + \lambda_s(B)$$

a

$$\forall r,s \in \{1,\ldots,n\}, r+s \geq n+1 : \lambda_{r+s-n}(A+B) \geq \lambda_r(A) + \lambda_s(B).$$

Algoritmus 2 Nepřímá interlace metoda

Require: Symetrická intervalová matice $\mathbf{A}^S$

1: Spočítáme $\lambda_1(A^c) \geq \cdots \geq \lambda_n(A^c)$
2: Pomocí (1) spočítejme $\lambda_1^u([-A^\Delta, A^\Delta]), \ldots, \lambda_n^u([-A^\Delta, A^\Delta])$
3: **for** $k = 1, \ldots, n$ **do**
4: $\quad \lambda_k^u(\mathbf{A}^S) \leftarrow \min_{i=1,\ldots,k} \lambda_i(A^c) + \lambda_{k-i+1}^u([-A^\Delta, A^\Delta])$
5: **end for**
6: **return** $\lambda_k^u(\mathbf{A}^S), k = 1, \ldots, n$

V MATLABu jsme naimplementovali funkce `interlace_direct` a `interlace_indirect` na základe uvedených výš algoritmů. Tyto funkce aplikují algoritmy 1 a 2 na vstupní symetrickou intervalovou matici $\mathbf{A}^S$ a na $-\mathbf{A}^S$, čímž dostáváme odhad všech vlastních čísel matice $\mathbf{A}^S$.

3.3 Filtrovací metoda

Nechť $\mathbf{A}^S \in \mathbb{IR}^{n\times n}$ je symetrická intervalová matice. Podle (Hladík a kol., 2011) platí následující věta.

Věta 7 (Hlavní věta u (Hladík a kol., 2011))**.** *Nechť $\mathbf{A} \in \mathbb{IR}^{n\times n}$, $\lambda^0 \notin \Lambda(\mathbf{A})$ a označme $\mathbf{M} := \mathbf{A} - \lambda^0 I_n$. Pak $(\lambda^0 + \lambda) \notin \Lambda(\mathbf{A})$ pro každé $\lambda \in \mathbb{R}$ takové, že*

$$|\lambda| < \frac{1 - \frac{1}{2}\rho\Big(|I - QM^c| + |I - QM^c|^T + |Q|M^\Delta + (M^\Delta)^T|Q|^T\Big)}{\frac{1}{2}\rho\Big(|Q| + |Q|^T\Big)}, \tag{3.7}$$

kde $Q \in \mathbb{R}^{n\times n}, Q \neq 0$.

Důsledek 1. *Nechť $\mathbf{A} \in \mathbb{IR}^{n\times n}$, $\lambda^0 \notin \Lambda(\mathbf{A})^{\mathbf{S}}$ a označme $\mathbf{M}^S := \mathbf{A}^S - \lambda^0 I_n$. Pak $(\lambda^0 + \lambda) \notin \Lambda(\mathbf{A}^S)$ pro každé $\lambda \in \mathbb{R}$ takové, že*

$$|\lambda| < \frac{1 - \frac{1}{2}\rho\Big(|I - QM^c| + |I - M^cQ| + |Q|M^\Delta + M^\Delta|Q|\Big)}{\rho\Big(|Q|\Big)} \tag{3.8}$$

kde $Q \in \mathbb{R}^{n\times n}, Q \neq 0$.

Na této vlastnosti je založena metoda odhadu vlastních čísel, která je implementována námi funkcí `filter_interval_sym` v praktické částí této práce. Podobně jako v předchozí sekci, `filter_interval_sym` spustí následující algoritmus na $\mathbf{A}^S$ a na $-\mathbf{A}^S$, čímž dostáváme odhad celého vlastního čísla.

Uveďme pseudokód teto metody.

Algoritmus 3 Filtrovací metoda z (Hladík a kol., 2011)

$\mathbf{b} \leftarrow \mathbf{a}$
$t \leftarrow 0$
$\lambda \leftarrow \varepsilon b^\Delta + 1$
while $\lambda > \varepsilon b^\Delta$ and $t < T$ **do**
    $t \leftarrow t + 1$
    $\mathbf{M} \leftarrow \mathbf{A} - \overline{b} I_n$
    $Q \leftarrow (M^c)^{-1}$
    $\lambda \leftarrow \frac{1 - \frac{1}{2}\rho(|I - QM^c| + |I - M^c Q| + |Q|M^\Delta + M^\Delta|Q|}{\rho(|Q|)}$
    if $\lambda > 0$ **then**
        $\overline{b} \leftarrow \overline{b} - \lambda$
    end if
    if $\overline{b} < \underline{b}$ **then**
        $\mathbf{b} \leftarrow \emptyset$
    end if
end while
return $\mathbf{b}$

3.4 Filtrování pomocí testování regularity

Nechť $\mathbf{A} \in \mathbb{IR}^{n \times n}$ a uvažme symetrickou matici $\mathbf{A}^S$. Podle 3.1 uvažme

$$\lambda_i^*(\mathbf{A}^S) = [\lambda_i(A^c) - \rho(A^\Delta), \lambda_i(A^c) + \rho(A^\Delta)].$$

Rozčleňme $\lambda_i^*(\mathbf{A}^S)$ na $N \in \mathbb{N}$ intervalu a označme j-tý podinterval $\lambda_i^j(\mathbf{A}^S)$.

Pokud matice $\mathbf{A}^S - \lambda_i^j(\mathbf{A}^S) I_n$ je regulární, tak jistě neobsahuje žádné vlastní číslo žádné matice z $\mathbf{A}^S$ protože podle definice regularity platí

$$(\forall T \in \mathbf{A}^S - \lambda_i^j(\mathbf{A}^S) I_n)(\det T \neq 0).$$

Jinými slovy $(\forall A \in \mathbf{A}^S)(\forall \lambda \in \lambda_i^j(\mathbf{A}^S))(\det(A - \lambda I_n) \neq 0)$.

Na této vlastnosti je založená implementace metody `filter_sym_regularity` v praktické častí v MATLABu. Zřejmě čím je větší N, tím přesnější je odhad, ale je to i pomalejší. Asymptotická složitost takové metody odhadu vlastních čísel je $O(NT')$, kde T' je asymptotická složitost zvolené metody testování regularity.

4. Implementace v MATLABu a IntLabu

Praktická část této práce je implementace algoritmů uvedených v kapitolách 2 a 3 a jejich numerická analýza v MATLABu s použitím knihovny IntLab (Rump, 1999) pro intervalovou aritmetiku. V této kapitole uvedeme krátkou dokumentaci.

Program v MATLABu se skládá z následujících modulů:

- třída `parametric_system` representuje v paměti parametrický intervalový systém
- `gauss_seidel` poskytuje funkce pro řešení parametrických intervalových systémů Gaussovou-Seidelovou metodou
- `optimized_preconditioner` poskytuje funkce pro hledání optimální předpodmíňovací matice a její využíti v Gaussove-Seidelove metodě
- `regularity` poskytuje funkce pro testování regularity symetrických intervalových matic
- `eigenvalue_estimating` poskytuje funkce pro hledání intervalů vlastních čísel.
- `Random_matrix_generators` je modul s třídami pro generování různých typu reálných a intervalových matic
- `analysis` je modul se skripty pro generování experimentálních dat

4.1 Modul `parametric_system`

Táto třída poskytuje funkce pro práci s parametrickými intervalovými systémy a representuje parametrický intervalový systém v paměti.

- `P` vektor parametrů intervalového systému
- `n` velikost čtvercové matice
- `K` délka vektoru P
- `A0` počáteční matice levé strany nevyskytující se v parametrizaci vektorem P
- `b0` počáteční vektor pravé strany nevyskytující se v parametrizaci vektorem P
- `As` cell-vektor (cell-array) sparse-matic pro parametrizaci
- `bs` matice velikosti $n \times K$, kde k-tý sloupec je parametrizován k-tým parametrem

Metoda `relax_paramteric_system`

Instanční metoda třídy `parametric_system` vracející relaxovaný lineární systém: matice levé strany bez lineárních závislostí a vektor pravé strany.

Metoda `paramteric_system_instance(P)`

Instanční metoda třídy `parametric_system`. Nechť `C` je instance třídy `parametric_system` reprezentující lineární parametrický systém $\{A(p)x = b(p) | p \in \mathbf{p}\}$ pro nějaké $\mathbf{p} \in \mathbb{IR}^K$, $K \in \mathbb{N}$. `paramteric_system_instance(p)` vrátí $A(p)$ a $b(p)$ pro nějaké $p \in \mathbf{p}$.

Funkce `parametric_from_symmetric(sym, b)`

Funkce konstruující instance třídy `parametric_system` ze symetrické intervalové matice `sym` a intervalového vektoru pravé strany `b`.

4.2 Modul `gauss_seidel`

Poskytuje funkce pro řešení parametrického intervalového systému Gaussovou-Seidelovou metodou. Hlavní z nich je `gauss_seidel`, která provádí N iterací Gaussove-Seidelove metody nad instancí třídy.

Funkce `gauss_seidel(S, C, x0, N)`

- `S` je instance `parametric_system`, systému na který spustíme Gaussovu-Seidelovu metodu
- `C` je reálná matice předpodmínění (zleva)
- `x0` je počáteční řešení Gaussovy-Seidelovy metody
- `N` je počet iterací které provede metoda. Metoda se zastaví při provádění menšího počtu iterací, pokud pro dvě iterace metoda dostane stejné výsledky

Vrátí dva výstupní parametry:

- `x` řešení
- `num` skutečný počet provedených iterací

4.3 Modul `optimized_preconditioner`

Poskytuje funkce pro hledání optimální matice předpodmíňování pro Gaussovu-Seidelovu metodu.

Funkce `gauss_seidel_opt(S, x0, N, parameter)`

Konstruuje předpomíňovácí matice pro optimalizaci nějakého parametru (`'width'`, `'lowerbound'`, `'upperbound'`) a provádí N iterací Gaussovy-Seidelovy metody nad předpodmíněným parametrickým intervalovým systémem.

Parametry:

- `S` je instance `parametric_system` reprezentující parametrický intervalový systém
- `x0` je počáteční box pro intervalovou Gaussovu-Seidelovu metodu
- `N` je počet iteraci
- `parameter` je řetězec z `'width'`, `'lowerbound'`, nebo `'upperbound'` ukazující který parametr bude optimalizován

V každém kroku Gaussovy-Seidelovy metody na základě hodnoty `parameter` se zavolá jedna z následujících funkcí pro vytváření lineárního programu.

4.3.1 Optimalizace (minimalizace) šířky intervalu

Poskytuje funkce pro hledání takové předpodmíňovací matice, že Gaussova-Seidelova metoda vrátí nejkratší interval pro každou složku výsledného vektoru.

Funkce `build_linprog_width(S, x, i)`

Konstruuje lineární program minimalizující výsledný interval aktuální složky v aktuální iteraci Gaussovy-Seidelovy metody. Podrobnosti konstrukce lineárního programu jsou v sekci 2.5.

Parametry:

- `S` je instance `parametric_system` reprezentující parametrický intervalový systém
- `x` je box počátečního řešení Gaussovou-Seidelovou metodou
- `i` je pozice vektoru `x`.

Výstupní parametry:

- `f` je vektor účelové funkce výsledného lineárního programu
- `A` je matice koeficientu levých stran nerovnosti lineárního programu
- `b` je vektor pravých stran nerovnosti lineárního programu
- `Aeq` je matice levých stran rovnosti lineárního programu
- `beq` je vektor pravých stran rovnosti lineárního programu

4.3.2 Optimalizace (minimalizace) supremu

Poskytuje funkce pro hledání takové matice předpodmíňování, že jediná iterace Gaussovy-Seidelovy metody má nejmenší možný supremum.

Funkce `build_linprog_upperbound`

Konstruuje lineární program minimalizující supremum výsledného intervalu v jediné iteraci Gaussovy-Seidelovy metody.

Parametry a výstupní parametry jsou stejné jako u `build_linprog_width(S, x, i)`.

4.3.3 Optimalizace (maximalizace) infimu

Poskytuje funkce pro hledání takové matice předpodmíňování, že jediná iterace Gaussovy-Seidelovy metody má největší možný infimum.

Funkce `build_linprog_lowerbound`

Konstruuje lineární program maximalizující infimum výsledného intervalu v jediné iteraci Gaussovy-Seidelovy metody.

Parametry a výstupní parametry jsou stejné jako u `build_linprog_width(S, x, i)`.

4.4 Modul `regularity`

Modul poskytuje funkce pro testování regularity symetrických intervalových matic.

Funkce `sym_double_precond(A, L, R)`

Funkce násobí zleva a zprava maticemi předpodmíňováni `L` a `R` matice `A` s využitím symetrie podle (2.1). Výstupem je předpodmíněná intervalová matice.

Funkce `sym_regular_precond(A, L, R)`

Testuje regularitu symetrické intervalové matice předpodmíněné z dvou stran pomocí 4.4 maticemi `L` a `R`. Testuje regularitu předpodmíněné intervalové matice `A` kontrolou (1.2).

Funkce `sym_regular_chol(A)`

Testuje regularitu symetrické matice metodou z 2.3 předpodmíněním z dvou stran maticemi z Choleského rozkladu. Použije na to `sym_regular_precond`. (Pouze pro intervalové matice, jejichž středová matice je pozitivně definitní).

Funkce `sym_regular_ldl(A)`

Testuje regularitu symetrické intervalové matice `A` předpodmíněním z dvou stran maticemi $(LD)^{-1}$ a $(L^T)^{-1}$, kde L a D jsou z LDL-rozkladu pomocí `sym_regular_precond`. Podrobnosti jsou v 2.2.

Funkce sym_regular_ldl2(A)

Testuje regularitu symetrické intervalové matice A předpodmíněním z dvou stran maticemi $(L\sqrt{|D|})^{-1}$ a $(\sqrt{|D|}L^T)^{-1}$ z LDL-rozkladu pomocí sym_regular_precond. Podrobnosti jsou v 2.2.

Funkce sym_regular_gs(A, C)

Testuje regularitu provedením iterace Gaussovy-Seidelovy metody podle 2.4. Parametry:

- A je vstupní symetrická intervalová matice
- C je předpodmíňovací matice. Pokud není uvedeno jinak, používá se standardní předpodmínění $C = (A^c)^{-1}$.

Funkce sym_regular_gs_opt(A, optimizing_parameter)

Funguje stejně jako sym_regular_gs, ale podle 2.5. Parametry:

- A je vstupní symetrická intervalová matice
- optimizing_parameter je řetězec pro optimalizaci parametru:
  - 'lowerbound' optimalizuje infimum
  - 'upperbound' optimalizuje supremum
  - 'width' optimalizuje délku výsledného intervalu

Funkce sym_regular_default_precond(A)

Konstruuje instance třídy parametric_system pro nějaký systém, ve kterém A je matice koeficientů levých stran, pak předpodmíní standardní předpodmíňovací matici a testuje dostačující podmínku (1.2) na relaxované levé straně parametrického systému.

4.5 Modul eigenvalue_estimating

Funkce basic_bounds(A)

Odhaduje intervaly vlastních čísel symetrické intervalové matici A pomocí Rohnovy věty 3.1. Výstupem je vektor odhadu intervalu vlastních čísel.

Funkci interlace_direct_procedure(A, next_index_proc) a interlace_indirect_procedure(A, next_index_proc)

Tyto dvě funkce jsou implementací algoritmů 1 resp. 2 ze sekci 3.2. Pro matice $\mathbf{A}^S \in \mathbb{IR}^{n\times n}$ vypočítají $\{\lambda_i^u(\mathbf{A}^s)|i \in \{1,...,n\}\}$, kde $\lambda_i^u(\mathbf{A}^s) \geq \overline{\lambda}_i(\mathbf{A}^s)$.

Výběr indexu pro minimalizaci $\overline{\lambda}_1(\mathbf{B}_i^s)$ kontroluje druhý parametr funkce, který ukazuje na jednu ze dvou funkcí:

- next_index_eig je implementace $i := \arg\min_{j=1,...,n-k+1} \lambda_1^u(B_j^S)$

- `next_index_square` je implementace $i := \arg\min_{j=1,\ldots,n-k+1} \sum_{r,s\neq j} |B_{r,s}|$

Ve výchozím stavu se používá `next_index_eig`.

Funkce `interlace_direct(mtx, next_ind_proc)`

Funkce aplikuje `interlace_direct_procedure` na `mtx` a na `-mtx` pro odhad všech intervalových vlastních čísel. Výstupem je intervalový vektor s odhadem.

Funkce `interlace_indirect(mtx, next_ind_proc)`

Funkce aplikuje `interlace_indirect_procedure` na `mtx` a na `-mtx` pro odhad všech intervalových vlastních čísel. Výstup je intervalový vektor s odhadem.

Funkce `filter_interval_sym(A, a, T, eps)`

Funkce je implementací algoritmu 3 z 3.3.
Parametry:

- `A` je symetrická intervalová matice
- `a` je počáteční interval, který algoritmus bude filtrovat
- `T` je počet iterací
- `eps` je přesnost výpočtu

Výstupem je filtrovaný interval, nebo celé číslo 0, pokud se interval zfiltroval na prázdnou množinu.

Funkce `filter_sym(IntMtx, eps, T)`

Aplikací funkce `filter_interval_sym` na `IntMtx` a `-IntMtx` odhadne vlastní čísla intervalové matice `IntMtx` s presnosti `eps` pomocí `T` iterací. Jako počáteční i-tý interval se používá odhad i-tého vlastního čísla pomocí `basic_bounds`.

Vstupní parametry odpovídají vstupním parametrům z `filter_interval_sym`.

Výstup je vektor filtrovaných vlastních čísel.

Funkce `contain_eigenvalue(A, v, method)`

Parametry:

- `A` je symetrická intervalová matice
- `v` je nějaký intervalový odhad vlastního čísla matice `A`
- `method` je pointer na nějakou metodu z modulu `regularity`. Musí mít právě jeden vstupní parametr.

Pomocí nějaké metody `method` z modulu `regularity` rozhoduje o regularitě matice $\mathbf{A} - \mathbf{v}I_n$. Pokud matice je regulární, interval v neobsahuje žádné vlastní číslo z $\Lambda(\mathbf{A})$ podle 3.4.

Funkce `filter_sym_regularity(mtx, interval_num, reg_method)`

Parametry:

- `mtx` je symetrická intervalová matice
- `interval_num` je počet podintervalů, které bude testovat `contain_eigenvalue`
- `reg_method` je nějaká metoda z modulu `regularity`

Aproximuje intervalové vlastní číslo matice $\mathbf{A}^S$. Rozčlení odhad z `basic_bounds` na `interval_num` podintervalů. Pro každý podinterval λ otestuje $\mathbf{A} - \lambda I$ na regularitu pomocí metody `reg_method`.

4.6 Příklad

Uveďme snadný příklad. V tomto příkladu definujme matici a parametrický systém, pak rozhodneme o regularitě definované matice a najdeme odhady její vlastních čísel.

```
>> A = midrad([3,1;1,2], 0.1);
```

Definovali jsme symetrickou intervalovou matici $\mathbf{A} = \begin{pmatrix} [2.9, 3.1] & [0.9, 1.1] \\ [0.9, 1.1] & [1.9, 2.1] \end{pmatrix}$. Zkonstruujme parametrický systém pro lineární intervalový systém s matici koeficientů levých stran $\mathbf{A}$ a nulovým vektorem na pravé stráně.

```
>> PS = parametric_from_symmetric(A, intval(zeros(2,1)))
PS =
  parametric_system with properties:

     P: [3×1 intval]
     n: 2
     K: 3
    A0: [2×2 double]
    b0: [2×1 double]
    As: {[2×2 double]  [2×2 double]  [2×2 double]}
    bs: [2×3 double]
```

Definovali jsme instance parametrického systému $\mathbf{Ax} = \mathbf{0}$. Pak můžeme na něho použít Gaussovu-Seidelovu metodu s počátečním boxem $\mathbf{x}^0 = [-1,1]^3$ se standardním předpodmíněním $C = (A^c)^{-1}$. Matice C lze najít pomocí přikážu `inv(A.mid)`.

```
>> GSres = gauss_seidel(PS, inv(A.mid), repmat(infsup(-1,1), 3, 1), 1)
intval GSres =
[   -0.0639,    0.0639]
[   -0.0870,    0.0870]
```

Vidíme, že výsledný box je v topologickém vnitřku počátečního boxu $\mathbf{x}^0$, což znamená že matice A je regulární.

Odhadneme její vlastní čísla pomocí Rohnovy věty (3.1) a podíváme se jak moc tyto odhady uřízne metoda `filter_sym_regularity` s rozčleněním intervalu z Rohnovy věty na 100 podintervalů. Použijme `filter_sym_regularity` s `sym_regular_default_precond` pro testování regularity. Pak zkusme rozčlenit vstupní interval na 500 podintervalů. Podíváme se také na odhad pomocí `filter_sym` s přesností $\varepsilon = 0.1$ a počtem iteraci $T = 30$. Aplikujme také přímou interlace metodu se standardním výběrem indexu.

```
>> basic_bounds(A)
intval ans =
[    3.4180,    3.8181]
[    1.1819,    1.5820]
>> filter_sym_regularity(A, 100, @sym_regular_default_precond)
intval ans =
[    3.4260,    3.8101]
[    1.1899,    1.5740]
>> filter_sym_regularity(A, 500, @sym_regular_default_precond)
intval ans =
[    3.4292,    3.8085]
[    1.1915,    1.5708]
>> filter_sym(A, 0.1, 30)
intval ans =
[    3.4264,    3.8106]
[    1.1894,    1.5736]
>> interlace_direct(A, @next_index_eig)
intval ans =
[    3.4180,    3.8084]
[    1.1819,    1.5820]
```

Vidíme, že odhad pomocí Rohnovy věty byl docela přesný, ale často to tak není. Rozdíl přesností odhadů mezi rozčleněním na 100 a na 500 podintervalů je také zřejmý.

Relaxujme parametrický systém. Jak se očekává, dostali jsme relaxaci systému a ztratili jsme závislosti mezi prvky.

```
[Ar, br] = PS.relax_parametric_system
intval Ar =
[    2.8999,    3.1001] [    0.8999,    1.1001]
[    0.8999,    1.1001] [    1.8999,    2.1001]
intval br =
    0.0000
    0.0000
```

Podíváme se na intervalový vektor parametrů. Najdeme nějakou instanci pro nějaký reálný vektor z intervalového vektoru parametrů.

```
>> PS.P
intval ans =
[    2.8999,    3.1001]
[    1.8999,    2.1001]
[    0.8999,    1.1001]
>> [Ai, bi] = PS.parametric_system_instance([3; 2; 1])
Ai =
     3     1
     1     2
bi =
     0
     0
```

Podívali jsme se na příklad použití vytvořeného námi programu. Přestože program může vypadat komplikovaně, všechny funkce, které jsou určeny k použití uživatelem jsou dokumentované.

5. Numerická analýza

V této kapitole uveďme výsledky testování regularity pro různé typy matic a porovnání kvality odhadů vlastních čísel. Na začátku uveďme seznam funkcí, které se budou testovat; uveďme popis každého typu matic a výsledky testování.

Pro některé typy generovaných matic uveďme navíc střední poloměr generované matice ($\mu(A^{\Delta}_{ij})$) a standardní odchylku ($\sigma(A^{\Delta}_{ij})$) pro každý set, protože se mohou lišit od očekávaných.

5.1 Numerická analýza algoritmů testování regularity

5.1.1 Testující se algoritmy

Značení	Funkce	Popis
DP	`sym_regular_default_precond`	Sekce 2.1
CHOL	`sym_regular_chol`	Sekce 2.3
LDL1	`sym_regular_ldl`	Sekce 2.2
LDL2	`sym_regular_ldl2`	Sekce 2.2
GS	`sym_regular_gs`	Sekce 2.4
GSSUP	`sym_regular_gs_opt(m, 'upperbound')`	Sekce 2.5, 2.5.2
GSINF	`sym_regular_gs_opt(m, 'lowerbound')`	Sekce 2.5, 2.5.3
GSWID	`sym_regular_gs_opt(m, 'width')`	Sekce 2.5, 2.5.1

5.1.2 Metodika

Efektivitu testování regularity jsme odhadli pro tři typy symetrických náhodných intervalových matic $\mathbf{A} \in \mathbb{IR}^{n\times n}$:

- generované matice splňující postačující podmínku pro regularitu ($\rho(|(A^c)^{-1}|A^{\Delta}) < 1$);
- takové $\mathbf{A}$, že $((A^c)^{-1}\mathbf{A})^{\Delta} \le \varepsilon$ pro postačující malé ε;
- náhodné symetrické matice $\mathbf{A}$ s regulární A^c.

Pokud se algoritmus rozhodne, že matice je regulární podle nějakého algoritmu, označme tu matici jedničkou, v opačném případě nulou. Pak pro každý algoritmus spočítejme průměr a standardní odchylku této hodnoty přes všechny matice.

5.1.3 Náhodné regulární matice

První typ matic je symetrická intervalová matice $\mathbf{B} = [A^c - \frac{A^{\Delta}}{c+\varepsilon}, A^c + \frac{A^{\Delta}}{c+\varepsilon}]$, kde $A^c = L^T L$, $c = \rho(|(A^c)^{-1}|A^{\Delta})$, $\varepsilon = \frac{c(1-\rho')}{\rho'}$ a $\rho' \in [0,1]$ je náhodné. Dolní trojúhelníkovou regulární $L \in \mathbb{R}^{n\times n}$ a symetrickou $A^{\Delta} \in \mathbb{R}^{n\times n}$ náhodně zvolíme.

Pro tento typ matic vždy položme $L_{ij} \in [-20, 20]$. Pro generovanou matici $\mathbf{B}$ platí $\rho(|(B^c)^{-1}|B^\Delta) = \rho'$. Přestože $\rho' \in$ [0,1], nikdy se nestalo, aby $\rho' = 1$, protože ρ' je rovnoměrně rozložená náhodná veličina a pravděpodobnost takového jevu je malá.

Metoda	Průměrná hodnota	Standardní odchylka
DP	1.0000	0.0000
CHOL	0.9940	0.0773
LDL1	0.9920	0.0892
LDL2	0.9920	0.0892
GS	0.4520	0.4982
GSSUP	0.8840	0.3205
GSINF	0.8880	0.3157
GSWID	0.8920	0.3107
	$\mu(A^\Delta_{ij}) = 0.1903$	$\sigma(A^\Delta_{ij}) = 0.2557$

Tabulka 5.1: Testování regularity na náhodných regulárních maticích, $n = 5$, $\rho' \in [0, 1]$, $A^\Delta \in [0, 1]$, 500 matic

Metoda	Průměrná hodnota	Standardní odchylka
DP	1.0000	0.0000
CHOL	0.9840	0.1256
LDL1	0.9840	0.1256
LDL2	0.9840	0.1256
GS	0.4760	0.4999
GSSUP	0.8960	0.3056
GSINF	0.8880	0.3157
GSWID	0.8980	0.3030
	$\mu(A^\Delta_{ij}) = 0.2312$	$\sigma(A^\Delta_{ij}) = 0.3549$

Tabulka 5.2: Testování regularity na náhodných regulárních maticích, $n = 5$, $\rho' \in [0, 1]$, $A^\Delta \in [0, 1.5]$, 500 matic

Metoda	Průměrná hodnota	Standardní odchylka
DP	1.0000	0.0000
CHOL	0.9940	0.0773
LDL1	0.9940	0.0773
LDL2	0.9940	0.0773
GS	0.4860	0.5003
GSSUP	0.8700	0.3366
GSINF	0.8680	0.3388
GSWID	0.8800	0.3253
	$\mu(A^\Delta_{ij}) = 0.1161$	$\sigma(A^\Delta_{ij}) = 0.1346$

Tabulka 5.3: Testování regularity na náhodných regulárních maticích, $n = 5$, $\rho' \in [0, 1]$, $A^\Delta \in [0, 0.5]$, 500 matic

Metoda	Průměrná hodnota	Standardní odchylka
DP	1.0000	0.0000
CHOL	0.9800	0.1407
LDL1	0.9800	0.1407
LDL2	0.9800	0.1407
GS	0.1900	0.3943
GSSUP	0.5900	0.4943
GSINF	0.5600	0.4989
GSWID	0.5900	0.4943
	$\mu(A^{\Delta}_{ij}) = 0.0025$	$\sigma(A^{\Delta}_{ij}) = 0.0110$

Tabulka 5.4: Testování regularity na náhodných regulárních maticích, $n = 10$ $\rho' \in [0, 1]$, $A^{\Delta} \in [0, 1]$

Metoda	Průměrná hodnota	Standardní odchylka
DP	1.0000	0.0000
CHOL	1.0000	0.0000
LDL1	1.0000	0.0000
LDL2	1.0000	0.0000
GS	0.3400	0.4761
GSSUP	0.6600	0.4761
GSINF	0.6300	0.4852
GSWID	0.6700	0.4726
	$\mu(A^{\Delta}_{ij}) = 0.0006$	$\sigma(A^{\Delta}_{ij}) = 0.0007$

Tabulka 5.5: Testování regularity na náhodných regulárních maticích, $n = 10$ $A^{\Delta} \in [0.001, 0.002]$

Z výsledků uvedených výš je vidět, že testování regularity testováním postačující podmínky s předpodmíněním maticemi z Choleského a LDL-rozkladů dává skoro stejné výsledky a jsou přiblížené k tím, které dává testování regularity se standardním předpodmíněním. Testování pomocí Gaussovy-Seidelovy metody je méně efektivní.

5.1.4 Náhodné symetrické H-matice

Táto třída obsahuje hodně intervalových matic $\mathbf{A} \in \mathbb{IR}^{n\times n}$, takových, že pro systém $\mathbf{A}\mathbf{x} = 0$ jedna iterace Gaussovy-Seidelovy metody pro $\mathbf{x}^0 = [-1,1]^n$ vrátí box v topologickém vnitřku $\mathbf{x}^0$.

Symetrická intervalová matice $A = [A^c - A^{\Delta}, A^c + A^{\Delta}]$ je generovaná následujícím způsobem: na začátku se generuje regulární symetrická $A^c \in [-20, 20]^{n\times n}$, pak se generuje $A^{\Delta} \in [A^{\Delta}_{\min}, A^{\Delta}_{\max}]^{n\times n}$ taková, že $((A^c)^{-1}\mathbf{A})^{\Delta} \le \varepsilon$ pro nějaký dostatečně malý $\varepsilon \in \mathbb{R}$ a dané $A^{\Delta}_{\min}, A^{\Delta}_{\max} \in \mathbb{R}$, $A^{\Delta}_{\min} \le A^{\Delta}_{\max}$. Vždy předpokládáme, že $\varepsilon \le A^{\Delta}_{\min}$, ale pro velké matice to nezaručuje, že dostaneme H-matici.

Na tomto typu matic netestujeme CHOL-metodu, protože většina matic nemají středovou matici pozitivně definitní, takže táto metoda nerozpozná regularitu těchto matic ve většině případů.

Metoda	Průměrná hodnota	Standardní odchylka
DP	1.0000	0.0000
LDL1	0.9599	0.1969
LDL2	0.5000	0.5025
GS	1.0000	0.0000
GSSUP	1.0000	0.0000
GSINF	1.0000	0.0000
GSWID	1.0000	0.0000
	$\mu(A^{\Delta}_{ij}) = 0.7061$	$\sigma(A^{\Delta}_{ij}) = 0.1390$

Tabulka 5.6: Testování regularity na H-maticích, $n = 5$, $A^{\Delta}_{ij} \in [0.5, 1]$, $((A^c)^{-1}\mathbf{A})^{\Delta} \leq 0.1$, 100 matic

Pro tento set je vidět, že $\varepsilon = 0.1$ je dostatečně malé, abychom dostali H-matice velikosti 5×5; 95.99 % z nich jsou rozpoznatelné pomoci LDL1, ale jen polovina jsou rozpoznatelné pomoci LDL2.

Metoda	Průměrná hodnota	Standardní odchylka
DP	0.8003	0.4001
LDL1	0.3829	0.4865
LDL2	0.1697	0.3756
GS	0.5450	0.4983
GSSUP	0.6471	0.4782
GSINF	0.6471	0.4782
GSWID	0.6471	0.4782
	$\mu(A^{\Delta}_{ij}) = 0.7472$	$\sigma(A^{\Delta}_{ij}) = 0.1437$

Tabulka 5.7: Testování regularity na H-maticích, $n = 5$, $A^{\Delta}_{ij} \in [0.5, 1]$, $((A^c)^{-1}\mathbf{A})^{\Delta} \leq 0.5$, 100 matic

Metoda	Průměrná hodnota	Standardní odchylka
DP	1.0000	0.0000
LDL1	0.9940	0.0773
LDL2	0.8664	0.3405
GS	1.0000	0.0000
GSSUP	1.0000	0.0000
GSINF	1.0000	0.0000
GSWID	1.0000	0.0000
	$\mu(A^{\Delta}_{ij}) = 0.3679$	$\sigma(A^{\Delta}_{ij}) = 0.0725$

Tabulka 5.8: Testování regularity na H-maticích, $n = 5$, $A^{\Delta}_{ij} \in [0.25, 0.5]$, $((A^c)^{-1}\mathbf{A})^{\Delta} \leq 0.1$, 660 matic

Metoda	Průměrná hodnota	Standardní odchylka
DP	1.0000	0.0000
LDL1	0.0050	0.0707
LDL2	0.0000	0.0000
GS	1.0000	0.0000
GSSUP	1.0000	0.0000
GSINF	1.0000	0.0000
GSWID	1.0000	0.0000
	$\mu(A^c) = 0.3691$	$\sigma(A^c) = 0.0716$

Tabulka 5.9: Testování regularity na H-maticích, $n = 10$, $A^\Delta_{ij} \in [0.25, 0.5]$, $((A^c)^{-1}\mathbf{A})^\Delta \leq 0.1$

Metoda	Průměrná hodnota	Standardní odchylka
DP	1.0000	0.0000
LDL1	0.9698	0.1712
LDL2	0.6833	0.4656
GS	0.9985	0.0388
GSSUP	0.9985	0.0388
GSINF	0.9985	0.0388
GSWID	0.9985	0.0388
	$\mu(A^\Delta_{ij}) = 0.1225$	$\sigma(A^\Delta_{ij}) = 0.0432$

Tabulka 5.10: Testování regularity na H-maticích, $n = 10$, $A^\Delta_{ij} \in [0.05, 0.2]$, $((A^c)^{-1}\mathbf{A})^\Delta \leq 0.05$, 660 matic

Pro tento set je vidět, že střední délka intervalu 0.1225 stačí, aby matice byla silně regulární. Všechny matice v tomto setu jsou regulární a LDL1 skoro všechny z nich rozpoznává. LDL2 pracuje s těmi maticemi hůř.

5.1.5 Náhodné symetrické intervalové matice

Tato třída matic je množina náhodných intervalových matic $\mathbf{A} \in \mathbb{IR}^{n\times n}$ takových, že platí $\mathbf{A} = \mathbf{A}^T$. Každá taková $\mathbf{A}$ je generována následujícím způsobem. Na začátku je generována náhodná symetrická regulární $A^c \in [-20, 20]^{n\times n}$ a náhodná $A^\Delta \in [A^\Delta_{\min}, A^\Delta_{\max}]^{n\times n}$, pak $\mathbf{A} = [A^c - A^\Delta, A^c + A^\Delta]$. Konkretní hodnoty $A^\Delta_{\min}, A^\Delta_{\max} \in \mathbb{R}$ uvedeme dál.

Metoda	Průměrná hodnota	Standardní odchylka
DP	1.0000	0.0000
LDL1	0.2467	0.4318
LDL2	0.0000	0.0000
GS	0.9901	0.0997
GSSUP	0.9967	0.0578
GSINF	0.9967	0.0578
GSWID	0.9967	0.0578

Tabulka 5.11: Testování regularity na náhodných symetrických maticích, $n = 5$, $A^{\Delta}_{\min} = 0, A^{\Delta}_{\max} = 3.5$

Metoda	Průměrná hodnota	Standardní odchylka
DP	0.9801	0.1403
LDL1	0.0167	0.1283
LDL2	0.0000	0.0000
GS	0.9301	0.2556
GSSUP	0.9567	0.2040
GSINF	0.9567	0.2040
GSWID	0.9567	0.2040

Tabulka 5.12: Testování regularity na náhodných symetrických maticích, $n = 5$, $A^{\Delta}_{\min} = 0.5, A^{\Delta}_{\max} = 3.5$

Metoda	Průměrná hodnota	Standardní odchylka
DP	0.8034	0.3982
LDL1	0.0000	0.0000
LDL2	0.0000	0.0000
GS	0.6367	0.4818
GSSUP	0.6901	0.4633
GSINF	0.6901	0.4633
GSWID	0.6901	0.4633

Tabulka 5.13: Testování regularity na náhodných symetrických maticích, $n = 5$, $A^{\Delta}_{\min} = 1, A^{\Delta}_{\max} = 3.5$

Metoda	Průměrná hodnota	Standardní odchylka
DP	0.1634	0.3703
LDL1	0.0000	0.0000
LDL2	0.0000	0.0000
GS	0.0634	0.2440
GSSUP	0.1067	0.3093
GSINF	0.1067	0.3093
GSWID	0.1067	0.3093

Tabulka 5.14: Testování regularity na náhodných symetrických maticích, $n = 5$, $A^{\Delta}_{\min} = 1.5, A^{\Delta}_{\max} = 3.5$

Metoda	Průměrná hodnota	Standardní odchylka
DP	0.9234	0.2666
LDL1	0.0000	0.0000
LDL2	0.0000	0.0000
GS	0.5267	0.5002
GSSUP	0.7134	0.4530
GSINF	0.7134	0.4530
GSWID	0.7134	0.4530

Tabulka 5.15: Testování regularity na náhodných symetrických maticích, $n = 10$, $A^{\Delta}_{\min} = 0, A^{\Delta}_{\max} = 2$

5.1.6 Výkon

Ukažme výsledky měřeni času testování regularity pro typ matic uvedený v sekci 5.1.3. Pro jiné typy matic výsledky by měly byt obdobné, protože časová složitost nezáleží na typu testující se matice.

Metoda	Průměrný čas, sec	Standardní odchylka
DP	0.0313	0.0020
CHOL	0.4042	0.0141
LDL1	0.4053	0.0130
LDL2	0.4040	0.0133
GS	0.2063	0.0099
GSSUP	0.3426	0.0110
GSINF	0.3420	0.0097
GSWID	0.2992	0.0095

Tabulka 5.16: Měření času testování regularity, $n = 5$, $\rho' \in [0, 1]$, $L_{ij} \in [-20, 20]$, $A^{\Delta}_{ij} \in [0, 1]$.

Metoda	Průměrný čas, sec	Standardní odchylka
DP	0.0961	0.0070
CHOL	6.0858	0.1817
LDL1	6.0232	0.1588
LDL2	6.0307	0.1736
GS	2.3543	0.0369
GSSUP	3.0650	0.0448
GSINF	2.7462	0.7048
GSWID	2.2477	0.9894

Tabulka 5.17: Měření času testování regularity, $n = 10$, $\rho' \in [0, 1]$, $L_{ij} \in [-20, 20]$, $A^{\Delta}_{ij} \in [0.001, 0.002]$.

5.2 Numerická analýza algoritmu odhadu vlastních čísel

V této sekci se podíváme na kvalitu odhadů vlastních čísel symetrických intervalových matic. Následující tabulka pojmenovává algoritmy a odkazuje na předchozí sekce s jejich popisem v tomto textu. Metody začínající "FR" používají nějakou funkci z modulu `regularity` z praktické části jako třetí parametr. Podrobnosti jsou v sekcích 4.5 a 4.4.

Značení	Funkce	Popis
FRdef	`filter_sym_regularity` s `sym_regular_default_precond`	Sekce 3.4, 2.1
FRLDL1	`filter_sym_regularity` s `sym_regular_ldl`	Sekce 3.4, 2.2
FRLDL2	`filter_sym_regularity` s `sym_regular_ldl2`	Sekce 3.4, 2.2
FRGS	`filter_sym_regularity` s `sym_regular_gs`	Sekce 3.4, 2.4
FRGSwidth	`filter_sym_regularity` s `sym_regular_gs_opt`	Sekce 3.4, 2.5, 2.5.1
FRGSup	`filter_sym_regularity` s `sym_regular_gs_opt`	Sekce 3.4, 2.5, 2.5.2
FRGSlow	`filter_sym_regularity` s `sym_regular_gs_opt`	Sekce 3.4, 2.5, 2.5.3
F1	`filter_sym`	Sekce 3.3, alg. 3, $\varepsilon = 0.5, T = 20$
F2	`filter_sym`	Sekce 3.3, alg. 3, $\varepsilon = 0.25, T = 20$

Značení	Funkce	Popis
DIeig	`interlace_direct` s `next_index_eig`	Sekce 3.2, Alg 1
DIsq	`interlace_indirect` s `next_index_square`	Sekce 3.2, Alg 1 s použitím (3.6)
IIeig	`interlace_indirect` s `next_index_eig`	Sekce 3.2, Alg 2
DIsq	`interlace_indirect` s `next_index_square`	Sekce 3.2, Alg 2 s použitím (3.6)

Tabulka 5.18: Seznam testujících se algoritmů odhadu vlastních čísel

5.2.1 Metodika

Každá symetrická intervalová matice velikosti $n \times n$ má n vlastních čísel, která jsou reálné intervaly. Tyto intervaly můžeme odhadnout pomocí Rohnovy věty relativně rychle. Kvalitu upřesnění těchto intervalu můžeme reprezentovat jako podíl délky upřesněného intervalu a původního. Formálně, pokud $\lambda_i^j \in \mathbb{IR}$ je j-té vlastní číslo (interval) z Rohnovy věty a $\lambda_i'^j \in \mathbb{IR}$ je upřesněný interval pomocí nějaké konkrétní metody pro intervalovou matici A_i, počítáme průměrný $1 - \frac{(\lambda_i'^j)^\Delta}{(\lambda_i^j)^\Delta}$ přes všechna i.

Je zřejmé, že čím je větší počet podintervalů pro FRdef, FRLDL1, FRLDL2, FRGS, FRGSwidth, FRGSup, FRGSlow metody, tím je přesnější odhad vlastního čísla. V popisu tabulek níž je speciálně označen počet podintervalů pro tyto metody.

5.2.2 Náhodné symetrické intervalové matice typu $\mathbf{B}^T\mathbf{B}$

Generované matice jsou typu $\mathbf{A} = \mathbf{B}^T\mathbf{B}$, kde $\mathbf{B}$ je náhodná symetrická intervalová matice, $B^c \in [-20, 20]^{n\times n}$. Relativní průměr ukazuje podíl, kolik algoritmus v průměru uřízl od aproximaci pomocí Rohnovy věty.

Metoda	Relativní průměr	Standardní odchylka
FRdef	0.1975	0.1787
FRLDL1	0.1458	0.1732
FRLDL2	0.1331	0.1685
FRGS	0.0102	0.0501
FRGSwidth	0.0833	0.1509
FRGSup	0.0833	0.1509
FRGSlow	0.0833	0.1509
F1	0.1494	0.1415
F2	0.1510	0.1435
DIeig	0.0346	0.0755
DIsq	0.0321	0.0735
IIeig	0.0012	0.0122
IIsq	0.0012	0.0122

Tabulka 5.19: Přesnost odhadu vlastních čísel na $\mathbf{B}^T\mathbf{B}$, $n = 3$, $B^\Delta_{ij} \in [0, 0.25]^{n\times n}$, 500 podintervalů pro FR-metody, 100 matic

Metoda	Relativní průměr	Standardní odchylka
FRdef	0.1865	0.1667
FRLDL1	0.1110	0.1464
FRLDL2	0.1012	0.1406
FRGS	0.0061	0.0294
FRGSwidth	0.0742	0.1292
FRGSup	0.0742	0.1292
FRGSlow	0.0742	0.1292
F1	0.1341	0.1234
F2	0.1376	0.1281
DIeig	0.0409	0.0751
DIsq	0.0391	0.0742
IIeig	0.0000	0.0000
IIsq	0.0000	0.0000

Tabulka 5.20: Přesnost odhadu vlastních čísel na $\mathbf{B}^T\mathbf{B}$, $n = 3$, $B^\Delta_{ij} \in [0, 0.5]$, 500 podintervalů pro FR-metody, 100 matic

Metoda	Relativní průměr	Standardní odchylka
FRdef	0.1808	0.1355
FRLDL1	0.0620	0.0876
FRGSwidth	0.0388	0.0776
F1	0.1216	0.0967
F2	0.1260	0.1033
DIeig	0.0171	0.0402
IIeig	0.0000	0.0003

Tabulka 5.21: Přesnost odhadu vlastních čísel na $\mathbf{B}^T\mathbf{B}$, $n = 5$, $B^{\Delta}_{ij} \in [0, 0.25]$, 500 podintervalů pro FR-metody, 100 matic

Je vidět, ze nejlepší výsledek dává FRdef; F1 a F2 jsou skoro stejně efektivní. Předpodmínění maticemi z LDL rozkladu není nejefektivnější a je pomalejší než FRdef, což je vidět dal. Interlace metody nejsou moc efektivní pro náhodné matice, protože jsou založené na interlace-vlastnosti a dobře fungují na maticích s překrývajícími se vlastními intervaly podle (Hladík a kol., 2010), věta 3.3).

Metoda	Relativní průměr	Standardní odchylka
FRdef	0.0282	0.0623
FRLDL1	0.0072	0.0249
FRLDL2	0.0072	0.0249
FRGSwidth	0.0009	0.0080
F1	0.0194	0.0438
F2	0.0195	0.0445
DIeig	0.0121	0.0275
IIeig	0.0030	0.0104

Tabulka 5.22: Přesnost odhadu vlastních čísel na $\mathbf{B}^T\mathbf{B}$, $n = 10$, $B^{\Delta}_{ij} \in [0, 0.25]$, 50 podintervalů pro FR-metody 100 matic

Metoda	Relativní průměr	Standardní odchylka
FRdef	0.1791	0.1354
FRLDL1	0.0672	0.1036
FRGSwidth	0.0352	0.0793
F1	0.1247	0.0970
F2	0.1297	0.1036
DIeig	0.0150	0.0412
IIeig	0.0000	0.0000

Tabulka 5.23: Přesnost odhadu vlastních čísel na $\mathbf{B}^T\mathbf{B}$, $n = 5$, $B^{\Delta}_{ij} \in [0, 0.25]$, 100 podintervalů pro FR-metody, 100 matic

Metoda	Relativní průměr	Standardní odchylka
FRdef	0.0067	0.0239
FRLDL1	0.0038	0.0164
FRLDL2	0.0038	0.0164
FRGSwidth	0.0000	0.0000
F1	0.0055	0.0191
F2	0.0055	0.0193
DIeig	0.0182	0.0324
IIeig	0.0078	0.0180

Tabulka 5.24: Přesnost odhadu vlastních čísel na $\mathbf{B}^T\mathbf{B}$, $n = 10$, $B_{ij}^{\Delta} \in [0, 0.5]$, 50 podintervalů pro FR-metody, 100 matic

5.2.3 Náhodné symetrické intervalové matice s regulární středovou maticí

Táto třída symetrických intervalových matic je generovaná následujícím způsobem. Nechť $A^c \in \mathbb{R}^{n\times n}$ je náhodná symetrická regulární (ne nutně pozitivně definitní) reálná matice, $A^{\Delta} \in [0,r]^{n\times n}$ je náhodná symetrická reálná matice pro nějaké $r \in \mathbb{R}$. $\mathbf{A} = [A^c - A^{\Delta}, A^c + A^{\Delta}]$ je nějaká generovaná symetrická intervalová matice.

Metoda	Relativní průměr	Standardní odchylka
FRdef	0.1858	0.1573
FRLDL1	0.1724	0.1574
FRLDL2	0.1683	0.1556
FRGS	0.0092	0.0386
FRGSwid	0.0808	0.1280
FRGSsup	0.0808	0.1280
FRGSinf	0.0808	0.1280
F1	0.1480	0.1199
F2	0.1481	0.1201
DIeig	0.0084	0.0375
DIsq	0.0073	0.0360
IIeig	0.0001	0.0001
IIsq	0.0001	0.0001
	$\mu(A^{\Delta}) = 0.1233$	$\sigma(A^{\Delta}) = 0.0592$

Tabulka 5.25: Přesnost odhadu vlastních čísel, A^c je regulární, $n = 3$, $r = 0.25$, 500 podintervalů pro FR-metody, 100 matic

Metoda	Relativní průměr	Standardní odchylka
FRdef	0.1781	0.1602
FRLDL1	0.1557	0.1589
FRLDL2	0.1466	0.1551
FRGS	0.0070	0.0327
FRGSwid	0.0816	0.1329
FRGSsup	0.0816	0.1329
FRGSinf	0.0816	0.1329
F1	0.1413	0.1223
F2	0.1416	0.1227
DIeig	0.0105	0.0398
DIsq	0.0103	0.0398
IIeig	0.0001	0.0001
IIsq	0.0001	0.0001
	$\mu(A_{ij}^{\Delta}) = 0.2488$	$\sigma(A_{ij}^{\Delta}) = 0.1174$

Tabulka 5.26: Přesnost odhadu vlastních čísel, A^c je regulární, $n = 3$, $r = 0.5$, 500 podintervalů pro FR-metody, 100 matic

Metoda	Relativní průměr	Standardní odchylka
FRdef	0.1881	0.1571
FRLDL1	0.1818	0.1570
FRLDL2	0.1795	0.1564
FRGS	0.0042	0.0236
FRGSwid	0.0800	0.1224
FRGSsup	0.0800	0.1224
FRGSinf	0.0800	0.1224
F1	0.1499	0.1193
F2	0.1500	0.1194
DIeig	0.0072	0.0331
DIsq	0.0054	0.0251
IIeig	0.0001	0.0001
IIsq	0.0001	0.0001
	$\mu(A_{ij}^{\Delta}) = 0.0513$	$\sigma(A_{ij}^{\Delta}) = 0.0242$

Tabulka 5.27: Přesnost odhadu vlastních čísel, A^c je regulární, $n = 3$, $r = 0.1$, 500 podintervalů pro FR-metody, 100 matic

Metoda	Relativní průměr	Standardní odchylka
FRdef	0.2226	0.1425
FRLDL1	0.1695	0.1408
FRLDL2	0.1542	0.1389
FRGS	0.0000	0.0000
FRGSwid	0.0656	0.1177
FRGSsup	0.0656	0.1177
FRGSinf	0.0656	0.1177
F1	0.1757	0.1089
F2	0.1763	0.1095
DIeig	0.0016	0.0104
DIsq	0.0009	0.0076
IIeig	0.0001	0.0001
IIsq	0.0001	0.0001
	$\mu(A^{\Delta}_{ij}) = 0.1225$	$\sigma(A^{\Delta}_{ij}) = 0.0554$

Tabulka 5.28: Přesnost odhadu vlastních čísel, A^c je regulární, $n = 5$, $r = 0.25$, 500 podintervalů pro FR-metody, 100 matic

Metoda	Relativní průměr	Standardní odchylka
FRdef	0.2135	0.1444
FRLDL1	0.1293	0.1330
FRLDL2	0.1064	0.1229
FRGS	0.0002	0.0023
FRGSwid	0.0593	0.1072
FRGSsup	0.0593	0.1072
FRGSinf	0.0593	0.1072
F1	0.1672	0.1091
F2	0.1692	0.1113
DIeig	0.0035	0.0188
DIsq	0.0025	0.0168
IIeig	0.0002	0.0017
IIsq	0.0001	0.0004
	$\mu(A^{\Delta}) = 0.2489$	$\sigma(A^{\Delta}_{ij}) = 0.1103$

Tabulka 5.29: Přesnost odhadu vlastních čísel, A^c je regulární, $n = 5$, $r = 0.5$, 500 podintervalů pro FR-metody, 100 matic

Metoda	Relativní průměr	Standardní odchylka
FRdef	0.2593	0.1213
FRLDL1	0.1413	0.1166
FRLDL2	0.1053	0.1094
FRGS	0.0000	0.0000
FRGSwid	0.0444	0.0845
FRGSsup	0.0444	0.0845
FRGSinf	0.0444	0.0845
F1	0.2123	0.0938
F2	0.2142	0.0951
DIeig	0.0000	0.0008
DIsq	0.0000	0.0008
IIeig	0.0000	0.0000
IIsq	0.0000	0.0000
	$\mu(A^{\Delta}_{ij}) = 0.0498$	$\sigma(A^{\Delta}_{ij}) = 0.0214$

Tabulka 5.30: Přesnost odhadu vlastních čísel, A^c je regulární, $n = 10$, $r = 0.1$, 50 podintervalů pro FR-metody, 50 matic

5.2.4 Výkon

Jako u měření času při testování regularity, uveďme v této sekci výsledky měření času pro jeden typ matic popsaný v 5.2.2.

Metoda	Průměrný čas, sec	Standardní odchylka
FRdef	0.6334	0.2154
FRLDL1	1.9307	1.0989
FRLDL2	2.0392	1.2343
FRGS	0.4311	0.2523
FRGSwidth	2.4144	1.3985
FRGSup	2.8850	1.6037
FRGSlow	2.8657	1.8273
F1	0.0205	0.0030
F2	0.0219	0.0051
DIeig	0.0317	0.0034
DIsq	0.0195	0.0018
IIeig	0.0348	0.0031
IIsq	0.0209	0.0018

Tabulka 5.31: Časové měření odhadu vlastních čísel, $n = 3$, $r \in [0, 0.25]$, 500 podintervalů pro FR-metody, 100 matic

Metoda	Průměrný čas, sec	Standardní odchylka
FRdef	3.4890	0.8647
FRLDL1	140.7042	18.7553
FRLDL2	142.5501	17.4365
FRGSwidth	56.6158	3.5760
F1	0.0549	0.0042
F2	0.0544	0.0039
DIeig	0.2533	0.0113
IIeig	0.2499	0.0093

Tabulka 5.32: Časové měření odhadu vlastních čísel, $n = 10$, $r \in [0, 0.25]$, 50 podintervalů pro FR-metody, 100 matic

Nejrychlejší metody jsou interlace a filtrovací metody F1 a F2, přičemž při zvětšování velikosti matice tyto metody zachovají svou rychlost. Filtrovací metody s testováním regularity jsou velmi pomalé, protože opakují testování regularity na velkém počtu matic.

5.3 Diskuze

Podle uvedených experimentů je vidět, že standardní předpodmínění dává nejlepší, stabilní a předpověditelné výsledky pro testování regularity.

CHOL, LDL1 a LDL2 dávají skoro stejné výsledky jako DP. Rozklad Choleského a LDL-rozklad jsou pomalejší, než hledání inverzní matice, což vede k zpomalení testování regularity s použitím těchto předpodmínění.

Testování regularity pomocí Gaussove-Seidelove metody není tak efektivní, jako pomoci DP, ale využití hledání optimálního předpodmínění pro Gaussovu-Seidelovu metodu zlepšuje výsledky. Problém je ale v tom, ze GSWID, GSSUP a GSINF metody řeší n lineárních programu pro matice velikosti $n \times n$.

Standardní předpodmínění může být dostatečně efektivně použité na odhady vlastních čísel symetrických intervalových matic když přesnost je důležitější než rychlost. Hlavní výhoda filtrování pomocí testování regularity je v tom, že můžeme kontrolovat přesnost odhadu vlastního čísla.

Filtrovací metody F1 a F2 jsou mnohem rychlejší než FRdef, což je výhoda. Další výhodou je možnost kontrolovat přesnost filtrovací metody.

Interlace metody jsou užitečné jen pro ty matice, které mají překrývající se vlastní čísla, proto mají nejhorší přesnost v našich experimentech.

Kompromisem pro odhady vlastních čísel pro symetrické intervalové matice by mohla být filtrovací metoda nebo filtrování pomocí testování regularity se standardním předpodmíněním, nebo kombinace těchto dvou metod.

Závěr

V této prací jsme se podívali na koncept regularity intervalových matic a na základní vlastnosti symetrických intervalových matic a lineárních intervalových systému.

Vyzkoušeli jsme dva různé způsoby testování regularity: pomocí testování dostatečné podmínky, která zaručuje silnou regularitu, a pomocí Gaussovy-Seidelovy metody. Pro každý z těchto dvou způsobů jsme zformulovali konkretní způsoby předpodmíněni. Pro testování dostatečné podmínky popsali jsme dvoustranné předpodmínění maticemi z Choleského a LDL-rozkladu středové matice, ukázali jsme jak předpodmínit ze dvou stran s zachováním lineárních závislosti u symetrické intervalové matice a definovali jsme standardní předpodmínění. Dále jsme uvedli, že otestovat regularitu lze provedením n iterací Gaussovy-Seidelovy metody pro intervalový systém s matici koeficientu levých stran velikosti $n \times n$ a nulovým vektorem na pravé straně. Ukázali jsme, jak tento systém lze předpodmínit tak, aby výsledek Gaussovy-Seidelovy metody byl nejlepší vzhledem k délce intervalu, jeho supremu, nebo infimu, (což vede k rozpoznávaní regularity u většího počtu matic než se standardním předpodmíněním.)

Popsali jsme koncept vlastních čísel pro intervalové matice a zmínili jsme se, že vlastní čísla symetrických intervalových matic jsou podmnožiny $\mathbb{R}$ a navíc že i-té vlastní číslo symetrické intervalové matici tvoří uzavřený interval. Tyto uzavřené intervaly pro různá vlastní čísla se mohou překrývat, což nás vedlo k využití interlace vlastnosti pro odhad supremu a infimu vlastních čísel symetrických intervalových matic. Podívali jsme se na způsob filtrování libovolné množiny vlastních čísel intervalové matice a uvedli jsme algoritmus pro tuto metodu. Poté jsme popsali jak lze použít testování regularity pro odhady vlastních čísel symetrických intervalových matic.

Dále jsme uvedli dokumentaci pro naši implementaci těchto algoritmu v systému MATLAB s využitím knihovny IntLab. Popsali jsme jak fungují a jak jsou implementovány základní funkce založené na algoritmech uvedených v 2. a 3. kapitolách.

V poslední kapitole jsme se věnovali numerickému porovnání implementace algoritmu pro testování regularity a odhad vlastních čísel. Generovali jsme na to několik druhů matic. Z uvedených výsledků vyplývá, že pro testování regularity nejefektivnější metodou je testování dostatečné podmínky se standardním předpodmíněním. Vysoká efektivita této metody je dosažená díky její asymptotické složitosti a počtu rozpoznaných matic; potvrzují to časová a numerická měřeni.

Jiná situace je u odhadů vlastních čísel. Interlace metody jsou efektivní pouze pro ty intervalové matice, jejichž vlastní čísla se překrývají. Pro obecnou symetrickou intervalovou matici filtrovací metoda, která je uvedena v sekcí 3.3, dává nejlepší výsledky. Není tak pomalá a pomocí ní lze odhadnout vlastní čísla s velkou přesnosti. Filtrovací algoritmy založené na testování regularity jsou pomalé kvůli mnohokrátnému testování regularity, ale jejich použitím lze také odhadnout vlastní číslo s velkou přesností.

Seznam použité literatury

Hansen, E. R. (1965). Interval arithmetic in matrix computations, Part I. *J. Soc. Ind. Appl. Math., Ser. B, Numer. Anal.*, **2**(2), 308–320.

Hansen, E. R. a Smith, R. (1967). Interval arithmetic in matrix computations. II. *SIAM J. Numer Anal.*, **4**(1), 1–9.

Hladík, M. (2016). Optimal preconditioning for the interval parametric Gauss–Seidel method. In Nehmeier et al., M., editor, *Scientific Computing, Computer Arithmetic, and Validated Numerics: 16th International Symposium, SCAN 2014, Würzburg, Germany, September 21-26*, volume 9553 of *LNCS*, pages 116–125. Springer. ISBN 978-3-319-31769-4. doi: 10.1007/978-3-319-31769-4_10.

Hladík, M., Daney, D. a Tsigaridas, E. (2010). Bounds on real eigenvalues and singular values of interval matrices. *SIAM J. Matrix Anal. Appl.*, **31**(4), 2116–2129. doi: 10.1137/090753991.

Hladík, M., Daney, D. a Tsigaridas, E. P. (2011). A filtering method for the interval eigenvalue problem. *Appl. Math. Comput.*, **217**(12), 5236–5242.

Kearfott, R. B. (1996). *Rigorous Global Search: Continuous Problems*. Kluwer, Dordrecht.

Kearfott, R. B., Hu, C. a Novoa III, M. (1991). A review of preconditioners for the interval Gauss–Seidel method. *Interval Comput.*, **1991**(1), 59–85.

Neumaier, A. (1991). *Interval Methods for Systems of Equations*. Encyclopedia of Mathematics and its Applications. Cambridge University Press. doi: 10.1017/CBO9780511526473.

Rump, S. (1999). INTLAB - INTerval LABoratory. In Csendes, T., editor, *Developments in Reliable Computing*, pages 77–104. Kluwer Academic Publishers, Dordrecht. `http://www.tuhh.de/ti3/rump/`.

Seznam tabulek