

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra matematiky



Bakalářská práce
Odhady prvního vlastního čísla
okrajových úloh

Plzeň, 2019

Vedoucí práce:
doc. RNDr. Jiří Benedikt, Ph.D.

Autor:
Jan Půlpán

Poděkování

Rád bych poděkoval panu doc. RNDr. Jiřímu Benediktovi, Ph.D., za návrh tématu bakalářské práce, za její vedení, za rady a připomínky, za otevření nových obzorů a hlavně za trpělivost a podporu, které mi pomohly při vypracování této práce, ale i při studiu samotném.

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni, 22. 5. 2019

Jan Půlpán

Abstrakt

Tématem bakalářské práce jsou odhady prvního vlastního čísla okrajových úloh. Odvodili jsme vztah pro horní odhad pomocí Rayleighova podílu a pro dolní odhad s využitím Piconeho identity. Zaměřili jsme se na konstrukci testovací funkce pomocí kubického splinu. Abychom získali zaručený odhad, použili jsme intervalovou aritmetiku. Řešení demonstrujeme na dvou konkrétních úlohách. Potřebné výpočty byly provedeny v Pythonu.

Klíčová slova: okrajová úloha, první vlastní číslo, zaručený odhad, Rayleighův podíl, Piconeho identita, kubický spline, intervalová aritmetika

Abstract

The topic of this thesis are estimates of principal eigenvalue of boundary value problems. We deduced a formula for upper bound using Rayleigh quotient and for lower bound with Picone identity. We focused on building the test function using the cubic spline. We utilized an interval arithmetic to obtain the guaranteed estimate. Two particular problems are used to demonstrate the calculations, which are programmed in Python.

Keywords: boundary value problem, principal eigenvalue, guaranteed estimate, Rayleigh quotient, Picone identity, cubic spline, interval arithmetic

Použité značení

\mathbb{R}	Množina reálných čísel.
\mathbb{R}^+	Množina kladných reálných čísel.
$C(0,1)$ $C[0,1]$	Prostor spojitých funkcí, definovaných na intervalu $(0,1)$, respektive $[0,1]$.
$C^k(0,1)$ $C^k[0,1]$	Prostor funkcí, definovaných na intervalu $(0,1)$, respektive $[0,1]$, které mají na daném intervalu spojité derivace až do řádu k .
$\text{supp } u$	Nosič (support) funkce $u = u(t)$, definované na intervalu $(0,1)$, je množina $\text{supp } u = \overline{\{t \in (0,1); u(t) \neq 0\}}$.
$C_0^\infty(0,1)$	Prostor funkcí, definovaných na intervalu $(0,1)$, které mají na $(0,1)$ spojitě derivace všech řádů a jejich nosič je podmnožina intervalu $(0,1)$.
$\mathcal{L}^2(0,1)$	Prostor funkcí $u = u(t)$, které jsou definované na $(0,1)$ a $\int_0^1 u(t) ^2 dt$ je konečný.
$L^2(0,1)$	Faktorový prostor $\mathcal{L}^2(0,1) \mathcal{N}_0(0,1)$. \mathcal{N}_0 je množina všech měřitelných funkcí $u = u(t)$ definovaných na $(0,1)$, které jsou skoro všude na $(0,1)$ rovny nule.

$W^{1,2}(0, 1)$ Prostor funkcí $u \in L^2(0, 1)$, jejichž zobecněná derivace u' patří opět do $L^2(0, 1)$.

$W_0^{1,2}(0, 1)$ Prostor funkcí $u \in W^{1,2}(0, 1)$, jejichž limita v krajních bodech je rovna 0.

λ_1 První vlastní číslo.

u_1 První vlastní funkce příslušející k λ_1 .

$\inf_{t \in (0,1)}$ Infimum na intervalu $(0, 1)$.

Obsah

1	Úvod	6
2	Odhady λ_1	7
2.1	Okrajová úloha s parametrem	7
2.2	Horní odhad λ_1	8
2.3	Dolní odhad λ_1	9
3	Výpočty odhadů λ_1	13
3.1	Metoda střelby	13
3.2	Úvod do intervalové aritmetiky	13
3.3	Kubický spline	16
3.4	Úlohy	17
3.5	Metoda výpočtu	17
3.5.1	Horní odhad	19
3.5.2	Dolní odhad	21
3.6	Výsledky	25
4	Závěr	27
	Literatura	28

1 | Úvod

Vlastní čísla okrajových úloh hrají zásadní roli v různých typech aplikací, počínaje kmitáním a konče kvantovou mechanikou. I proto je tento problém zkoumán nejméně tři staletí a v jeho historii se objevují Euler, Lagrange, Cauchy nebo Fourier. I v naší práci narazíme na jména, která se problémem okrajových úloh a s ním spojeným problémem vlastních funkcí a vlastních čísel zabývala. Jacques Charles François Sturm a Joseph Liouville v 19. století pracovali na tom, co dnes známe jako Sturmova-Liouvilleova úloha, kde vlastní čísla a vlastní funkce hrají klíčovou roli. Dalším byl John William Strutt, 3. baron Rayleigh, anglický vědec a nobelista, zabývající se na přelomu 19. a 20. století studiem atmosféry nebo třeba akustikou. A také Mauro Picone, italský matematik, který v roce 1910 přišel s identitou dnes známou jako Piconeho. A nebyl to nikdo jiný než David Hilbert, kdo dal na začátku 20. století vlastním číslům a vlastním funkcím jejich dnešní anglické jméno *eigenvalues* a *eigenfunctions*.

My se v této práci zabýváme jen jedním konkrétním aspektem okrajových úloh s parametrem, a to prvním vlastním číslem. Pro úlohy, které neumíme analyticky vyřešit, je možné získat numerickou aproximaci prvního vlastního čísla. U té však často nejsme schopni zjistit, nebo jen velmi komplikovaně, jak velkou chybu při výpočtu představuje. A proto jsou velmi důležité zaručené horní a dolní odhady, které nám možná neposkytnou přesné řešení, ale zaručí, že hledaná hodnota je určitě pod nebo nad tou vypočtenou.

V kapitole 2 si nadefinujeme úlohu a odvodíme k ní vztah pro horní a dolní odhad prvního vlastního čísla. V kapitole 3 potom budeme hledat vhodnou testovací funkci, kterou bychom do těchto vztahů dosadili. Na dvou konkrétních úlohách si ukážeme, jak výpočet provést.

Všechny výpočty byly naprogramovány v Pythonu s použitím knihoven na numerické výpočty SciPy a NumPy, na intervalovou aritmetiku mpmath, symbolické výpočty SymPy. O grafický výstup se postaral Matplotlib.

2 | Odhady λ_1

2.1 | Okrajová úloha s parametrem

Výchozím bodem bude pro nás následující okrajová úloha.

Mějme obyčejnou diferenciální rovnici 2. řádu, doplněnou o homogenní Dirichletovy okrajové podmínky

$$\begin{cases} -u''(t) = \lambda g(t) u(t), & t \in (0, 1), \\ u(0) = u(1) = 0, \end{cases} \quad (2.1)$$

kde $g \in C[0,1]$ je na intervalu $[0, 1]$ kladná váhová funkce a λ je neznámý parametr.

Takováto úloha se nazývá okrajová úloha s parametrem. Vyřešením úlohy se rozumí nalezení párů parametru $\lambda \in \mathbb{R}$ a funkce $u \not\equiv 0$ na $(0, 1)$, $u \in C^2(0, 1)$, která řeší úlohu (2.1).

Konkrétní hodnota λ se nazývá vlastní číslo, pokud pro něj existuje nenulové řešení úlohy $u(t)$. Této funkci se pak říká vlastní funkce úlohy příslušející k vlastnímu číslu λ .

Úloha (2.1) je jedním z mnoha tvarů tzv. Sturmovy-Liouvilleovy úlohy a pro její řešení platí tedy následující obecně známá věta.

Věta 2.1. *Pro Sturmovu-Liouvilleovu úlohu platí:*

- i. Všechna vlastní čísla úlohy jsou reálná a kladná.*
- ii. Množina všech vlastních čísel $\{\lambda_n\}$ je spočetná a lze ji seřadit do posloupnosti tak, že*

$$\lambda_1 < \lambda_2 < \lambda_3 < \dots$$

- iii. Pro každé vlastní číslo λ_n existuje vlastní funkce u_n s $n - 1$ kořeny v intervalu $(0, 1)$.*

Podle věty 2.1 jsme tedy schopni říci, že pokud má úloha řešení, pak existuje reálné první (nejmenší) vlastní číslo a k němu příslušná první vlastní funkce, která nemá na daném otevřeném intervalu definovaném okrajovými podmínkami žádný kořen. Toto první vlastní číslo budeme značit λ_1 a k němu příslušnou první vlastní funkci u_1 . Protože nemá u_1 na $(0, 1)$ žádné nulové body, můžeme bez újmy na obecnosti od teď uvažovat jen $u_1 > 0$.

Jen velmi málo takovýchto úloh umíme řešit analyticky a většinou nezbyvá nic jiného, než se pokusit o řešení numerické. Pokud je ale naším cílem získat odhad prvního vlastního čísla λ_1 , nemusíme řešení úlohy ani znát.

2.2 | Horní odhad λ_1

Rayleighův podíl je vztah, který je asi obecně více znám pro numerický výpočet vlastních čísel matic. My ho ale použijeme pro horní odhad λ_1 .

Nechť je dvojice $\lambda, u \in C^2(0, 1)$ řešením úlohy (2.1). Z předpokladů $u \in C^2(0, 1)$ a $u(0) = u(1) = 0$, platí také $u \in W_0^{1,2}(0, 1)$. Provedeme tedy následující kroky. Rovnici (2.1) nejprve vynásobíme funkcí u a poté integrujeme obě strany rovnice na intervalu $t \in (0, 1)$, který odpovídá okrajovým podmínkám. Dostaneme

$$\int_0^1 -u''u \, dt = \int_0^1 \lambda g u^2 \, dt.$$

Integrál na levé straně integrujeme per-partes, na pravé straně vytkneme λ před integrál

$$[-u'u]_0^1 + \int_0^1 u'u' \, dt = \lambda \int_0^1 g u^2 \, dt.$$

První část levé strany rovnice je díky okrajovým podmínkám rovna nule. Z rovnice tak můžeme vyjádřit

$$\lambda = \frac{\int_0^1 (u')^2 \, dt}{\int_0^1 g u^2 \, dt}.$$

Výraz na pravé straně rovnosti je již zmíněný Rayleighův podíl a značit ho budeme R_v , kde proměnnou je vlastní funkce úlohy (2.1), tedy i první vlastní funkce u_1 příslušející k prvnímu vlastnímu číslu λ_1 . Co kdybychom ale do Rayleighova podílu dosadili místo u_1 libovolnou funkci $v \in W_0^{1,2}(0, 1)$? Lze dokázat, že první vlastní číslo λ_1 úlohy je rovno infimu z Rayleighova podílu přes funkce

$v \in W_0^{1,2}(0,1)$, $v \neq 0$. Funkci v budeme nazývat testovací funkcí.

$$\lambda_1 = \inf_{\substack{v \in W_0^{1,2}(0,1) \\ v \neq 0}} R_v = \inf_{\substack{v \in W_0^{1,2}(0,1) \\ v \neq 0}} \frac{\int_0^1 (v')^2 dt}{\int_0^1 gv^2 dt}. \quad (2.2)$$

Ze vztahu (2.2) okamžitě vyplývá následující věta.

Věta 2.2 (Horní odhad λ_1). *Nechť λ_1 je první vlastní číslo úlohy (2.1).*

Pro libovolnou funkci $v \in W_0^{1,2}(0,1)$, $v \neq 0$, pak platí

$$\lambda_1 \leq \frac{\int_0^1 (v')^2 dt}{\int_0^1 gv^2 dt}. \quad (2.3)$$

Poznámka 2.1. Rovnost v (2.3) nastane, když $v = ku_1$, $k \neq 0$.

2.3 | Dolní odhad λ_1

Věty a jejich důkazy v této kapitole jsou inspirovány článkem [1].

Abychom byli schopni uvést vztah pro výpočet dolního odhadu λ_1 úlohy (2.1) a provést jeho důkaz, musíme nejprve formulovat a dokázat následující věty.

Věta 2.3. *Nechť u, v jsou funkce, $u \geq 0, v > 0, u, v \in C^1(0,1)$. Potom u, v splňují Piconeho identitu*

$$(u')^2 + \frac{u^2}{v^2}(v')^2 - 2\frac{u}{v}u'v' = (u')^2 - \left(\frac{u^2}{v}\right)' v' \geq 0. \quad (2.4)$$

Označíme-li levou stranu rovnosti $L(u, v)$ a pravou stranu $R(u, v)$, můžeme vztah (2.4) přepsat ve tvaru

$$L(u, v) = R(u, v) \geq 0. \quad (2.5)$$

Platí, že $L(u, v) = 0$ na intervalu $(0,1)$ právě tehdy, když $\left(\frac{u}{v}\right)' = 0$ na $(0,1)$. Z toho vyplývá, že $u = kv$ pro nějakou konstantu $k \in \mathbb{R}$.

Důkaz. Nejprve dokažme rovnost v Piconeho identitě pomocí derivace podílu $\frac{u^2}{v}$ v $R(u, v)$.

$$R(u, v) = (u')^2 - \left(\frac{u^2}{v}\right)' v' = (u')^2 - \frac{2uvu' - u^2v'}{v^2} v' = (u')^2 + \frac{u^2}{v^2}(v')^2 - 2\frac{u}{v}u'v'.$$

Použijeme-li substituci $\alpha = u'$ a $\beta = \frac{uv'}{v}$, tak platí

$$L(u, v) = \alpha^2 - 2\alpha\beta + \beta^2 = (\alpha - \beta)^2.$$

Tento výraz je díky druhé mocnině vždy nezáporný a v Piconeho identitě tedy platí $L(u, v) = R(u, v) \geq 0$.

Rovnost $L(u, v) = 0$ platí právě tehdy, když $\alpha = \beta$. Dosazením za funkce α, β získáme $u' = \frac{uv'}{v}$, po vynásobení $u'v - uv' = 0$. Celý výraz vydělíme v^2 a již je zřejmé, že $\frac{u'v - uv'}{v^2} = \left(\frac{u}{v}\right)' = 0$. Rovnost integrujeme a vyjádřením získáváme $u = kv, k \in \mathbb{R}$. ■

Nyní uvažujme úlohu (2.1) a předpokládejme existenci prvního vlastního čísla a k němu příslušné první vlastní funkce.

Věta 2.4. *Předpokládejme, že nějaká funkce $v \in C^2(0, 1), v > 0$ na $(0, 1)$ splňuje $-v'' \geq \lambda gv, g > 0$ na $(0, 1), 0 < \lambda \in \mathbb{R}$. Potom pro libovolné $u \geq 0, u \in C^2(0, 1), u(0) = u(1) = 0$, platí*

$$\int_0^1 (u')^2 dt \geq \lambda \int_0^1 gu^2 dt \quad (2.6)$$

a navíc $\lambda_1 \geq \lambda$.

Důkaz. Necht' $[\alpha, \beta], 0 < \alpha < \beta < 1$, je interval, to jest $[\alpha, \beta] \subset (0, 1)$. Zvolme funkci $\varphi \in C_0^\infty(0, 1), \varphi \geq 0, \varphi = 0$ na $(0, 1) \setminus [\alpha, \beta]$. Potom platí

$$\begin{aligned} 0 &\stackrel{\text{(I.)}}{\leq} \int_\alpha^\beta L(\varphi, v) dt = \int_\alpha^\beta R(\varphi, v) dt = \\ &= \int_\alpha^\beta (\varphi')^2 dt - \int_\alpha^\beta \left(\frac{\varphi^2}{v}\right)' v' dt \stackrel{\text{(II.)}}{=} \int_\alpha^\beta (\varphi')^2 dt + \int_\alpha^\beta \left(\frac{\varphi^2}{v}\right) v'' dt \leq \\ &\stackrel{\text{(III.)}}{\leq} \int_\alpha^\beta (\varphi')^2 dt - \lambda \int_\alpha^\beta g\varphi^2 dt \stackrel{\text{(IV.)}}{=} \int_0^1 (\varphi')^2 dt - \lambda \int_0^1 g\varphi^2 dt. \quad (2.7) \end{aligned}$$

Označené kroky v (2.7) si zaslouží detailnější vysvětlení.

(I.) Platí díky větě (2.3).

(II.) Integrací per-partes dostáváme

$$\int_\alpha^\beta \left(\frac{\varphi^2}{v}\right)' v' dt = \left| \begin{array}{ll} x = v' & y' = \left(\frac{\varphi^2}{v}\right)' \\ x' = v'' & y = \frac{\varphi^2}{v} \end{array} \right| = \left[\frac{\varphi^2}{v} v'\right]_\alpha^\beta - \int_\alpha^\beta v'' \frac{\varphi^2}{v} dt.$$

Protože $\varphi \in C_0^\infty(0, 1)$, $\varphi \geq 0$, $\varphi = 0$ na $(0, 1) \setminus [\alpha, \beta]$, platí $\left[\frac{\varphi^2}{v}v'\right]_\alpha^\beta = 0$.

(III.) Z předpokladu $-v'' \geq \lambda gv$ vyjádříme $\frac{v''}{v} \leq -\lambda g$. Konstantu poté můžeme vytknout před integrál.

(IV.) Protože je funkce φ na $(0, 1) \setminus [\alpha, \beta]$ rovna nule, a tím pádem i φ' je nulová, můžeme oba integrály rozšířit na celý interval $(0, 1)$ a jejich hodnota se nezmění.

Pro každou funkci $u \in C^2(0, 1)$, $u(0) = u(1) = 0$, platí také $u \in W_0^{1,2}(0, 1)$. Množina $C_0^\infty(0, 1)$ je hustá ve $W_0^{1,2}(0, 1)$ vzhledem k normě $W^{1,2}(0, 1)$, a proto existuje posloupnost funkcí $\varphi_n \in C_0^\infty(0, 1)$, $\varphi_n \xrightarrow{W^{1,2}(0,1)} u$. Protože integrály v (2.6) jsou spojité vzhledem k normě $W^{1,2}(0, 1)$, můžeme psát

$$0 \leq \int_0^1 (u')^2 dt - \lambda \int_0^1 gu^2 dt. \quad (2.8)$$

Tím jsme dokázali (2.6) a můžeme dokázat $\lambda_1 \geq \lambda$.

Za u zvolme první vlastní funkci u_1 . Podle věty 2.2 a poznámky 2.1 platí

$$\lambda_1 = \frac{\int_0^1 (u_1')^2 dt}{\int_0^1 gu_1^2 dt}. \quad (2.9)$$

Do (2.6) dosadíme $u = u_1$ a vyjádříme λ

$$\frac{\int_0^1 (u_1')^2 dt}{\int_0^1 gu_1^2 dt} \geq \lambda.$$

Dosazením λ_1 z (2.9) dostáváme $\lambda_1 \geq \lambda$. ■

Poznámka 2.2. Pokud za u a v zvolíme kladný násobek u_1 a za λ zvolíme λ_1 , bude v požadované nerovnosti $-v'' \geq \lambda gv$ platit rovnost, stejně tak i v (2.6).

Nyní již můžeme formulovat a dokázat větu, která nám umožní vypočítat dolní odhady prvního vlastního čísla úlohy (2.1).

Věta 2.5 (Dolní odhad λ_1). *Nechť λ_1 je první vlastní číslo úlohy (2.1). Nechť v je testovací funkce, $v \in C^2(0, 1)$, $v(t) > 0$, $t \in (0, 1)$. Potom pro λ_1 platí*

$$\lambda_1 \geq \inf_{(0,1)} \frac{-v''}{gv}. \quad (2.10)$$

Důkaz. Určitě platí

$$\frac{-v''}{gv} \geq \inf_{(0,1)} \frac{-v''}{gv}. \quad (2.11)$$

Zvolíme $\lambda = \inf_{(0,1)} \frac{-v''}{gv}$. Z této volby a z (2.11) pak platí $-\frac{v''}{gv} \geq \lambda$, a tedy $-v'' \geq \lambda gv$. Tím jsme dokázali předpoklad z věty 2.4.

Z věty 2.4 víme, že $\lambda_1 \geq \lambda$, a proto díky volbě λ platí

$$\lambda_1 \geq \inf_{(0,1)} \frac{-v''}{gv}.$$

■

Poznámka 2.3. Rovnost v (2.10) nastane, když $v = cu_1$, kde u_1 je první vlastní funkce příslušná k prvnímu vlastnímu číslu λ_1 a c je libovolná konstanta $c \in \mathbb{R}^+$.

V dalším textu budeme podíl $-\frac{v''}{gv}$ pro jednoduchost zápisu značit P_v .

3 | Výpočty odhadů λ_1

To, jak se budeme schopni výpočtem přiblížit pomocí horního a dolního odhadu ke skutečné hodnotě λ_1 , závisí největší měrou na výběru testovací funkce. Optimální by bylo použít první vlastní funkci u_1 . Pokud bychom ale u_1 znali, znali bychom i λ_1 . V případech, kdy nejsme schopni u_1 analyticky najít, můžeme použít její numerickou aproximaci. V dalších výpočtech musíme už ale zohlednit zaokrouhlování způsobené počítačovou aritmetikou. Potřebujeme získat zaručený odhad takový, který například u horního odhadu zajistí, že získaná hodnota nebude menší než skutečné λ_1 . Kdybychom pro výpočet použili standardní počítačovou aritmetiku, tak se to stát může. Proto pro naše výpočty použijeme aritmetiku intervalovou. Body, které získáme numerickým výpočtem v počítačové aritmetice, interpolujeme s použitím intervalové aritmetiky kubickým splinem a ten následně znovu intervalově dosadíme do vztahu (2.3) pro horní odhad, respektive (2.10) pro dolní odhad.

Než ovšem budeme schopni sestavit konkrétní algoritmus výpočtu, popišme si obecně používanou numerickou metodu, intervalovou aritmetiku a výpočet kubického splinu.

3.1 | Metoda střelby

Numerickou aproximaci u_1 získáme metodou střelby. Ta předpokládá převod okrajové úlohy na posloupnost počátečních úloh ve formě soustavy dvou rovnic prvního řádu, kterou řešíme nějakou standardní metodou pro řešení počáteční úlohy. V našich výpočtech vybereme metodu RK45, která kombinuje metody Runge-Kutta 4. a 5. řádu. Pomocí bisekce pak hledáme hodnotu prvního vlastního čísla λ_1 tak, aby řešení splňovalo okrajové podmínky původní úlohy.

3.2 | Úvod do intervalové aritmetiky

Numerické výpočty na počítači v rámci počítačové aritmetiky vedou kvůli zaokrouhlování na množinu strojově reprezentovatelných čísel jen k přibližným vý-

sledkům, v některých případech dokonce k výsledkům zcela nesprávným. Příkladem může být např. vyjádření hodnoty funkce dvou proměnných, které popsal Siegfried Rump v článku [7] a posléze bylo upřesněno v [8].

Pokud necháme počítač ve standardní IEEE 754 aritmetice vyjádřit hodnotu funkce

$$f(x,y) = (333,75 - x^2)y^6 + x^2(11x^2y^2 - 121y^4 - 2) + 5,5y^8 + \frac{x}{2y}, \quad (3.1)$$

dostaneme v bodě $(x,y) = (77617, 33096)$ výsledek 1,1726039400531787. Přibližná hodnota funkce v tomto bodě je ale $-0,827396059946821$, jak se můžeme přesvědčit například symbolickým výpočtem. Pomocí výpočtů v aritmetice s libovolnou přesností můžeme kontrolovat přesnost numerických výpočtů, ale neexistuje žádný univerzální návod, jak se v podobných případech dostat k alespoň přibližnému výsledku. V tab. 3.1 je několik hodnot pro různé hodnoty přesnosti (počet platných desetinných míst) pro funkci (3.1).

Tabulka 3.1: Hodnota funkce (3.1) pro $(x,y) = (77617, 33096)$ pro různé přesnosti při použití aritmetiky s libovolnou přesností.

přesnost	hodnota $f(x, y)$
10	$-7,737125245534 \times 10^{25}$
15	1,1726039400531787
25	$-137438953470,8273960599468211$
36	$-0,827396059946821368141165095479816292005$

Teprve nastavení přesnosti na 36 desetinných míst umožní získat přibližný výsledek s velmi malou chybou. Jakýkoliv výsledek vypočtený pomocí aritmetiky s menším počtem desetinných míst je zatížen ohromnou chybou.

Pokud není možné naše výpočty dělat symbolicky, je vhodné v těchto případech využít intervalovou aritmetiku. Ta místo s jednotlivými čísly (dané přesností) počítá s uzavřenými intervaly, jejichž horní a dolní hranice jsou čísla, která dokáže počítač reprezentovat. Hodnota, s kterou počítáme, leží pak vždy v tomto intervalu. Nad intervaly je definováno intervalové rozšíření $\oplus, \ominus, \otimes, \oslash$ operací sčítání, odčítání, násobení, dělení.

Jakých hodnot bude nabývat funkce (3.1) při použití intervalové aritmetiky, opět pro různé přesnosti, je vidět v tab. 3.2.

Tabulka 3.2: Hodnota funkce (3.1) pro $(x,y) = (77617, 33096)$ pro různé přesnosti při použití intervalové aritmetiky.

přesnost	hodnota $f(x, y)$
10	$[-1,547425049107 \times 10^{26}, 3,094850098258 \times 10^{26}]$
15	$[-3,5417748621522339 \times 10^{21}, 3,5417748621522344 \times 10^{21}]$
25	$[-137438953470,8273960599468229, 412316860417,1726039400531789]$
36	$[-0,827396059946821368141165095479816292005, -0,827396059946821368141165095479816291817]$

Hledaná hodnota leží vždy ve výsledném intervalu. Intervalová aritmetika nám nedává výsledky přesnější než výpočty s libovolnou přesností, ve většině případů spíše naopak, ale výsledky jsou zaručené a hledaná hodnota v intervalu vždy leží. Této vlastnosti využijeme při výpočtu zaručených horních a dolních odhadů.

V rámci našich výpočtů horního odhadu budeme muset intervalově vypočítat určitý integrál dané funkce. Jednoduchá a přitom i dostatečně přesná metoda je založena na Riemannových součtech, viz [2]. Interval, na kterém integrál počítáme, rozdělíme ekvidistantně na dostatečný počet intervalů m , na nichž pak intervalově vyjádříme hodnotu funkce $u(t)$.

$$\int_{\alpha}^{\beta} u(t) dt \in \sum_{i=0}^{m-1} \left(\left[\min_{t \in \mathcal{I}_i} u(t), \overline{\max_{t \in \mathcal{I}_i} u(t)} \right] \otimes \mathcal{I}_h \right), \quad (3.2)$$

$$\mathcal{I}_i = \left[\underline{\alpha + i(\beta - \alpha)/m}, \overline{\alpha + (i + 1)(\beta - \alpha)/m} \right],$$

$$\mathcal{I}_h = \left[\underline{(\beta - \alpha)/m}, \overline{(\beta - \alpha)/m} \right].$$

Intervaly značené $[\underline{a}, \overline{b}]$ reprezentují vyjádření v intervalové aritmetice. Hodnota, s kterou počítáme, je v takovém intervalu vždy obsažena. Horní mez $\underline{a} \leq a$ a dolní mez $\overline{b} \geq b$ jsou strojově reprezentovatelná čísla ve standardní aritmetice IEEE 754. Sumu pak chápeme jako součet intervalů operátorem \oplus .

Více o intervalové aritmetice lze nalézt například v [2] a [7], případně v dokumentaci k [9].

3.3 | Kubický spline

Podle věty 2.5 potřebujeme testovací funkci třídy C^2 , proto volíme kubický spline. Počítání se splinem vyššího než 3. stupně je náročnější a nepřináší žádné další výhody.

Následující postup výpočtu kubického splinu $s \in C^2(0, 1)$ je založen na textech z přednášek [4] a [5].

Jednotlivé oblouky kubického splinu mají formu

$$s_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i.$$

Při n obloucích interpolační křivky potřebujeme $4n$ koeficientů a_i, b_i, c_i, d_i , $i = 0, 1, \dots, n-1$. Potřebujeme tedy $4n$ podmínek, abychom byli schopni koeficienty vypočítat. K dispozici máme následující podmínky.

1. $n + 1$ interpolačních podmínek (hodnoty v uzlových bodech, získané numerickým výpočtem).
2. $n - 1$ podmínek na spojitost funkce s .
3. $2(n - 1)$ podmínek na spojitost 1. derivace a 2. derivace funkce s .

Celkem dostáváme $4n - 2$ podmínek pro $4n$ neznámých, musíme tedy dvě podmínky doplnit. V našem případě budeme přidávat podmínky na hodnoty druhých derivací v počátečním a koncovém bodě kubického splinu.

Pro výpočet koeficientů je dobře známa metoda, kdy z výpočtených hodnot druhých derivací v jednotlivých uzlech stanovíme potřebné koeficienty.

Druhé derivace σ_i dostaneme vyřešením soustavy lineárních rovnic

$$\begin{pmatrix} 4 & 1 & 0 & \dots & 0 \\ 1 & 4 & 1 & \dots & 0 \\ & & \ddots & & \\ 0 & \dots & 1 & 4 & 1 \\ 0 & \dots & 0 & 1 & 4 \end{pmatrix} \begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \vdots \\ \sigma_{n-1} \end{pmatrix} = \frac{6}{h^2} \begin{pmatrix} y_2 - 2y_1 + y_0 \\ y_3 - 2y_2 + y_1 \\ \vdots \\ \vdots \\ y_n - 2y_{n-1} + y_{n-2} \end{pmatrix}, \quad (3.3)$$

kde h je vzdálenost dvou po sobě jdoucích hodnot parametru t při ekvidistantním dělení a σ_i jsou hodnoty druhých derivací ve vnitřních bodech. Po nastavení hodnot druhých derivací σ_0 a σ_n v krajních bodech kubického splinu vypočteme jednotlivé koeficienty podle následujících vztahů.

$$\begin{aligned}
a_i &= \frac{\sigma_{i+1} - \sigma_i}{6h}, \\
b_i &= \frac{\sigma_i}{2}, \\
c_i &= \frac{y_{i+1} - y_i}{h} - h \frac{2\sigma_i + \sigma_{i+1}}{6}, \\
d_i &= y_i.
\end{aligned} \tag{3.4}$$

3.4 | Úlohy

Na základě úlohy (2.1) definujme dvě konkrétní okrajové úlohy, na kterých ukážeme výpočet horního a dolního odhadu λ_1 pro různé nastavení jednotlivých parametrů výpočtu. Pro první úlohu volíme váhovou funkci $g(t) \equiv 1$

$$-u'' = \lambda u, \quad t \in (0, 1), \quad u(0) = u(1) = 0, \tag{3.5}$$

pro druhou úlohu váhovou funkci $g(t) = 1 + t$

$$-u'' = \lambda(1 + t)u, \quad t \in (0, 1), \quad u(0) = u(1) = 0. \tag{3.6}$$

Úloha (3.5) je všeobecně známá školní úloha, která je řešitelná analyticky. Řešením je posloupnost vlastních čísel $\lambda_k = (k\pi)^2$ a k nim příslušející posloupnost vlastních funkcí $u_k = \sin k\pi t, t \in (0, 1), k \in \mathbb{N}$. Budeme tedy schopni snadno porovnat vypočtený odhad s přesným řešením. Úlohu (3.6) již analyticky vyřešit neumíme a správnost odhadu tedy musíme posoudit pomocí jiných kritérií, např. porovnáním s přibližnou hodnotou λ_1 vypočtenou numericky.

3.5 | Metoda výpočtu

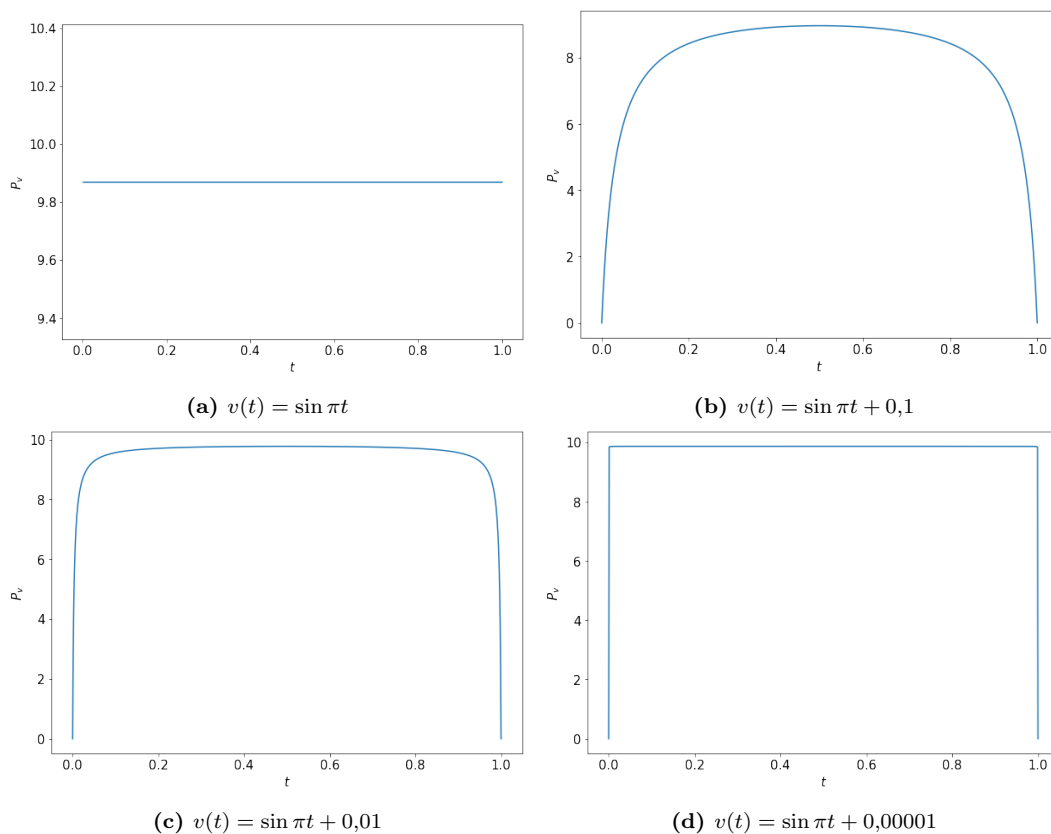
Naším hlavním cílem je najít vhodnou testovací funkci v , která bude vyhovovat předpokladům vět 2.2 a 2.5 a zároveň umožní co nejpřesnější odhady λ_1 . Algoritmus výpočtu jak horního tak i dolního odhadu nejprve numericky nalezneme uzlové body, pomocí kterých sestrojíme kubický spline a ten nakonec dosadíme do vztahů pro horní a dolní odhad.

1. Vypočteme $n + 1$ bodů numerické aproximace první vlastní funkce u_1 metodou střelby. Tento výpočet je možné provést ve standardní počítačové aritmetice. Intervalový výpočet by sice možný byl, ale jen by celý algoritmus zesložil a zpomalil jeho běh. Uzlové body získané počítačovou aritmetikou nám i tak umožní sestavit kubický spline, který bude odpovídat předpokladům na testovací funkci v .

2. Aproximaci u_1 získanou numerickým výpočtem použijeme jako uzlové body kubického splinu o n částech (obloučích), který je těmito body jednoznačně určen a splňuje podmínku spojitosti C^2 . Koeficienty splinu již ale musíme získat pomocí intervalové aritmetiky. Pokud bychom to neudělali, výsledný spline nemusí být kvůli nepřesnosti počítačové aritmetiky ani spojitý. Hlavně bychom ale nebyli schopni, kvůli zaokrouhlování v počítačové aritmetice, získat zaručený odhad.

Pro výpočet dolního odhadu musíme navíc posunout body získané numericky o malou kladnou konstantu δ . Při výpočtu v intervalové aritmetice může interval reprezentující hodnoty funkce v na některém z krajů intervalu $(0, 1)$ obsahovat uvnitř 0. To by ovšem znamenalo, že intervalová hodnota funkce $P_v = -v''/gv$ by byla rovna $(-\infty, \infty)$, což není použitelný dolní odhad.

Přičtení δ k testovací funkci v změní ovšem průběh funkce P_v . Na obr. 3.1a je průběh funkce P_v v úloze (3.5), pokud za testovací funkci v zvolíme přímo první vlastní funkci $\sin \pi t$ a žádné δ nepřičteme. Na obr. 3.1b-d je stejná funkce P_v , k $v(t) = \sin \pi t$ přičítáme ale různě velké δ .



Obrázek 3.1: Průběh P_v pro $v(t) = \sin \pi t$ a různé hodnoty δ .

Pokud k testovací funkci v přičteme libovolně malé kladné δ , bude funkce P_v na krajích konvergovat k 0. Hodnota $\inf_{(0,1)} P_v$ tak vždy vyjde 0. Proto

při výpočtu kubického splinu nastavíme druhé derivace σ_0, σ_n v krajních bodech funkce v na zápornou hodnotu. Ta zaručí, že se kraje funkce P_v „zvednou“ a přitom nenaruší předpoklady na testovací funkci v .

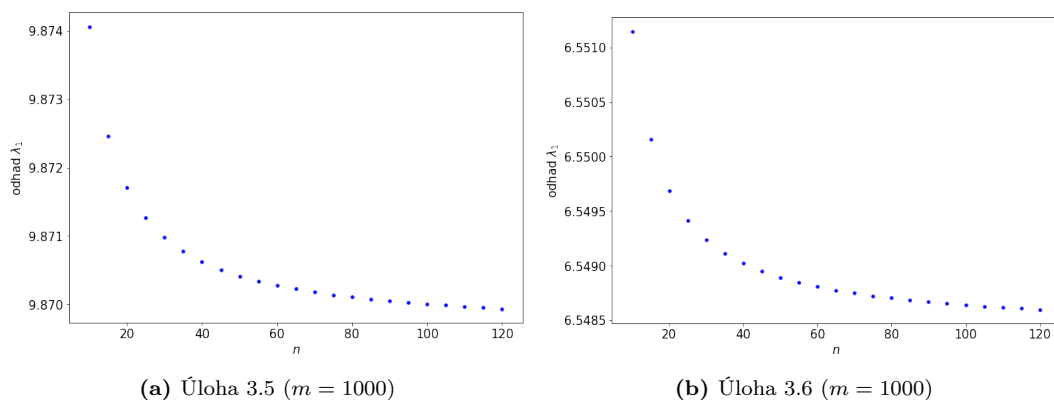
3. Testovací funkce v nyní vyhovuje podmínkám vět 2.2 a 2.5. Můžeme ji tedy dosadit do vztahů pro výpočet horního (2.3) a dolního (2.10) odhadu. Kvůli časové náročnosti výpočtu inverzní matice \mathbf{A}^{-1} není možné vypočítat kubický spline s velkým množstvím oblouků, přibližně $n > 300$. Rozdělíme tedy každý oblouk na m intervalů a ty teprve dosadíme do intervalového výpočtu odhadů λ_1 . Menší dosazované intervaly budou v rámci intervalové aritmetiky znamenat přesnější odhad.

Je vidět, že na přesnost výpočtu odhadů λ_1 mají vliv různé parametry. Většinou platí, že zjemnění dělení, nebo zpřesnění jiného parametru, zlepší i výsledný odhad. Výpočet horního odhadu pomocí Rayleighova podílu je velmi přímočarý. Pro dolní odhad musíme ale pečlivě volit hodnotu posunutí δ a hodnoty 2. derivace σ_0, σ_n v krajních bodech kubického splinu. Zde je popis kompletní metody výpočtu, zvlášť pro horní a poté dolní odhad.

3.5.1 Horní odhad

1. Metodou střelby (viz kapitola 3.1) získáme $n+1$ bodů aproximace u_1 . Zastavovací podmínka pro výpočet λ_1 pomocí bisekce je 10^{-12} . Počáteční úlohu řešíme funkcí `scipy.integrate.solve_ivp` z Python balíčku Scipy.
2. Body získané v kroku 1 použijeme jako uzlové body kubického splinu, naší testovací funkce v .

Pro horní odhad platí, že vyšší počet n oblouků splinu odhad zpřesní. To je ilustrováno na obr. 3.2. V závorce pod grafy jsou vždy uvedeny hodnoty dalších parametrů využitých při výpočtu.



Obrázek 3.2: Závislost horního odhadu na počtu oblouků kubického splinu n .

Získáme koeficienty kubického splinu. Řešíme soustavu lineárních rovnic (3.3), kterou můžeme reprezentovat ve zjednodušené formě $\mathbf{A} \cdot \boldsymbol{\sigma} = \mathbf{b}$.

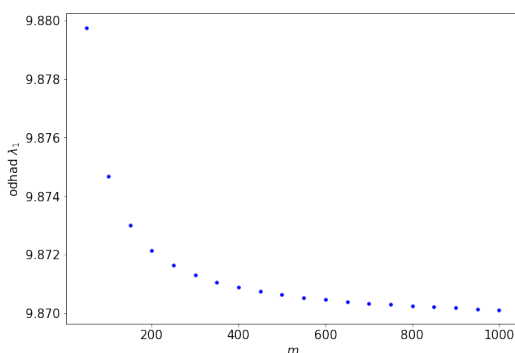
- Matici \mathbf{A} sestavíme symbolicky.
- Symbolicky vypočteme inverzní matici \mathbf{A}^{-1} . Protože je matice \mathbf{A} řídká, můžeme použít LDL metodu, a tím urychlit výpočet. Navíc se matice \mathbf{A} pro různé úlohy nemění, je závislá jen na počtu oblouků n . Proto lze \mathbf{A}^{-1} předpočítat.
- Od teď již musíme vše kvůli zaručenému odhadu počítat intervalově. Podle (3.3) sestavíme vektor pravé strany \mathbf{b} . Protože jsme v numerické metodě použili ekvidistantní kroky, volíme h rovno diferenci uzlů splinu.
- Intervalově vyřešíme rovnici $\mathbf{A} \cdot \boldsymbol{\sigma} = \mathbf{b}$. Stačí jen intervalově vynásobit $\boldsymbol{\sigma} = \mathbf{A}^{-1} \cdot \mathbf{b}$. Tím získáme hodnoty 2. derivací ve vnitřních uzlových bodech. Hodnoty σ_0, σ_n v krajních bodech nastavíme na 0.
- Koeficienty jednotlivých oblouků splinu vypočteme intervalově podle vzorců (3.4).

3. Dosadíme testovací funkci v do vztahu (2.3) pro horní odhad.

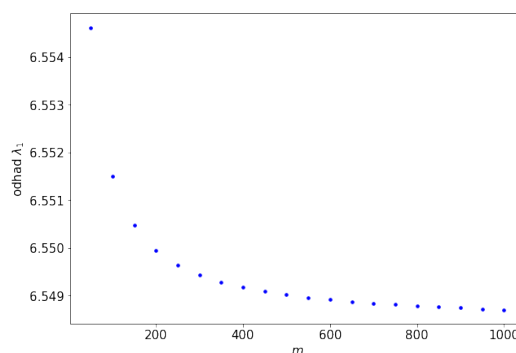
- Intervalově vypočítáme 1. derivaci testovací funkce v . Protože se jedná o po částech polynom třetího stupně, který je C^2 spojitý, stačí jen intervalově přenásobit koeficienty.

$$(at^3 + bt^2 + ct + d)' = 3at^2 + 2bt + c.$$

- Každý oblouk splinu rozdělíme dále na m intervalů. Už jsme uvedli v předchozím popisu, že jemnější dělení oblouků znamená přesnější hodnotu horního odhadu, jak je vidět i z obr. 3.3.



(a) Úloha 3.5 ($n = 80$)



(b) Úloha 3.6 ($n = 80$)

Obrázek 3.3: Závislost horního odhadu λ_1 na hodnotě m .

- Podle vztahu (3.2) vypočítáme intervalově oba integrály $\int_0^1 (v')^2 dt$ a $\int_0^1 v^2 dt$.

- (d) Hodnoty obou integrálů dosadíme do vztahu (2.3). Výsledkem je interval $[\underline{R}_v, \overline{R}_v]$ reprezentující horní odhad λ_1 .
- (e) Pro zaručený horní odhad volíme horní mez $\overline{R}_v \geq R_v \geq \lambda_1$.

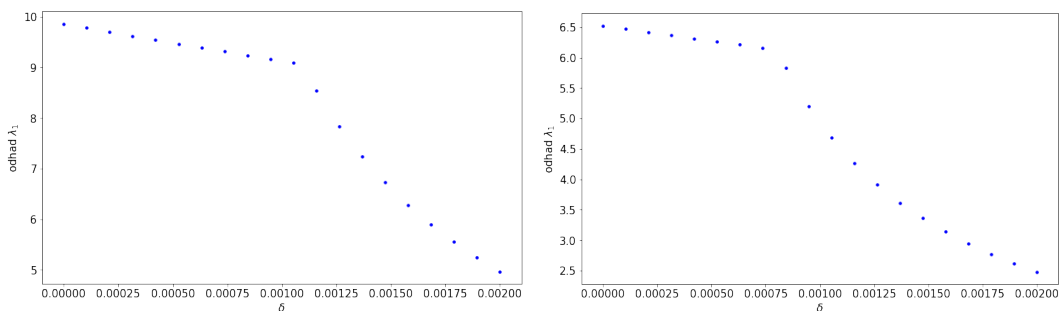
3.5.2 Dolní odhad

Výpočet dolního odhadu provedeme v podobných krocích jako výpočet odhadu horního.

1. Metodou střelby získáme $n + 1$ bodů aproximace u_1 . K výsledku přičteme malou kladnou konstantu δ . Tu volíme co nejmenší, abychom co nejméně ovlivnili dolní odhad. Na obr. 3.4 je závislost hodnoty odhadu λ_1 na velikosti δ .

Z pravé části grafu je zřejmé, že příliš velké δ pro zvolený počet oblouků n odhad významně zhorší. Krajiní intervaly totiž zasahují do poklesu funkce P_v , jak jsme ji ukázali na obrázcích 3.1.

Nesmíme ale δ zvolit příliš malé vzhledem k dělení na n oblouků. Již dříve jsme ukázali, že v tom případě by výsledný dolní odhad byl $-\infty$.

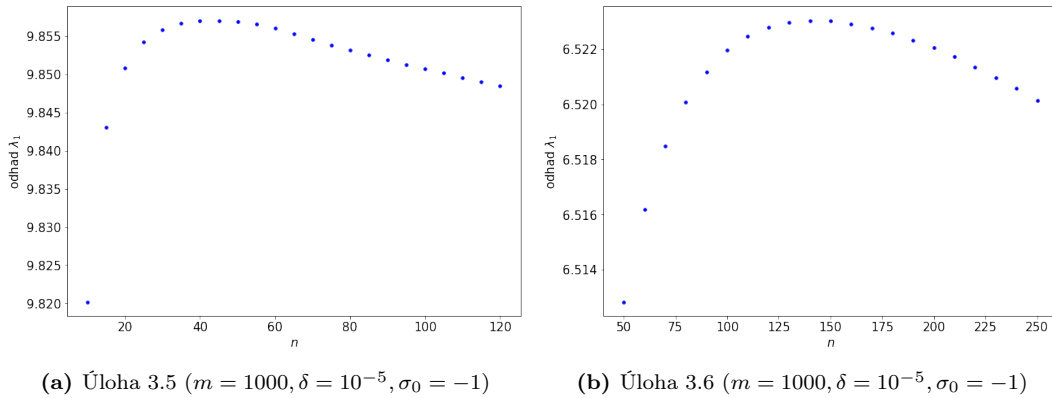


(a) Úloha 3.5 ($n = 80, m = 1000, \sigma_0 = -10^{-2}$)

(b) Úloha 3.6 ($n = 80, m = 1000, \sigma_0 = -10^{-2}$)

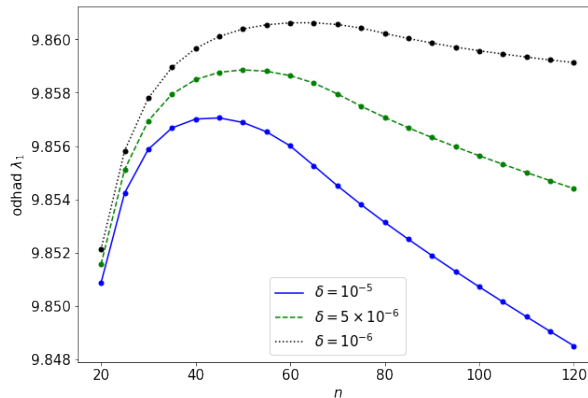
Obrázek 3.4: Závislost dolního odhadu λ_1 na hodnotě δ .

2. Velikost δ má ovšem ještě jeden zásadní vliv. Narozdíl od horního odhadu neplatí, že dělení splinu na více oblouků výsledný dolní odhad λ_1 zpřesní. Na obr. 3.5 je závislost hodnoty dolního odhadu λ_1 na počtu oblouků splinu. Je patrné, že hodnota n má své optimum.



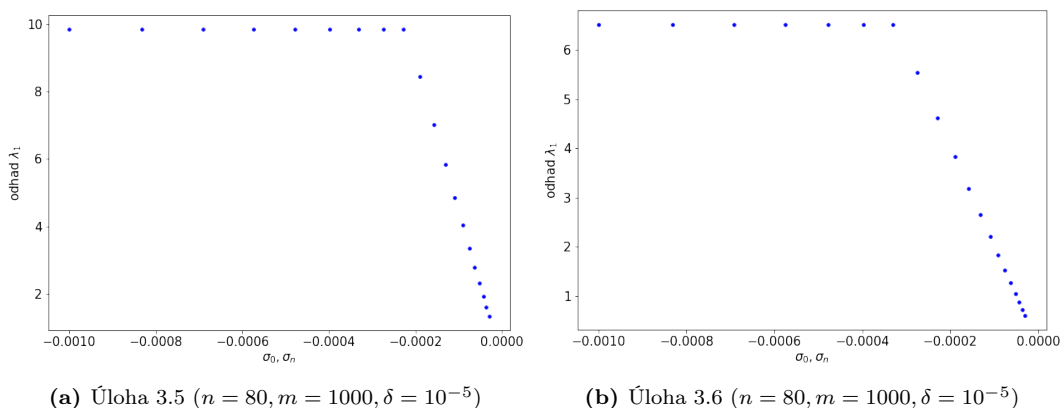
Obrázek 3.5: Závislost dolního odhadu λ_1 na počtu oblouků n .

Toto optimum je závislé právě na hodnotě δ . Pokud n zvolíme moc velké, a tím pádem krajní oblouky budou moc malé, záporné hodnoty σ_0 a σ_n zvednou hodnotu v krajních intervalech dostatečně, ale hodnoty P_v v druhém a předposledním uzlovém bodě funkce v budou stále moc nízké. Proto se hodnota dolního odhadu λ_1 bude s rostoucím počtem oblouků n dále zmenšovat. Na obr. 3.6 je průběh závislosti odhadu λ_1 na počtu oblouků n pro tři různá δ .



Obrázek 3.6: Úloha 3.5 - závislost dolního odhadu λ_1 na počtu oblouků n pro různá δ ($m = 1000, \sigma_0 = -1$).

Koeficienty kubického splinu vypočteme stejným algoritmem jako v případě horního odhadu, jen musíme nastavit zápornou hodnotu σ_0, σ_n . Při konstantních ostatních parametrech na tom, jakou hodnotu toto číslo bude mít, prakticky nezáleží (kromě krajních případů hodnoty σ_0, σ_n blíží se k 0 zleva). Od určité hodnoty σ_0, σ_n , a tím dostatečné korekce funkce P_v , je výsledný odhad λ_1 identický, viz obr. 3.7. Proto zpravidla volíme $\sigma_0, \sigma_n = -1$.



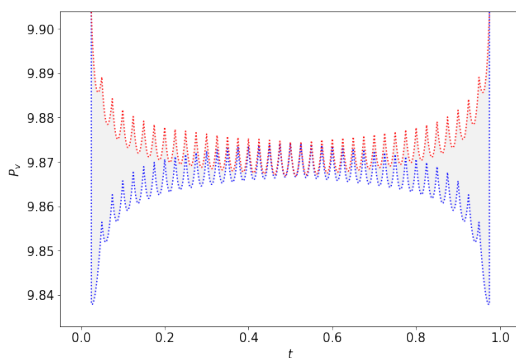
Obrázek 3.7: Závislost dolního odhadu na hodnotě σ_0, σ_n .

3. Dosadíme testovací funkci v do vztahu (2.10) pro dolní odhad.

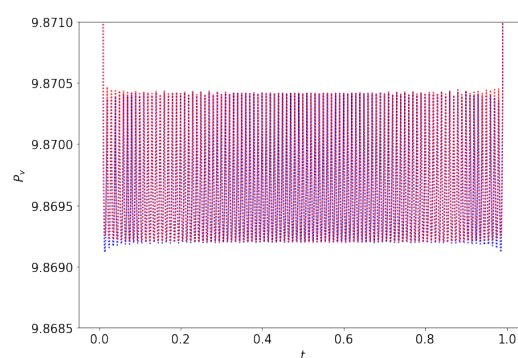
- (a) Intervalově vypočteme 2. derivaci testovací funkce v . Stejně jako u 1. derivace stačí jen přenásobit (intervalově) koeficienty jednotlivých oblouků splinu.

$$(at^3 + bt^2 + ct + d)'' = 6at + 2b.$$

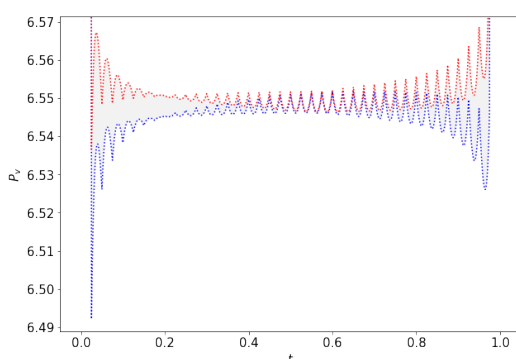
- (b) Každý z n oblouků funkce v rozdělíme na m intervalů a pro každý takto vzniklý interval vyjádříme hodnotu podílu P_v intervalově. Výsledkem jsou intervaly $\mathcal{I}_i, i \in \{1, \dots, n \times m\}$. Na obr. 3.8 je zobrazen intervalový průběh P_v pro obě naše úlohy a různé parametry výpočtu.



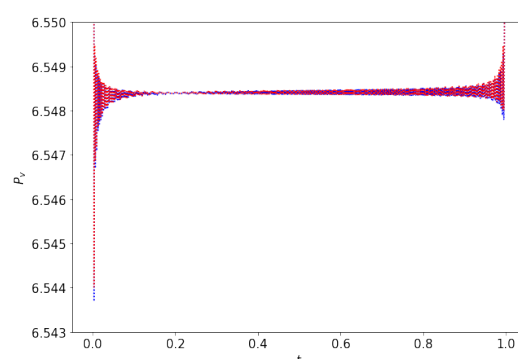
(a) Úloha (3.5)
($n = 40, m = 300, \delta = 10^{-5}, \sigma_0 = -1$)



(b) Úloha (3.5)
($n = 100, m = 10^5, \delta = 10^{-9}, \sigma_0 = -1$)



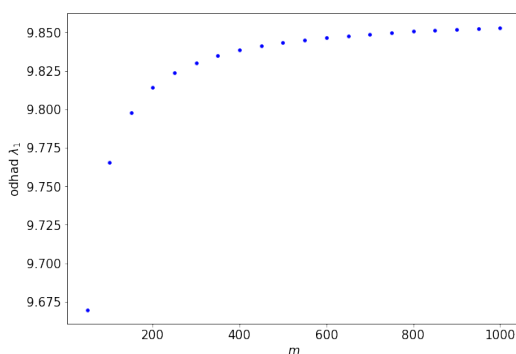
(c) Úloha (3.6)
($n = 40, m = 300, \delta = 10^{-5}, \sigma_0 = -1$)



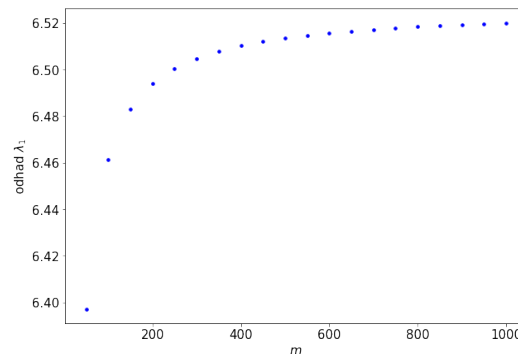
(d) Úloha (3.6)
($n = 300, m = 4 \times 10^4, \delta = 10^{-9}, \sigma_0 = -1$)

Obrázek 3.8: Intervalový průběh funkce P_v pro různé parametry.

Závislost dolního odhadu λ_1 na počtu intervalů m je na obr. 3.9. S jemnějším dělením se dolní odhad opět zpřesňuje.



(a) Úloha 3.5 ($n = 80, \delta = 10^{-5}, \sigma_0 = -1$)



(b) Úloha 3.6 ($n = 80, \delta = 10^{-5}, \sigma_0 = -1$)

Obrázek 3.9: Závislost dolního odhadu λ_1 na hodnotě m .

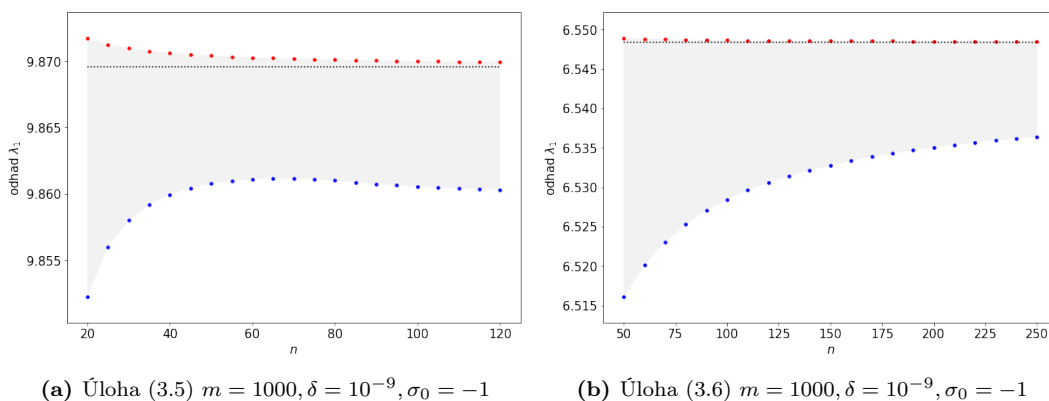
(c) Přes všechny intervaly $\mathcal{I}_i, i \in \{1, \dots, n \times m\}$ najdeme minimální dolní mez $\inf_{(0,1)} P_v \leq \inf_{(0,1)} P_v \leq \lambda_1$. A tu označíme za zaručený dolní odhad λ_1 .

3.6 | Výsledky

Z metody popsané v kapitole 3.5 je zřejmé, že výpočet horního odhadu λ_1 můžeme dále zpřesňovat, pokud budeme volit větší počet oblouků kubického splinu a zároveň dělit jednotlivé oblouky na více intervalů. Výsledná testovací funkce v se tím nijak nepokazí. Přesnost horního odhadu je tak závislá na složitosti použitých algoritmů výpočtu a výkonu použitého hardwaru, tedy na celkovém čase, který dokážeme výpočtu „obětovat“.

Při výpočtu dolního odhadu je třeba pečlivě volit optimální hodnotu počtu oblouků n v závislosti na hodnotě posunutí δ a nastavit dostatečně malé σ_0, σ_n . U ostatních parametrů je situace stejná jako u odhadu horního.

Na obr. 3.10 jsou oba odhady λ_1 v jednom grafu v závislosti na počtu oblouků splinu. Černá tečkovaná čára v levém grafu vyznačuje skutečnou hodnotu λ_1 v případě úlohy (3.5). Na grafu vpravo je pro úlohu (3.6) černě vyznačena pro porovnání přibližná hodnota λ_1 , získaná numericky. Z grafů je zřejmé, že vinou korekcí testovací funkce v a obecně složitějších výpočtů, nedosahuje dolní odhad takové přesnosti jako odhad horní.



Obrázek 3.10: Oba odhady v závislosti na n počtu oblouků splinu.

Nejpřesnější odhady λ_1 , které se nám pomocí popsané metody podařilo získat, jsou shrnuty v tab. 3.3 a 3.4. V úloze (3.5) můžeme porovnat vypočtenou hodnotu s analytickým řešením $\lambda_1 = \pi^2$. Pro úlohu (3.6) neumíme analytické řešení získat, a proto porovnáme oba odhady s hodnotou $\lambda_1 \approx 6,548395305831178$ vypočtenou numericky. Snad jen již pro ilustraci, výpočet jednoho odhadu, dolního nebo horního, na použitém hardwaru trvá přibližně jednu hodinu.

Tabulka 3.3: Výsledky výpočtů pro úlohu (3.5).

parametry	horní odhad λ_1	chyba vzhledem k přesnému řešení π^2
$n = 100, m = 10^5$	9,869609310406974	$4,9093176155423635 \times 10^{-6}$
parametry	dolní odhad λ_1	chyba vzhledem k přesnému řešení π^2
$n = 100, m = 10^5,$ $\sigma = -1, \delta = 10^{-9}$	9,869115030489464	$-4,893705998938458 \times 10^{-4}$

Tabulka 3.4: Výsledky výpočtů pro úlohu (3.6).

parametry	horní odhad λ_1	chyba vzhledem k numerickému odhadu
$n = 100, m = 10^5$	6,548398264577634	$2,9587464558389343 \times 10^{-6}$
parametry	dolní odhad λ_1	chyba vzhledem k numerickému odhadu
$n = 300,$ $m = 4 \times 10^4,$ $\sigma = -1, \delta = 10^{-9}$	6,543688536787878	$-4,706769043300696 \times 10^{-3}$

Všechny výpočty uvedené v této práci, včetně vygenerovaných grafů, jsou obsaženy v elektronické příloze a to jak ve formě spustitelných Jupyter notebooků, tak i jako PDF soubory.

4 | Závěr

Popsali jsme metodu, díky které lze pomocí Rayleighova podílu a Piconeho identity získat zaručené horní a dolní odhady prvního vlastního čísla λ_1 okrajových úloh s kladnou váhovou funkcí. Volba testovací funkce v ve formě kubického splinu a následný výpočet v intervalové aritmetice se ukázaly jako vhodné. Výpočet horního odhadu λ_1 je velmi přímočarý a výsledek se blíží skutečné hodnotě λ_1 jen s velmi malou chybou. Pro dolní odhad bylo třeba testovací funkci v pomocí dodatečných parametrů korigovat, abychom mohli použít intervalovou aritmetiku a zároveň dosáhnout co nejpřesnějších výsledků. Dolní odhad je kvůli tomu vždy zatížen větší chybou než odhad horní.

Dalšího zpřesnění výsledků výpočtu by bylo možné dosáhnout optimalizací použitých algoritmů. Zrychlení výpočtu, ať již použitím jiného programovacího jazyka nebo softwaru nebo například propojením Python knihovny na intervalovou aritmetiku `mpmath` s knihovnou `Numpy`, by dovolilo volit příhodnější parametry výpočtu a dále zpřesňovat výslednou hodnotu odhadu.

Zajímavým budoucím rozšířením by bylo upravení použité metody na okrajové úlohy ve více dimenzích, případně i úlohy s p -Laplaciánem (viz [1]). Metoda výpočtu horního a dolního odhadu prvního vlastního čísla bude velmi podobná i vzhledem k tomu, že vztahy pro odhady λ_1 v těchto úlohách používají stejné principy.

Literatura

- [1] Walter Allegretto, Yin Xi Huang: *A Picone's Identity for the p -Laplacian and Application*, *Nonlinear Analysis, Theory, Methods & Applications*, Vol 32., No. 7, pp. 819-830, 1998.
- [2] Petr Girg: *Rigorózní výpočty s nepřesnými čísly a matematické důkazy*, Matematika pro inženýry 21. století, Plzeň.
- [3] Pavel Drábek, Alois Kufner: *Úvod do funkcionální analýzy*, Západočeská univerzita, Fakulta aplikovaných věd, Katedra matematiky, Plzeň, 1993.
- [4] Bohumír Bastl: *texty k přednášce předmětu KMA/GM1*, Západočeská univerzita, Fakulta aplikovaných věd, Katedra matematiky, Plzeň, 2018.
- [5] C. Führer: *text k přednášce FMN081-Numerical Methods in Mechanics*, Lunds Universitet, Lund, Švédsko, 2006, <http://www.maths.lth.se/na/courses/FMN081/FMN081-06/lecture11.pdf>.
- [6] Radek Cibulka: *text k přednášce předmětu KMA/ODR*, Západočeská univerzita, Fakulta aplikovaných věd, Katedra matematiky, Plzeň, 2018.
- [7] S.M.Rump : *Algorithms for verified inclusions: Theory and practice*, In R.E.Moore, editor, *Reliability in Computing: The role of Interval methods in Scientific Computing*, chapter 1, Computer Arithmetic and Mathematical Software, pages 109-126, Academic Press, Boston, MA, 1988.
- [8] Loh, Eugene & Walster, G.: *Rump's Example Revisited* *Reliable Computing* 8, pages 245-248, 2002.
- [9] Fredrik Johansson and others: *mpmath: a Python library for arbitrary-precision floating-point arithmetic* (version 1.1.0), December 2018. <http://mpmath.org/>.

- [10] Meurer, Aaron and Smith, Christopher P. and Paprocki, Mateusz and Čertík, Ondřej and Kirpichev, Sergey B. and Rocklin, Matthew and Kumar, AMiT and Ivanov, Sergiu and Moore, Jason K. and Singh, Sartaj and Rathnayake, Thilina and Vig, Sean and Granger, Brian E. and Muller, Richard P. and Bonazzi, Francesco and Gupta, Harsh and Vats, Shivam and Johansson, Fredrik and Pedregosa, Fabian and Curry, Matthew J. and Terrel, Andy R. and Roučka, Štěpán and Saboo, Ashutosh and Fernando, Isuru and Kulal, Sumith and Cimrman, Robert and Scopatz, Anthony: *SymPy: symbolic computing in Python* <https://doi.org/10.7717/peerj-cs.103>.
- [11] Jones E, Oliphant E, Peterson P, et al.: *SciPy: Open Source Scientific Tools for Python* 2001-, <http://www.scipy.org/>.Online; accessed 2018-08-10.
- [12] Travis E, Oliphant.: *A guide to NumPy*, USA: Trelgol Publishing, (2006).