

**MASARYKOVA UNIVERZITA
PŘÍRODOVĚDECKÁ FAKULTA
ÚSTAV MATEMATIKY A STATISTIKY**

Bakalářská práce

BRNO 2017

LENKA HELDOVÁ



**MASARYKOVA UNIVERZITA
PŘÍRODOVĚDECKÁ FAKULTA
ÚSTAV MATEMATIKY A STATISTIKY**



Analýza a návrh algoritmů pro optimalizaci vizualizace dat

Bakalářská práce

Lenka Heldová

Vedoucí práce: Mgr. Jiří Zelinka, Dr.

Brno 2017

Bibliografický záznam

Autor: Lenka Heldová
Přírodovědecká fakulta, Masarykova univerzita
Ústav matematiky a statistiky

Název práce: Analýza a návrh algoritmů pro optimalizaci vizualizace dat

Studijní program: Matematika

Studijní obor: Modelování a výpočty

Vedoucí práce: Mgr. Jiří Zelinka, Dr.

Akademický rok: 2016/2017

Počet stran: ?? + ??

Klíčová slova: Klíčové slovo;

Bibliografický záznam

Autor:

Lenka Heldová
Prírodovedecká fakulta, Masarykova univerzita
Ústav matematiky a štatistiky

Názov práce: Analýza a návrh algoritmov pre optimalizáciu vizualizácie dát

Študijný program: Matematika

Študijný odbor: Modelovanie a výpočty

Vedúci práce: Mgr. Jiří Zelinka, Dr.

Akademický rok: 2016/2017

Počet strán: ?? + ??

Kľúčové slová: bbb

Bibliographic Entry

Author: Lenka Heldová
Faculty of Science, Masaryk University
Department of Mathematics and Statistics

Title of Thesis: Analysis and design of algorithms to optimize data visualization

Degree Programme: Mathematics

Field of Study: Modelling and Calculations

Supervisor: Mgr. Jiří Zelinka, Dr.

Academic Year: 2016/2017

Number of Pages: ?? + ??

Keywords:

Abstrakt

V této bakalářské/diplomové/rigorózní práci se věnujeme ...

Abstrakt

...

Abstract

In this thesis we study ...

Místo tohoto listu vložte kopii oficiálního (podepsaného) zadání práce.

Pod'akovanie

Na tomto místě bych chtěl(-a) poděkovat ...

Prohlášení

Prohlašuji, že jsem svoji bakalářskou práci vypracovala samostatně s využitím informačních zdrojů, které jsou v práci citovány.

Brno xx. měsíce 20xx

.....

Lenka Heldová

Obsah

Úvod	ix
Kapitola 1. PerfCake	1
1.1 O nástroji	1
1.2 Použitie	3
1.3 Výstup	5
1.3.1 Dáta	7
1.3.2 Grafy	7
Kapitola 2. Matematická teória	8
2.1 Regresná analýza	8
2.2 Jadrové odhady	9
2.2.1 Jadrová funkcia	9
2.2.2 Šírka vyhladzovacieho okna	9
2.2.3 Vhodný výber vyhladzovacieho parametra	10
2.2.4 Typy jadrových odhadov	12
2.3 Miera polohy a miera variability	13
2.3.1 Aritmetický priemer	13
2.3.2 Rozptyl a smerodajná odchýlka	13
Kapitola 3. Analýza dát	15
Kapitola 4. Vyhľadenie dát	17
Kapitola 5. Redukcia dát	23
5.1 Redukcia pomocou smerodajnej odchýlky a priemeru	23
5.2 Redukcia podobná redukcii pri zrýchľovaní zvuku	29
5.3 Redukcia pomocou jadrových odhadov	30
Závěr	35
Seznam použité literatúry	38

Úvod

Test interpunkce:

Žlutoučký kůň úpěl dábelské ódy
ěščřžýáíél'tňd'úú

Toto je nějaký úvodní text, ve kterém se obvykle popisuje struktura práce, cíle a případně i výsledky. Toto je nějaký úvodní text, ve kterém se obvykle popisuje struktura práce, cíle a případně i výsledky. Toto je nějaký úvodní text, ve kterém se obvykle popisuje struktura práce, cíle a případně i výsledky. Toto je nějaký úvodní text, ve kterém se obvykle popisuje struktura práce, cíle a případně i výsledky. Toto je nějaký úvodní text, ve kterém se obvykle popisuje struktura práce, cíle a případně i výsledky. Toto je nějaký úvodní text, ve kterém se obvykle popisuje struktura práce, cíle a případně i výsledky.

Toto je nějaký úvodní text, ve kterém se obvykle popisuje struktura práce, cíle a případně i výsledky. Toto je nějaký úvodní text, ve kterém se obvykle popisuje struktura práce, cíle a případně i výsledky. Toto je nějaký úvodní text, ve kterém se obvykle popisuje struktura práce, cíle a případně i výsledky. Toto je nějaký úvodní text, ve kterém se obvykle popisuje struktura práce, cíle a případně i výsledky. Toto je nějaký úvodní text, ve kterém se obvykle popisuje struktura práce, cíle a případně i výsledky. Toto je nějaký úvodní text, ve kterém se obvykle popisuje struktura práce, cíle a případně i výsledky.

Quisque facilisis auctor sapien. Pellentesque gravida hendrerit lectus. Mauris rutrum sodales sapien. Fusce hendrerit sem vel lorem. Integer pellentesque massa vel augue. Integer elit tortor, feugiat quis, sagittis et, ornare non, lacus. Vestibulum posuere pellentesque eros. Quisque venenatis ipsum dictum nulla. Aliquam quis quam non metus eleifend interdum. Nam eget sapien ac mauris malesuada adipiscing. Etiam eleifend neque sed quam. Nulla facilisi. Proin a ligula. Sed id dui eu nibh egestas tincidunt. Suspendisse arcu.

Maecenas dui. Aliquam volutpat auctor lorem. Cras placerat est vitae lectus. Curabitur massa lectus, rutrum euismod, dignissim ut, dapibus a, odio. Ut eros erat, vulputate ut, interdum non, porta eu, erat. Cras fermentum, felis in porta congue, velit leo facilisis odio, vitae consectetur lorem quam vitae orci. Sed ultrices, pede eu placerat auctor, ante ligula rutrum tellus, vel posuere nibh lacus nec nibh. Maecenas laoreet dolor at enim. Donec molestie dolor nec metus. Vestibulum libero. Sed quis erat. Sed tristique. Duis pede leo, fermentum quis, consectetur eget, vulputate sit amet, erat.

Donec vitae velit. Suspendisse porta fermentum mauris. Ut vel nunc non mauris pharetra varius. Duis consequat libero quis urna. Maecenas at ante. Vivamus varius, wisi sed egestas tristique, odio wisi luctus nulla, lobortis dictum dolor ligula in lacus. Vivamus aliquam, urna sed interdum porttitor, metus orci interdum odio, sit amet euismod lectus

felis et leo. Praesent ac wisi. Nam suscipit vestibulum sem. Praesent eu ipsum vitae pede cursus venenatis. Duis sed odio. Vestibulum eleifend. Nulla ut massa. Proin rutrum mattis sapien. Curabitur dictum gravida ante.

Phasellus placerat vulputate quam. Maecenas at tellus. Pellentesque neque diam, dignissim ac, venenatis vitae, consequat ut, lacus. Nam nibh. Vestibulum fringilla arcu mollis arcu. Sed et turpis. Donec sem tellus, volutpat et, varius eu, commodo sed, lectus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque enim arcu, suscipit nec, tempus at, imperdiet vel, metus. Morbi volutpat purus at erat. Donec dignissim, sem id semper tempus, nibh massa eleifend turpis, sed pellentesque wisi purus sed libero. Nullam lobortis tortor vel risus. Pellentesque consequat nulla eu tellus. Donec velit. Aliquam fermentum, wisi ac rhoncus iaculis, tellus nunc malesuada orci, quis volutpat dui magna id mi. Nunc vel ante. Duis vitae lacus. Cras nec ipsum.

Morbi nunc. Aliquam consectetur varius nulla. Phasellus eros. Cras dapibus porttitor risus. Maecenas ultrices mi sed diam. Praesent gravida velit at elit vehicula porttitor. Phasellus nisl mi, sagittis ac, pulvinar id, gravida sit amet, erat. Vestibulum est. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur id sem elementum leo rutrum hendrerit. Ut at mi. Donec tincidunt faucibus massa. Sed turpis quam, sollicitudin a, hendrerit eget, pretium ut, nisl. Duis hendrerit ligula. Nunc pulvinar congue urna.

Nunc velit. Nullam elit sapien, eleifend eu, commodo nec, semper sit amet, elit. Nulla lectus risus, condimentum ut, laoreet eget, viverra nec, odio. Proin lobortis. Curabitur dictum arcu vel wisi. Cras id nulla venenatis tortor congue ultrices. Pellentesque eget pede. Sed eleifend sagittis elit. Nam sed tellus sit amet lectus ullamcorper tristique. Mauris enim sem, tristique eu, accumsan at, scelerisque vulputate, neque. Quisque lacus. Donec et ipsum sit amet elit nonummy aliquet. Sed viverra nisl at sem. Nam diam. Mauris ut dolor. Curabitur ornare tortor cursus velit.

Morbi tincidunt posuere arcu. Cras venenatis est vitae dolor. Vivamus scelerisque semper mi. Donec ipsum arcu, consequat scelerisque, viverra id, dictum at, metus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut pede sem, tempus ut, porttitor bibendum, molestie eu, elit. Suspendisse potenti. Sed id lectus sit amet purus faucibus vehicula. Praesent sed sem non dui pharetra interdum. Nam viverra ultrices magna.

Aenean laoreet aliquam orci. Nunc interdum elementum urna. Quisque erat. Nullam tempor neque. Maecenas velit nibh, scelerisque a, consequat ut, viverra in, enim. Duis magna. Donec odio neque, tristique et, tincidunt eu, rhoncus ac, nunc. Mauris malesuada malesuada elit. Etiam lacus mauris, pretium vel, blandit in, ultricies id, libero. Phasellus bibendum erat ut diam. In congue imperdiet lectus.

Aenean scelerisque. Fusce pretium porttitor lorem. In hac habitasse platea dictumst. Nulla sit amet nisl at sapien egestas pretium. Nunc non tellus. Vivamus aliquet. Nam adipiscing euismod dolor. Aliquam erat volutpat. Nulla ut ipsum. Quisque tincidunt auctor augue. Nunc imperdiet ipsum eget elit. Aliquam quam leo, consectetur non, ornare sit amet, tristique quis, felis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque interdum quam sit amet mi. Pellentesque mauris dui, dictum a, adipiscing ac, fermentum sit amet, lorem.

Ut quis wisi. Praesent quis massa. Vivamus egestas risus eget lacus. Nunc tincidunt, risus quis bibendum facilisis, lorem purus rutrum neque, nec porta tortor urna quis orci. Aenean aliquet, libero semper volutpat luctus, pede erat lacinia augue, quis rutrum sem

ipsum sit amet pede. Vestibulum aliquet, nibh sed iaculis sagittis, odio dolor blandit augue, eget mollis urna tellus id tellus. Aenean aliquet aliquam nunc. Nulla ultricies justo eget orci. Phasellus tristique fermentum leo. Sed massa metus, sagittis ut, semper ut, pharetra vel, erat. Aliquam quam turpis, egestas vel, elementum in, egestas sit amet, lorem. Duis convallis, wisi sit amet mollis molestie, libero mauris porta dui, vitae aliquam arcu turpis ac sem. Aliquam aliquet dapibus metus.

Vivamus commodo eros eleifend dui. Vestibulum in leo eu erat tristique mattis. Cras at elit. Cras pellentesque. Nullam id lacus sit amet libero aliquet hendrerit. Proin placerat, mi non elementum laoreet, eros elit tincidunt magna, a rhoncus sem arcu id odio. Nulla eget leo a leo egestas facilisis. Curabitur quis velit. Phasellus aliquam, tortor nec ornare rhoncus, purus urna posuere velit, et commodo risus tellus quis tellus. Vivamus leo turpis, tempus sit amet, tristique vitae, laoreet quis, odio. Proin scelerisque bibendum ipsum. Etiam nisl. Praesent vel dolor. Pellentesque vel magna. Curabitur urna. Vivamus congue urna in velit. Etiam ullamcorper elementum dui. Praesent non urna. Sed placerat quam non mi. Pellentesque diam magna, ultricies eget, ultrices placerat, adipiscing rutrum, sem.

Kapitola 1

PerfCake

Táto kapitola sa venuje nástroju na testovanie výkonu PerfCake, jeho použitiu a typu dát, ktorými sa budeme zaoberať. Celá kapitola je len povrchným zoznámením sa s nástrojom pre potreby tejto práce, čerpá hlavne z oficiálne stránky tohto nástroja, kde v prípade väčšieho záujmu je všetko o nástroji podrobne rozpisane.



Obr. 1.1: PerfCake

1.1 O nástroji

PerfCake je open-source framework vyvájaný spoločnosťou Red Hat, ktorý je primárne používaný na testovanie výkonu. Je to generátor zaťaženia s cieľom byť minimalistický a ľahko ovládateľný, poskytujúci stabilné výsledky, majúci minimálny vplyv na testovaný systém, byť nezávislý na platforme, umožňujúci vysokú priepustnosť a používajúci komponentový dizajn.

Nástroj PerfCake môže byť použitý na všetky druhy testovania výkonu, včetne:

- *záťažové testovanie (load testing)* - testuje sa správanie softvéru pri očakávanom zaťažení, pri tomto type testu sa prejavujú takzvané “najužšie miesto”(bottlenecks) v systéme, miesta, ktoré v nejakom smere sú systém limitujú,
- *testovanie hraničnej zátlače (stress testing)* - tento test hľadá hranice možného zaťaženia systému a zistuje správanie sa systému za touto hranicou,

- *testovanie odolnosti (endurance testing, soak testing)* - zistuje sa, či testovaný systém dokáže vydržať sústavnú záťaž po dlhšiu dobu a taktiež, či pri takejto sústavnej záťaži neklesá jeho výkon,
- *testovanie škálovateľnosti (scalability testing)* - testuje sa, či systém dobre škáluje a teda či je možné efektívne zvyšovať jeho kapacity bez nečakaných zmien správania,
- *benchmarking* - zaznamenávanie a porovnávanie výsledkov testov, za účelom získania rôznych metrík.

Pre účely spomínaných testov, podporuje PerfCake radu metrík na testovanie cieľového systému. Napríklad:

- monitorovanie vybraného atribútu správy, roztriedenie a spočítanie podľa konkrétnych hodnôt daného atribútu,
- počítanie priemernej priepustnosti za sekundu,
- sledovanie veľkosti spotrebovanej pamäte cieľovej JVM (Java virtual machine),
- zaznamenávanie času odpovede (response time) v milisekundách, pomocou HDR (High Dynamic Range) histogramu, ktorý dokáže riešiť problém Koordinovaného vynechania (Coordinated omission), viac o tomto probléme a riešení pomocou HRD histogramu sa môžeme dočítať na oficiálnej stránke nástroja v sekcii 4.7. Reporting [],
- počítanie štatistik, ako maximum, minimum, priemer z hodnôt priepustnosti alebo časov odpovedí, za čas od začiatku trvania testu, až po reportovanie výsledku alebo zo špecifikovaného časového okna,
- detekovanie pripravenosti systému, ktorá sa určuje pomocou splnenia 3 podmienok: momentálna priepustnosť sa mení o menej ako nastaviteľná hranica, špecifikovaný čas bol dosiahnutý a špecifikovaný počet iterácií bol vykonaný,
- rozoznávanie geo-lokácie tretích strán.

Užívateľ má niekoľko možností reportovania výsledkov, ako napríklad výpis na konzolu, súbor logov, CSV formát, ktorý môže byť importovaný ako tabuľkový zošit (*spreadsheet*) pomocou nástrojov ako Excel alebo LibreOffice Calc, alebo tiež dokáže generovať grafy výsledkov už v priebehu testovania.

Jeden z problémov podobných testovacích nástrojov je prípad, kedy testovaný softvér zlyhá a vzápätí odošle chybovú správu. Pre tento prípad sú v nástroji PerfCake zabudované validátory, ktoré dokážu takúto správu rozpoznať a nezaviesť tým chybové správy do výsledku výkonu softvéru. Podobné softvéry, ktoré takéto validácie neprevádzajú potom vytvárajú veľmi skreslený výsledok.

1.2 Použitie

Pre spustenie nástroja PerfCake je potrebné vytvoriť, takzvané scenáre (scenarios). V týchto súboroch sú definované testovacie bloky, ktoré hovoria PerfCaku čo a ako robiť. Scenáre sú rozdelené do niekolkých sekcií:

- *vlastnosti (properties)* - táto sekcia slúži na pridanie akýchsi meta-dát o scenárii, nie je to povinný parameter a slúži na komplexnejšie využitie testovacích scenárov,
- *beh (run)* - slúži na definovanie dĺžky trvania testu,
- *generátor (generator)* - špecifikuje ako a kolko zaťaženia sa má vytvárať,

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <scenario
3      xmlns="urn:perfcake:scenario:8.0"
4      <!-- Scenario properties (optional) -->
5      <properties>
6          <property name="..." value="..."/>
7          ...
8      </properties>
9      <!-- Run section (required) -->
10     <run ... >
11         ...
12     </run>
13     <!-- Generator section (required) -->
14     <generator ... >
15         ...
16     </generator>
17     <!-- Sequences section (optional) -->
18     <sequences>
19         ...
20     </sequences>
21     <!-- Sender section (required) -->
22     <sender ... >
23         ...
24     </sender>
25     <!-- Receiver section (optional) -->
26     <receiver>
27         ...
28     </receiver>
29     <!-- Reporting section (optional) -->
30     <reporting>
31         ...
32     </reporting>
33     <!-- Messages section (optional) -->
34     <messages>
35         ...
36     </messages>
37     <!-- Validation section (optional) -->
38     <validation>
39         ...
40     </validation>
41 </scenario>
```

Obr. 1.2: Štruktúra scenára

- *odosielateľ'(sender)* - táto sekcia je o spôsobe prenosu, o tom, kde zaťaženie posielat', poprípade o prijímaní odozvy z testovaného systému,

- *prijímateľ'(receiver)* - ďalšia rozširujúca sekcia, ktorá sa používa v prípade, že odozva z testovaného systému prichádza iným kanálom (protokolom) ako bola poslaná správa,
- *reportovanie (reporting)* - definuje metriky, spôsob a výstup reportovania výsledkov testu,
- *správy (messages)* - sekcia slúži na definovanie formátu a obsahu správ posielaných na testovaný systém,
- *sekvencie (sequencies)* - táto sekcia umožňuje používať sekvenciu hodnôt, ktoré sa menia pre každú správu (prípadne množinu správ, špecifikovaných v jednom scenárii), čo dáva možnosť vytvárať unikátné správy,
- *validácia (validation)* - dovoľuje validovať odpoveď testovaného systému.

Štruktúru testovacieho scenára vo formáte XML (eXtensible Markup Language) a povinnosť konkrétnych parametrov zobrazuje obrázok 1.2. Okrem tohto formátu podporuje PerfCake aj iné možnosti scenárov, napríklad DSL (Domain Specific Language) formát alebo použitie špeciálneho rozhrania pre programovanie aplikácií (API) v jazyku Java, kde si užívateľ môže nadefinovať celý scenár.

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <scenario
3      xmlns="urn:perfcake:scenario:7.0">
4      <run type="${perfcake.run.type:time}"
5          value="${perfcake.run.duration:30000}" />
6      <generator class="DefaultMessageGenerator"
7          threads="${perfcake.thread.count:100}" />
8      <sender class="HttpSender">
9          <target>http://${server.host}/post</target>
10         <property name="method" value="POST"/>
11     </sender>
12     <reporting>
13         <reporter class="IterationsPerSecondReporter">
14             <destination class="ConsoleDestination">
15                 <period type="time" value="1000" />
16             </destination>
17         </reporter>
18     </reporting>
19     <messages>
20         <message uri="plain.txt" />
21     </messages>
22 </scenario>
```

Obr. 1.3: Príklad scenára

Pre jednoduchú ukážku si uvedieme príklad scenára XML. Máme vyvinutý softvér, ktorý asynchrónne spracováva správy prichádzajúce pomocou POST metódy a HTTP protokolu. Následne tieto správy spracúva (validuje, vytvára záznamy do databázy, generuje

reporty, ...) a po úspešnom spracovaní pošle potvrdzujúcu správu späť. Na obrázku 1.3 sa nachádza scenár pre jednoduchý testovací blok, ktorý by sme mohli na takýto nami vytvorený softvér použiť. Zo scenára je možné vidieť, že tento test bude bežať 3000 ms = 3 sekundy, generuje správy použitím 100 vlákien, ktoré posiela pomocou HTTP protokolu a metódy POST. Text správy je špecifikovaný v súbore "plain.txt" a výsledky sú zaznamenávané na konzolu, každú 1 sekundu. Ako reporter je použitý "IterationsPerSecondReporter" a teda sa každú sekundu zapíše, koľko správ bolo softvérom spracovaných.

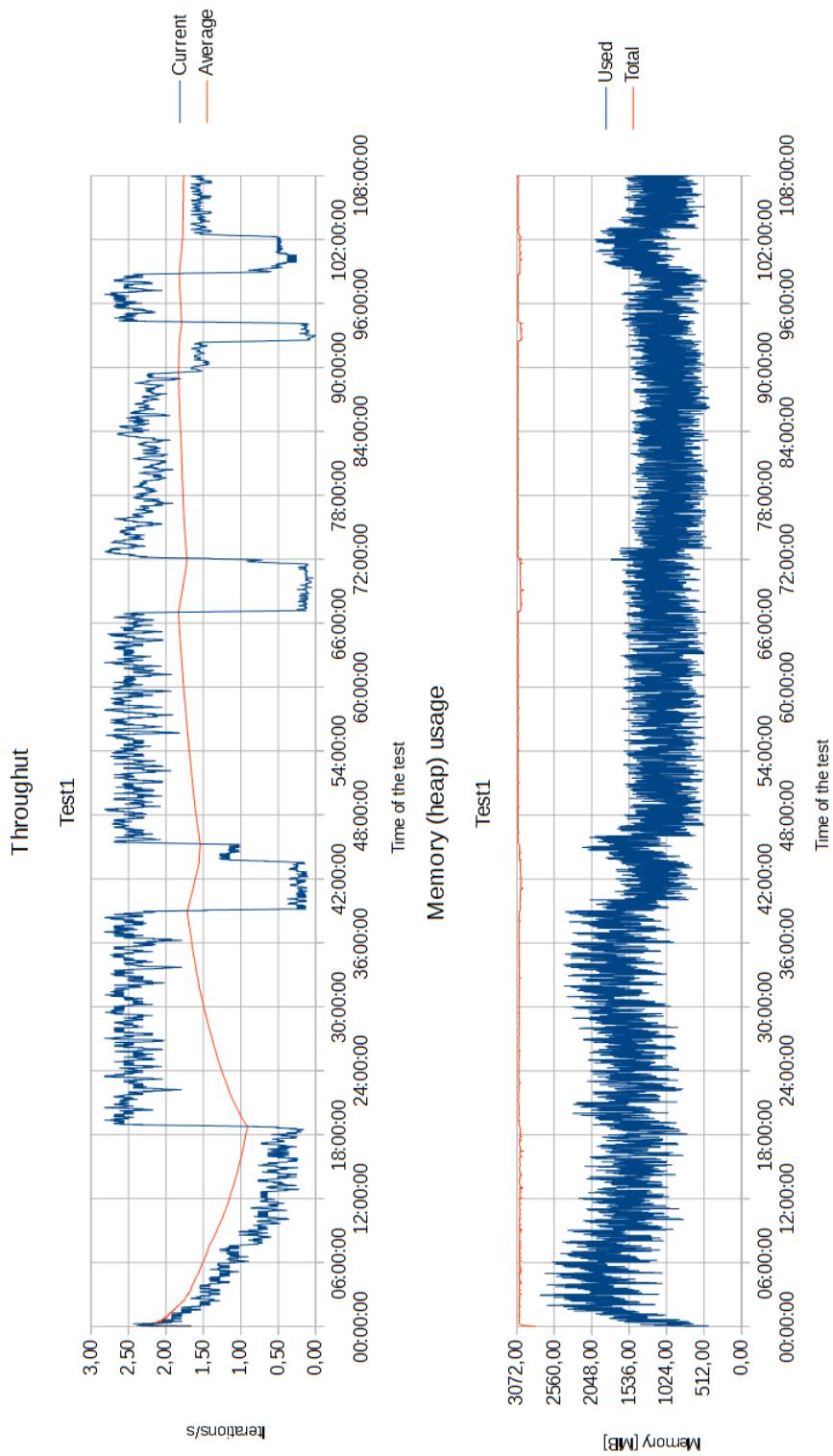
Veľmi podobným scenárom by sme mohli testovať, koľko softvér používa pamäte, stačilo by zmeniť atribút reporter na "MemoryUsageReporter". Týmto spôsobom sa veľmi ľahko odhalujú úniky pamäte. Scenáre sa púšťajú pomocou nástroja Maven alebo pomocou shell skriptu.

1.3 Výstup

Ako bolo uvedené v prvej podkapitole, PerfCake má bohaté možnosti východzích súborov. Pri vytváraní optimalizačného algoritmu budeme ale pracovať iba s dvoma typmi. Dáta budeme spracovať z CSV súboru a následne z nich budeme generovať grafy.

Throughput			Memory		
Time	Current	Average	Time	Used	Total
00:01:00	1,67	1,67	00:01:00	777,30	2817,06
00:02:00	1,67	1,67	00:02:00	528,81	2835,94
00:03:00	2,22	2,22	00:03:00	450,90	2901,19
00:04:00	2,08	2,08	00:04:00	585,94	2911,63
00:05:00	2,33	2,33	00:05:00	979,96	2953,00
00:06:00	2,22	2,22	00:06:00	855,63	2960,25
00:07:00	2,38	2,38	00:07:00	1193,20	2992,13
00:08:00	2,29	2,29	00:08:00	1175,80	3001,75
00:09:00	2,22	2,22	00:09:00	1197,66	3009,88
00:10:00	2,17	2,17	00:10:00	831,64	3013,38
00:11:00	2,27	2,27	00:11:00	1380,88	3019,63
00:12:00	2,22	2,22	00:12:00	815,44	3031,56
00:13:00	2,18	2,18	00:13:00	950,54	3035,38
00:14:00	2,26	2,26	00:14:00	726,44	3036,44
00:15:00	2,44	2,33	00:15:00	659,87	3042,88
00:16:00	2,18	2,19	00:16:00	715,86	3042,88
00:17:00	2,18	2,16	00:17:00	909,47	3042,44
00:18:00	2,05	2,13	00:18:00	1016,83	3042,31
00:19:00	2,14	2,19	00:19:00	697,72	3043,13
00:20:00	2,05	2,17	00:20:00	844,43	3043,75

Tabuľka 1.1: Dáta



Obr. 1.4: Grafy

1.3.1 Dáta

Pri optimalizácii dát pre potreby vykreslovania do grafov, bude používaný formát súboru CSV, ktorý môžeme previesť do podoby tabuľky.

Tabuľka 1.1 ukazuje dátá vygenerované nástrojom PerfCake za 20 minút, testovala sa prieplustnosť softvéru v iteráciách za sekundu a využitie pamäte v megabajtoch. Pri vytváraní algoritmu budeme pracovať len s dátami, ktoré boli vytvorené pri testovaní týchto dvoch vlastností softvéru. Pri prieplustnosti sa počítava priemer za časovú jednotku (current) a tiež priemer za niekoľko časových jednotiek (average), časová jednotka a veľkosť okna, s ktorými sa bude počítať sa dá nastaviť v scenári, pri dátach s tabuľky je to minúta a 15 záznamov. Priemer sa začína počítať, keď sa naplní okno a potom už pre každý ďalší záznam.

Pri testovaní využívania pamäte sa zaznamenáva použitá (used) a celková pamäť (total). Celková pamäť, je pamäť, ktorú si JVM alokuje a použitá pamäť je pamäť, ktorú naozaj využije - celková mínus voľná pamäť. Viac o pamäti v JVM sa čitateľ dozvie v oficiálnej dokumentácii (<http://docs.oracle.com/javase/8/docs/api/java/lang/Runtime.html>).

V popisovanej tabuľke sú dátá zaznamenané za 20 min, v praxi ale tie testy bežia väčšinou niekoľko desiatok hodín a teda tabuľky môžu dosahovať tisíce záznamov. Ukladať a následne spracovať takéto množstvo dát je pamäťovo aj časovo náročné a preto budem hľadať algoritmus, ktorý zmenší objem dát a zároveň neznehodnotí informáciu z nich vyplývajúcemu, pri vykreslení do grafov.

1.3.2 Grafy

Pomocou grafu je často jednoduchšie pochopiť a spracovať veľké množstvo informácií, ako zo surových čísel v obrovskej tabuľke. Na obrázku 1.4 sú znázornené grafy z testov, ktoré bežali 108 hodín, a ktorých prvých 20 záznamov sme si ukázali v tabuľke 1.1. Pre analýzu výsledku nie sú potrebné tak presné grafy, väčšinou je dôležitá hodnota okolo ktorej dátá v určitej časti grafu konvergujú, alebo či hodnoty veľmi "skáču". V bakalárskej práci budú tieto grafy upravené a následne porovnávané s grafmi, ktoré budú vykreslené z dát, na ktoré sa použije optimalizačný algoritmus.

Kapitola 2

Matematická teória

V tejto kapitole uvedieme znalosti z matematickej teórie, ktoré sú potrebné pre zstrojenie optimalizačných algoritmov, ktorými sa budeme zaoberať v následujúcich kapitolách. Najprv sa oboznámime s úvodom do regresnej analýzy a s neparametrickou metódou odhadu regresnej funkcie - metódou jadrových odhadov, ktorú použijeme na vyhľadenie dát. Na konci kapitoly si ešte uvedieme vzorce pre aritmetický priemer, rozptyl a smerodajnú odchýlku, ktoré budú použité na redukciu dát. Pojmy a definície sú prebraté zo zdrojov...

2.1 Regresná analýza

Pre pevné hodnoty nezávisle premennej X (v našom prípade čas) máme k dispozícii nameenané hodnoty závisle premennej Y (priepustnosť, použitá pamäť,...). Takýmito dvojicami bodov $(x_i, Y_i), i = 1, \dots, n$ chceme preložiť vhodnú krivku, tak aby boli odfiltrované výkyvy a bolo možné lepšie poznať štruktúru dát. Táto krivka sa nazýva *regresná krivka* a jej príslušný regresný vzťah zapisujem v tvare

$$Y_i = m(x_i) + \varepsilon_i, i = 1, \dots, n, \quad (2.1)$$

kde m je neznáma regresná funkcia a $\varepsilon_i, i = 1, \dots, n$, sú chyby merania. Cieľom regresnej analýzy je nájsť vhodnú approximáciu \hat{m} neznámej funkcie m . Hľadanie tejto regresnej krivky sa tiež nazýva *vyhľadzovanie* a je možné použiť dva spôsoby odhadu, *parametricky* a *neparametricky*:

- *Parametrický prístup* - predpokladá, že regresná funkcia je nejakého predpísaného tvaru. Odhadnutá regresná funkcia bude teda určitého tvaru a bude ju popisovať množina parametrov - to je dôvod pre názov *parametricky*.
- *Neparametrický prístup* - nepredpokladá predpísany tvar regresnej funkcie. Tento prístup sa vyhýba parametrizácii a tvar funkcie sa odhaduje priamo z dát. Predpokladá sa jedine istá hladkosť hľadanej funkcie.

Jedna z najjednoduchších neparametrických metód je *metóda klízavých priemerov*. Pre odhad hodnoty Y_i sa používa priemer niekolkých hodnôt $Y_j, j \in [i-h, i+h]$ v centrovanom okolí príslušného bodu x_i .

Konkrétnie,

$$\hat{m}(x) = \frac{\sum_{i=1}^n Y_i I_{[x-h, x+h]}(x_i)}{\sum_{i=1}^n I_{[x-h, x+h]}(x_i)}. \quad (2.2)$$

Na vyjadrenie

$$I_{[j,k]}(x) \begin{cases} 1 & x \in [j, k], \\ 0 & \text{inak.} \end{cases} \quad (2.3)$$

sa používa *indikátorová funkcia* $I_{[j,k]}(x)$.

Jadrové odhady sa považujú za zovšeobecnenie *metódy klízavých priemerov*.

2.2 Jadrové odhady

Pri odhadovaní regresnej funkcie pomocou jadrových odhadov, sa používajú vážené hodnoty Y v centrovanom okolí príslušného bodu x_i . Váhy hodnôt Y sú závislé na vzdialosti príslušných x bodov od konkrétneho x_i , bližšie hodnoty - väčšia váha. Toto nám pomáha dosiahnuť *jadrová funkcia*. Vzorec pre jadrové odhady vo všeobecnosti, môžme zapísť nasledovne

$$\hat{m}(x, h) = \sum_{i=1}^n W_i(x, h) Y_i, \quad (2.4)$$

kde $W_i(x, h)$ je váhová funkcia s vyhľadzovacím parametrom h . Ide teda o akýsi vážený súčet pozorovaní, ktorý v jednom z najjednoduchších prípadov môže mať tvar

$$W_i(x, h) = K_h(x - x_i) = \frac{1}{h} K\left(\frac{x - x_i}{h}\right), \quad (2.5)$$

kde K je jadrová funkcia. Váhová funkcia by mala dávať súčet váh 1.

2.2.1 Jadrová funkcia

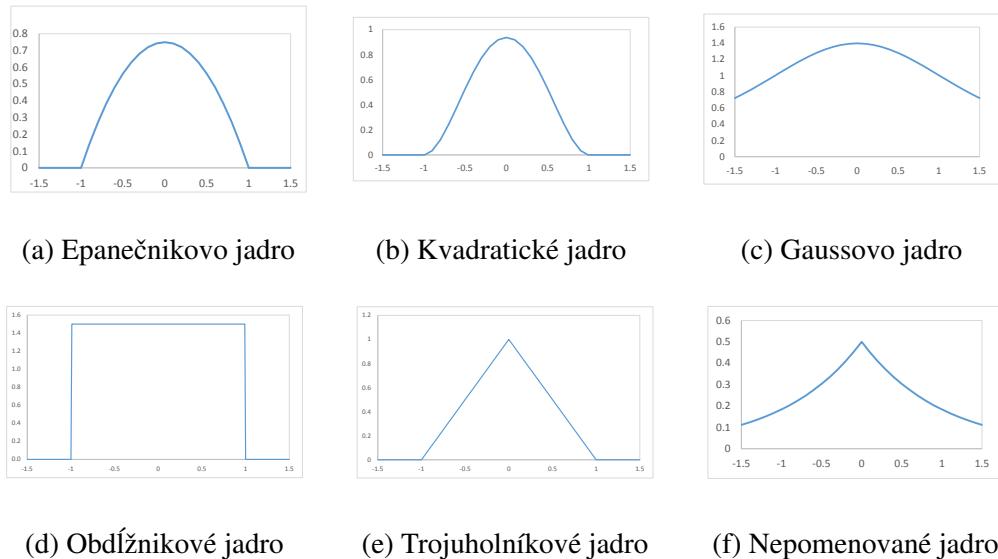
Jadrová funkcia determinuje tvar vyhľadzovacej funkcie. Na obrázku 2.1 môžeme vidieť niekoľko najpoužívanejších jadrových funkcií. V popise funkcie sa používa indikátorová funkcia (2.3).

Vo všeobecnosti hocijaká integrovateľná, obmedzená funkcia, ktorá spĺňa kritériá 2.6 môže byť jadrom.

$$\begin{aligned} a) \int K(z) dz &= 1 & b) \int z K(z) dz &= 0 \\ c) \int z^2 K(z) dz &< \infty & d) K(x) &\geq 0 \text{ pre všetky } x. \end{aligned} \quad (2.6)$$

2.2.2 Šírka vyhľadzovacieho okna

Okrem jadrovej funkcie, alebo jadra, je ďalším dôležitým parametrom tejto metódy šírka vyhľadzovacieho okna h . Šírka vyhľadzovacieho okna alebo aj vyhľadzovací parameter h udáva šírku vyhľadzovacej funkcie a teda aj silu vyhladenia.



Obr. 2.1: Rôzne tvary jadrových funkcií:

$$\begin{array}{lll}
 \text{a)} K(x) = \frac{3}{4}(1-x^2)I_{[-1,1]}(x), & \text{b)} K(x) = \frac{15}{16}(1-x^2)^2I_{[-1,1]}(x), & \text{c)} K(x) = \frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}, \\
 \text{d)} K(x) = \frac{1}{2}I_{[-1,1]}(x), & \text{e)} K(x) = (1-|x|)I_{[-1,1]}(x), & \text{f)} K(x) = \frac{1}{2}e^{-|x|}
 \end{array}$$

Malá šírka vyhľadzovacieho okna znamená, že odhad závisí na úzkom okolí bodu x_i a teda odhad do veľkej miery reprodukuje pôvodné dátá. Naopak, ak zvolíme vysokú hodnotu h , aj veľmi vzdialené hodnoty majú vysoký dopad na odhad, čo vedie k “prehladeniu” a pri dostatočne veľkej šírke h až k priemeru dát.

Spomínané rozdieli v šírke vyhľadzovacieho okna môžeme vidieť na obrázku 2.2. Pre ilustráciu vplyvu vyhľadzovacieho parametra sú použité reálne dátá z merania použitej pamäte nástroja PerfCake. Aplikované sú Nadaraya-Watsonove odhady, ktoré sú popísané v nasledujúcej podkapitole, s Epanečníkovým jadrovou funkciou.

2.2.3 Vhodný výber vyhľadzovacieho parametra

Hoci je v praxi bežné vybrať vhodnú šírku vyhľadzovacieho okna na základe niekoľkých pokusov a následného subjektívneho výberu, existuje niekoľko metód pre výber optimálnej šírky vyhľadzovacieho okna.

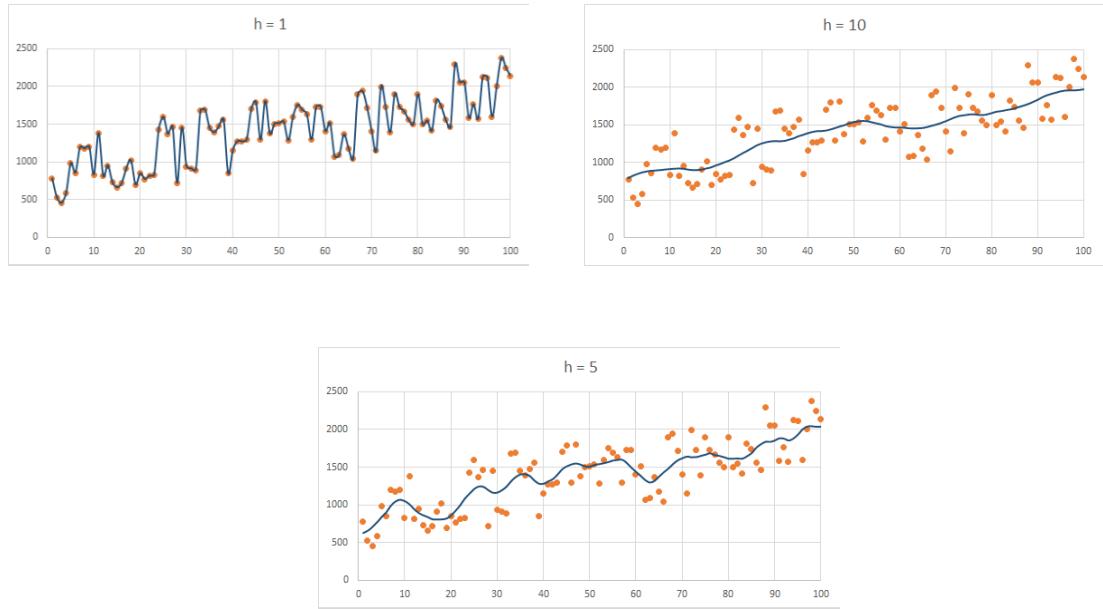
Kvalita jadrových odhadov regresnej funkcie môže byť lokálne popísaná pomocou *srednej kvadratickej chyby (MSE - Mean Square Error)*

$$MSE\{\hat{m}(x, h)\} = E\{\hat{m}(x, h) - m(x)\}^2.$$

Alebo tiež pomocou globálnej chyby - *priemernej strednej kvadratickej chyby (AMSE - average MSE)*

$$AMSE\{\hat{m}(., h)\} = \frac{1}{n}E \sum_{i=1}^n \{\hat{m}(x_i, h) - m(x_i)\}^2.$$

Minimalizovaním týchto chýb pre hodnotu parametra h , dostávame jeho optimum.



Obr. 2.2: Porovnanie šírky vyhľadzovacieho okna, bodky sú pôvodné dátá, súvislá čiara odhadnutá regresná funkcia

Kedže vzorec pre AMSE obsahuje neznáme hodnoty regresnej funkcie, použijeme pre výpočet optimálnej hodnoty parametra h , reziduálny súčet štvorcov (RSS), kde tieto neznáme hodnoty nahradíme nameranými hodnotami Y_i . Teda

$$RSS_n(h) = \frac{1}{n} \sum_{i=1}^n \{Y_i - \hat{m}(x_i, h)\}^2. \quad (2.7)$$

RSS ale nie je nestranným odhadom AMSE, preto sa metódy pre odhad optimálnej šírky vyhľadzovacieho okna snažia RSS upraviť tak, aby bol nestranný.

Jedna z najpoužívanejších metód tohto typu je *metóda krížového overenia* (*Cross-validation method*). Táto metóda spočíva vo vyniechaní pozorovania x_j pre spočítanie odhadu v bode x_j

$$\hat{m}_{-j}(x_j, h) = \sum_{\substack{i=1 \\ i \neq j}}^n W_i(x_j, h) Y_i.$$

S použitím tejto modifikácie, môžme RSS prepísat do tvaru

$$CV(h) = \frac{1}{n} \sum_{i=1}^n \{Y_i - m_{-i}(x_i)\}^2, \quad (2.8)$$

funkcia CV je tiež známa ako “cross-validačná” funkcia. Odhadnuté optimálne h dostaneme minimalizovaním (2.8)

$$\hat{h}_{CV} = \arg \min_{h \in H_n} CV(h). \quad (2.9)$$

Aj keď je táto metóda jedna z najpoužívanejších, negarantuje nestrannosť odhadu, ani že \hat{h}_{CV} minimalizuje AMSE (alebo hocijakú inú chybovú mieru). Dôsledkom toho je, že \hat{h}_{CV} nadobúda väčšinou jemne nižšiu hodnotu ako je optimálna šírka h .

Viac o metóde krížového overenia, ale aj iných metódach na odhad optimálnej šírky vyhľadzovacieho okna, napr. Malloowsova metóda, Penalizačná metóda, alebo metóda založená na Fourierovej transformácii, môžme nájsť v [(Kernel smoothing)].

2.2.4 Typy jadrových odhadov

V tejto podkapitole si uvedieme niektoré najznámejšie typy jadrových odhadov, ktoré sú uvedené v zdrojoch ...

Ak ako váhovú funkciu použijeme (2.5) dostávame **Priestley-Chaove odhady**

$$\hat{m}_{PCH}(x, h) = \frac{1}{n} \sum_{i=1}^n K_h(x - x_i) Y_i. \quad (2.10)$$

Modifikovaním tohto estimátora získame ďalší známy typ jadrových odhadov, **Gasser-Müllerove odhady**

$$\hat{m}_{GM}(x, h) = \sum_{i=1}^n Y_i \int_{s_{i-1}}^{s_i} K_h(t - x) dt, \quad (2.11)$$

kde $s_0 = 0, s_i = \frac{x_i + x_{i+1}}{2}, s_n = 1$, pre x_i ekvidistantne rozdelené na $[0,1]$. Tento estimátor sa dá vhodne použiť pre odhad derivácie regresnej funkcie.

Nasledujúce typy jadrových odhadov vychádzajú z takzvaných *lokálne polynomiálnych jadrových odhadov*. Odhad neznámej regresnej funkcie v bode x je získaný preložením polynómu stupňa d váženou metódou najmenších štvorcov. Nech tento polynom má tvar

$$P(u) = \beta_0 + \beta_1(u - x) + \dots + \beta_d(u - x)^d. \quad (2.12)$$

Váhy sú dané pomocou jadrovej funkcie

$$K_h(x - x_i) = \frac{1}{h} K\left(\frac{x - x_i}{h}\right). \quad (2.13)$$

Aplikujeme váženú metódu najmenších štvorcov, to znamená, že minimalizujeme výraz

$$\sum_{i=1}^n \{Y_i - \beta_0 - \beta_1(x_i - x) - \dots - \beta_d(x_i - x)^d\}^2 K_h(x_i - x) \quad (2.14)$$

v závislosti na parametroch $\boldsymbol{\beta} = (\beta_0, \dots, \beta_d)'$. Označme $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \dots, \hat{\beta}_d)'$ vektor, pre ktorý (2.14) nadobúda minimum. Odhad regresnej funkcie získaný popísanou metódou je hodnota parametra $\hat{\beta}_0$.

Špeciálnym prípadom lokálne polynomiálnych odhadov, kde stupeň polynómu $d = 0$ sú **Nadaraya-Watsonove odhady**

$$\hat{m}_{NW}(x, h) = \frac{\sum_{i=1}^n K_h(x - x_i) Y_i}{\sum_{i=1}^n K_h(x - x_i)}. \quad (2.15)$$

V tomto prípade prekladáme dátu konštantnou funkciou. Môžeme si tiež všimnúť, že ak použijeme Nadaraya-Watsonove odhady s obdĺžnikovou jadrovou funkciou, dostaneme metódu kľzavých priemerov (2.2).

Druhým špeciálnym prípadom lokálne polynomiálnych odhadov, kde stupeň polynómu $d = 1$ a ide teda o priamku, sú **Lokálne lineárne estimátory**

$$\hat{m}_{LL}(x, h) = \frac{1}{n} \sum_{i=1}^n \frac{\hat{s}_2(x; h) - \hat{s}_1(x; h)(x_i - x)K_h(x - x_i)Y_i}{\hat{s}_2(x; h)\hat{s}_0(x; h) - \hat{s}_1(x; h)^2}, \quad (2.16)$$

kde

$$\hat{s}_r(x, h) = \frac{1}{n} \sum_{i=1}^n (x_i - x)^r K_h(x - x_i). \quad (2.17)$$

2.3 Miera polohy a miera variability

Ako bolo uvedené na začiatku kapitoly, v tejto podkapitole sa oboznámime ešte s niektorými pojмami zo štatistiky.

Majme *súbor hodnôт* nejakej náhodnej veličiny

$$x_1, x_2, x_3, \dots, x_n, \quad (2.18)$$

kde n je rozsah súboru a jeho hodnoty sú intervalového alebo pomerového typu. To znamená, že je možné stanoviť vzdialenosť medzi meranými hodnotami, a teda počítať s ich rozdielami. Tento súbor hodnôт môžeme analyzovať niekoľkými spôsobmi. Oboznámime sa niekoľkými charakteristikami takýchto náhodných veličín.

2.3.1 Aritmetický priemer

Na určenie hodnoty, okolo ktorej sa hodnoty jednotlivých pozorovaní nachádzajú je vhodný aritmetický priemer daného súboru.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (2.19)$$

Aritmetický priemer je tiež nazývaný miera polohy štatistických znakov.

2.3.2 Rozptyl a smerodajná odchýlka

Miery variability určujú, spôsob akým sú merané hodnoty usporiadane okolo strednej hodnoty. Najpoužívanejšie miery variability sú rozptyl a smerodajná odchýlka.

Rozptyl je často označovaný ako *stredná kvadratická odchýlka* a je definovaný

$$s^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2. \quad (2.20)$$

V prípade, že rozptyl počítame iba zo vzorky hodnôт, a nie z celého súboru hodnôт, vzorec sa zmení na

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2, \quad (2.21)$$

vďaka tejto úprave bude vypočítaný rozptyl jemne väčší, čím sa stáva nestranným odhadom skutočného rozptylu. Táto korektúra sa používa na zmiernenie zkreslenia výsledku pri výpočte zo zmenšeného počtu dát.

Smerodajná odchýlka je daná ako odmocnina z rozptylu

$$s = \sqrt{s^2}. \quad (2.22)$$

Čím väčšie hodnoty rozptyl a smerodajná odchýlka naberajú, tým viac sú hodnoty rozpísané od priemeru, naopak menšie hodnoty indikujú “tesnejšie” usporiadanie meraných hodnôt.

Kapitola 3

Analýza dát

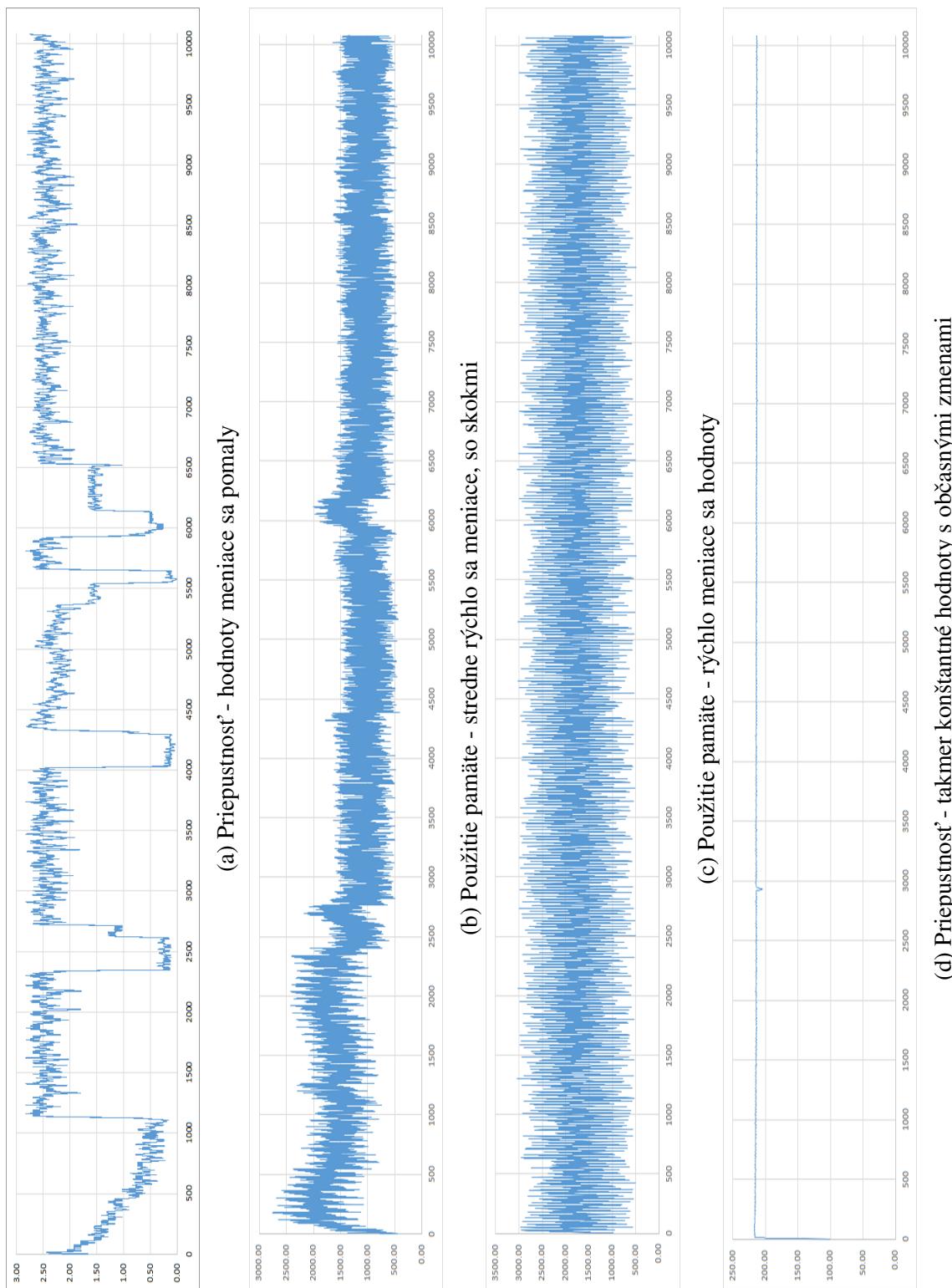
V tejto kapitole si ukážeme a zanalyzujeme konkrétné dáta vykreslené do grafov, s ktorými budeme pri vytváraní optimalizačného algoritmu pracovať.

Nástroj PerfCake môže testovať radu funkčností, ako bolo opísané v kapitole 1. Pre vytvorenie vhodného algoritmu máme k dispozícii štyri typy dát, na ktorých by mal algoritmus vhodne fungovať. Od rýchlo sa meniacich hodnôt dát (pripomínajúce šum), cez stredne rýchlo sa meniace hodnoty dát so “skokmi” a pomaly sa meniace hodnoty až po takmer konštantné hodnoty dát. Na obrázku 1.4 môžeme z vymenovaných vidieť dva typy dát, vykreslené do grafov z redukovaného počtu hodnôt, do podoby v akej sú potrebné pre analýzu výsledkov PerfCaku. V tejto práci, budeme pracovať s grafmi všetkých štyroch typov, každý pôvodne vykreslený z 10080 hodnôt, ktoré budeme následne redukovať, v podobe nami vyhovujúcej k naslednému spracovaniu a demonštrovaniu výsledkov. Tieto grafy môžeme vidieť na obrázku 3.1, sú to dáta vygenerované nástrojom PerfCake z testov, ktoré bežali 168 hodín, dva grafy z testov na použitú pamäť a dva grafy z testov na meranie príepustnosti.

Grafy z nástroja PerfCake sú určené ľudským užívateľom. Volným okom nie je možné rozoznať desaťtisíce hodnôt v jednom grafe, navyše 10080 hodnôt pochádzajúce z testov, ktoré bežali sedem dní a zaznamenávali dátu každú minútu, tieto testy môžu byť spustené na dlhší čas, poprípade zaznamenávať dátu častejšie. V takých prípadoch, môže byť záznamov niekoľko násobne viac, čo je aj v dnešnej dobe pamäťovo náročné a v prípade vykreslenia do grafu, náročné časovo. V neposlednom rade náročné na analýzu.

Z grafu je možné vyčítať niekoľko dôležitých informácií. Okolo akej hodnoty sa dátu pohybujú, ako veľmi a často “skáču” a teda ako veľmi sú stabilné, prípadne tesnosť ich rozptylenia. Samozrejme analýza grafu sa mení v závislosti na vlastnosti softvéru, ktorú meriame. Pri testovaní použitej pamäte je rozptyl hodnôt obvykle väčší ako pri testovaní príepustnosti, čo môžeme vidieť aj na obrázku 3.1. Vytvorené optimalizačné algoritmy dbajú na zachovanie, čo možno najmenšieho skresenia týchto informácií pri redukovaní počtu dát. Algoritmy sú konfigurovateľné, čiže veľkosť redukcie, na úkor väčšieho skresenia je nastaviteľný užívateľom, pomocou niekoľkých parametrov. Samozrejme sa predpokladá, že užívateľ, ktorý analyzuje výsledky, je oboznámený s aplikovaním optimalizačného algoritmu, poprípade si ho sám nakonfiguruje.

Postupné aplikovanie optimalizačných algoritmov na dátu, spolu s ich porovnaním si ukážeme v nasledujúcich kapitolách.



Obr. 3.1: Typy dát v grafoch

Kapitola 4

Vyhľadenie dát

V tejto kapitole aplikujeme jadrové odhady opisované v podkapitole 2.2 na dáta, čím ich vyhľadíme. Touto úpravou sa nestratí takmer žiadna podstatná informácia. Pre aplikovanie jadrových odhadov sme sa rozhodli po dôkladnej analýze dát a vyskúšaní kĺzavých priemerov.

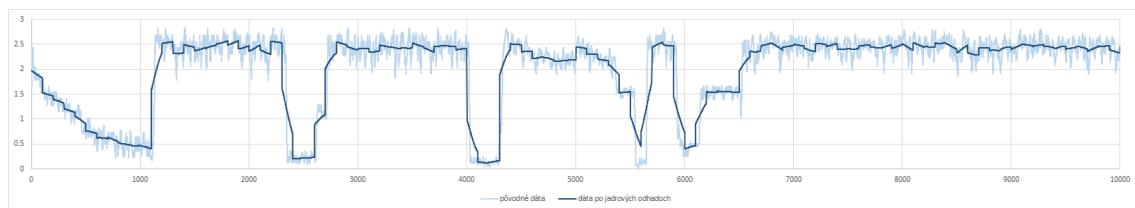
Pri počítaní jadrových odhadov budeme používať Nadaraya-Watsonove odhady po- písané v rovnici (2.15) s Epanečníkovým jadrom, ktoré môžme vidieť na obrázku 2.1, konkrétnie obrázok (a), a s použitím (2.13).

Použitý vzorec teda vyzerá

$$\hat{m}_{MW}(x; h) = \frac{\sum_{i=1}^n \frac{1}{h} \frac{3}{4} \left(1 - \left(\frac{x-x_i}{h}\right)^2\right) Y_i}{\sum_{i=1}^n \frac{1}{h} \frac{3}{4} \left(1 - \left(\frac{x-x_i}{h}\right)^2\right)}, \quad (4.1)$$

kde $\frac{3}{4} \left(1 - \left(\frac{x-x_i}{h}\right)^2\right)$ dáva 0, keď $|x - x_i| \geq h$.

Optimalizačný algoritmus, by mal byť spustený naraz s testovaním nástrojom PerfCake a mal by pracovať v reálnom čase, poprípade s nejakých zdržaním. Určite ale pri aplikovaní jadrových odhadov, nebudú k dispozícii celé dátá. Budeme teda počítať jadrové odhady vždy zo 100 záznamov. Čo by spôsobilo zdržanie maximálne 100 záznamov plus výpočet.

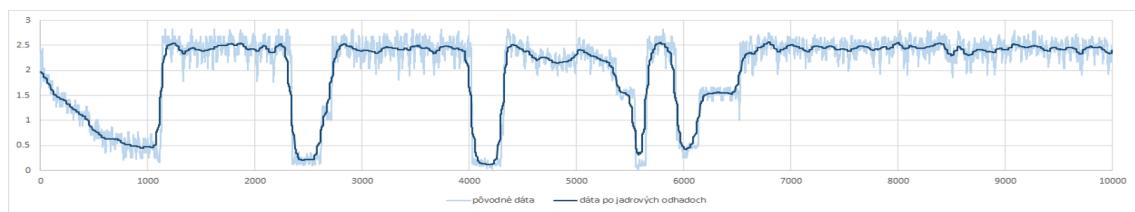


Obr. 4.1: Jadrové odhady bez prelínania

Na obrázku 4.1 môžme vidieť aplikovanie jadrových odhadov (4.1) na dáta, vždy pre každých 100 záznamov a dĺžkou vyhľadzovacieho okna 50. Graf, ktorý je výsledkom (tmavá čiara), je ale príliš "hranatý", čo je následkom rozdelenia dát na 100 záznamov

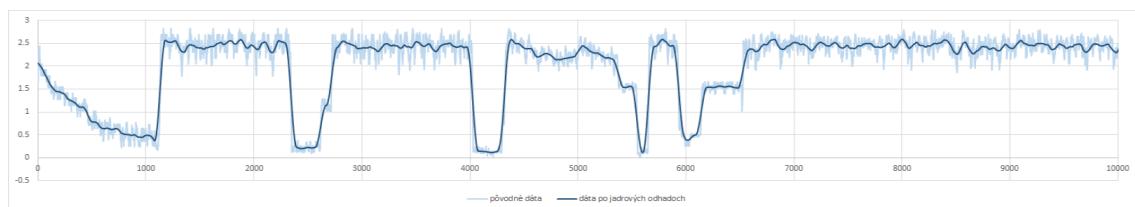
dlhé bloky. Tento algoritmus tiež z teoretického hľadiska zavádza mnoho chýb do odhadu, pretože na okraji intervalu nie je k dispozícii dostatočný záznamov pre spočítanie nestranného odhadu, čo sa pri opísanej aplikácii stáva na obidvoch okrajoch intervalu o dĺžke 100 záznamov.

Skúsmo upraviť vyhladzovací algoritmus tak, že budeme sice aplikovať jadrové odhady vždy na 100 záznamov dlhé bloky, ale tieto bloky sa budú prekrývať. Po aplikovaní na prvý blok záznamov, sa posunieme o 30 záznamov a aplikujeme jadrové odhady znova. To znamená, že pre väčšinu záznamov (okrem prvých 30 a posledných 30 záznamov) bude spočítaných niekoľko odhadov, pre výpočet výslednej hodnoty, tieto odhady spriemerujeme. Takto upravený algoritmus s dĺžkou vyhladzovacieho okna $h = 50$ aplikovaný na dátá môžeme vidieť na obrázku 4.2.



Obr. 4.2: Jadrové odhady s prelínaním

Takto upravený algoritmus na vyhladenie dát je ale skoro tri krát pomalší, pretože takmer všetky hodnoty odhadu sa počítajú tri krát. Skúsmo teda algoritmus upraviť ešte inak. Nebudeme dátá rozdeľovať na bloky z fixnou dĺžkou, keďže pri použití Epaničkovho jadra sa každá hodnota odhadu spočíta z množstva záznamov, ktoré určí dvojnásobok dĺžky vyhladzovacieho parametra. To znamená, že v prípade dĺžky okna 50, zoberieme pre každý odhad 50 záznamov naľavo, 50 napravo a z nich spočítame jadrový odhad. V prípade prvých 50 a posledných 50 použijeme nastáva skreslovanie, kvôli nedostatku záznamov pre spočítanie nestranného odhadu. Toto hraničné skreslovanie nastáva iba na začiatku a na konci celých dát. Týmto výpočtom simulujeme počítanie jadrových odhadov pre všetky dátá naraz, a zároveň spočítame odhad v každom bode iba raz. Použitie takého vyhladenia je zobrazené na obrázku 4.3, použitá dĺžka vyhladzovacieho okna je taktiež 50.

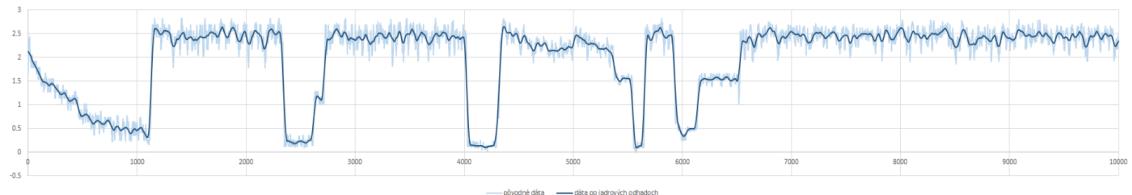


Obr. 4.3: Jadrové odhady s posúvaním bloku záznamov

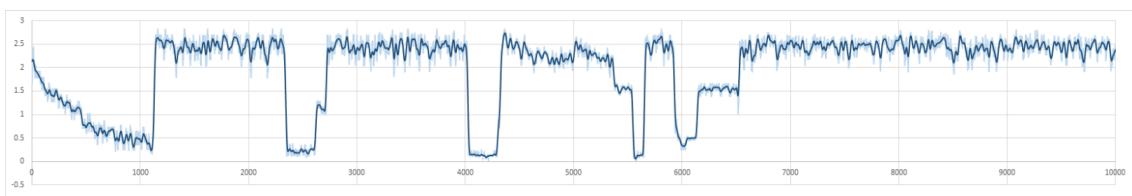
Zdržanie takého vyhladzovacieho algoritmu je naplnenia okna o dĺžke parametra h plus výpočet, a potom už iba výpočet pri pridaní ďalšieho záznamu. Dobu výpočtu je možné

pri použití Nadaraya-Watsonových odhadov skrátiť. V 4.1 môžeme vidieť, že váhová funkcia je závislá iba na parametri h a vzdialenosť danej hodnoty Y od bodu, pre ktorý sa odhad počíta. Preto si tieto váhy môžme spočítať iba raz a potom ich vhodne používať pre výpočet ostatných odhadov. Tento vyhľadzovací algoritmus s počítaním jadrových odhadov bude základom optimalizačných algoritmov, ktorých druhou časťou bude redukcia dát opísaná v ďalšej kapitole.

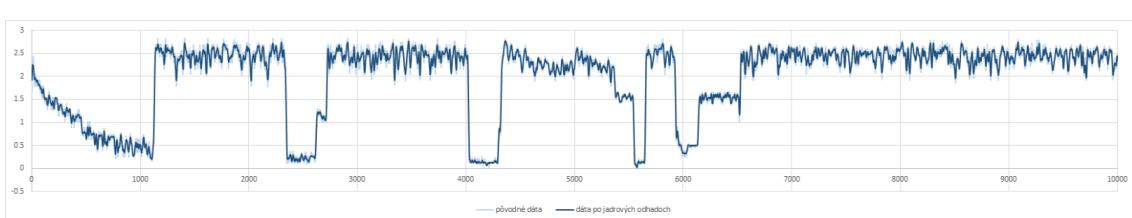
Ako už bolo spomínané dĺžka vyhľadzovacieho okna je veľmi dôležitým parametrom pri jadrových odhadoch. Počítať optimálnu dĺžku vyhľadzovacieho okna dynamicky pre každý nový záznam, by bolo veľmi náročné a navyše dáva tento parameter možnosť nastavenia veľkosti vyhľadzovacieho efektu. Preto bude dĺžka vyhľadzovacieho okna konfigurovateľný vstupný parameter pre algoritmus. Čím väčšiu hodnotu užívateľ zvolí, tým viac vyhľadené grafy budú výsledkom. Na obrázku 4.4 môžeme vidieť aplikovanie upravených jadrových odhadov s rôznymi dĺžkami vyhľadzovacieho okna, s dĺžkami 5, 15 a 30. Vyhľadzovanie s dĺžkou okna 50 bolo ukázané na obrázku 4.3.



(a) Upravené jadrové odhady s dĺžkou okna 30



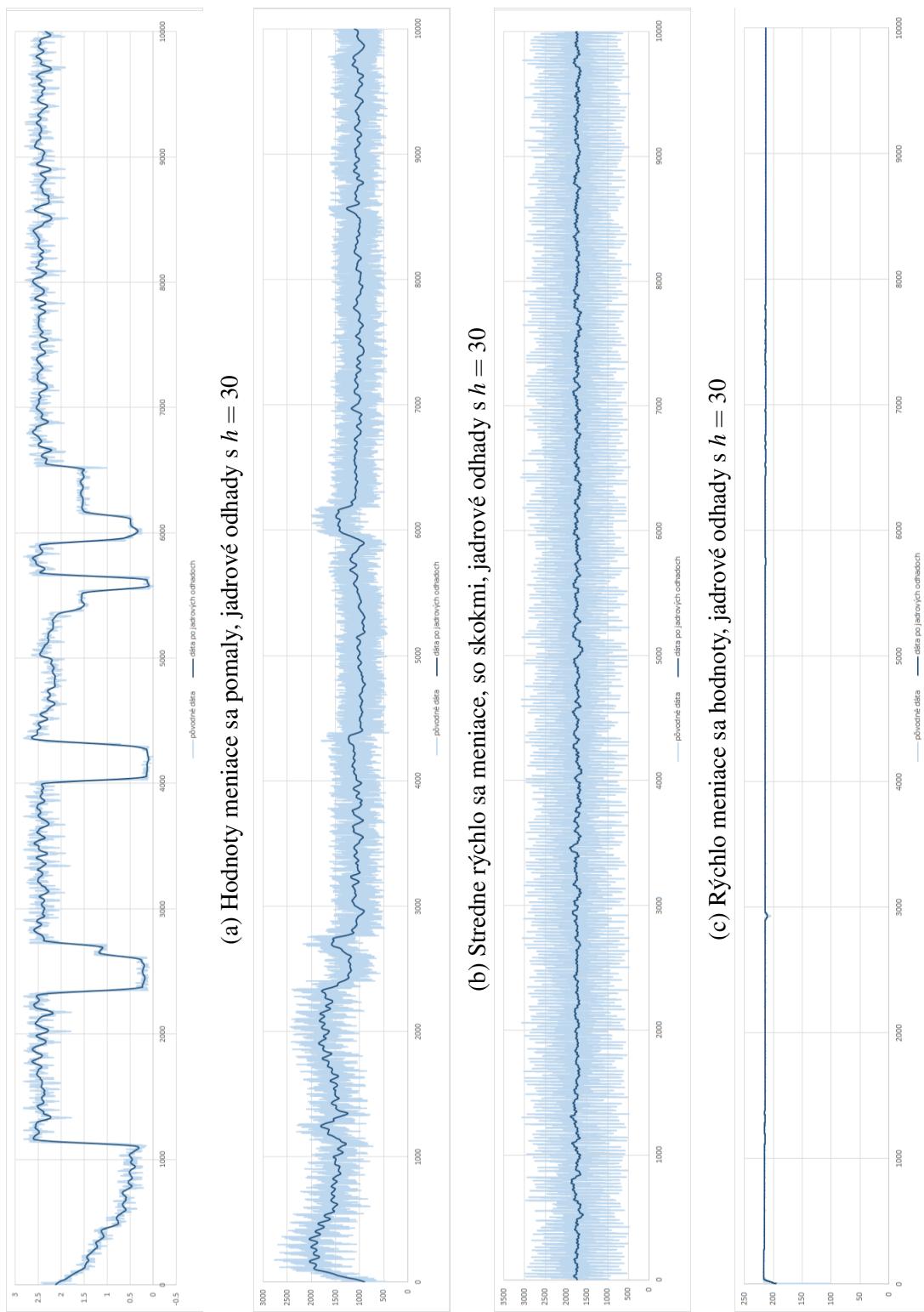
(b) Upravené jadrové odhady s dĺžkou okna 15



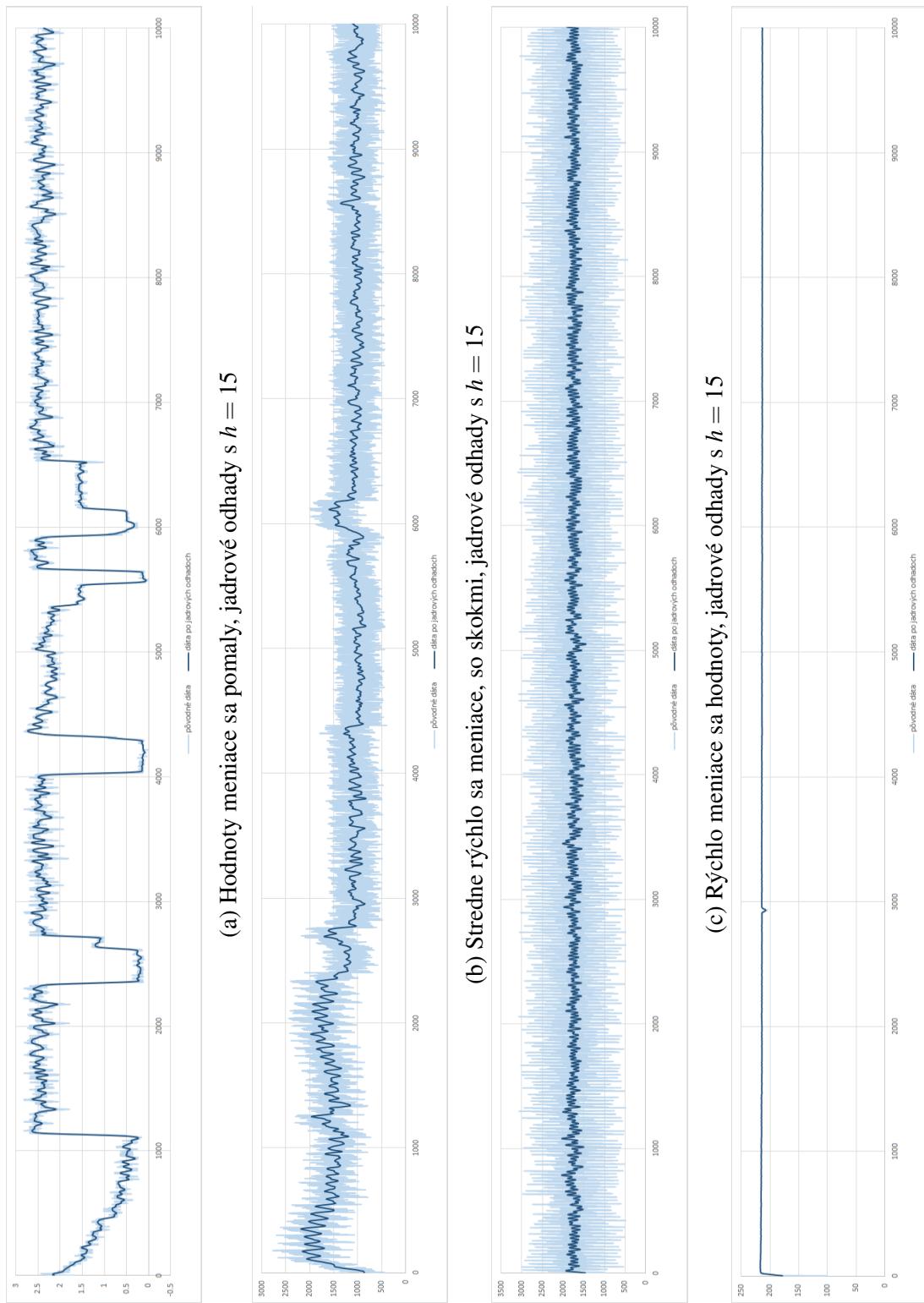
(c) Upravené jadrové odhady s dĺžkou okna 5

Obr. 4.4: Jadrové odhady s rôznymi hodnotami vychladzovacieho parametra

Vidíme, že zmena dĺžky vyhľadzovacieho okna nám naozaj dáva možnosť riadiť silu vyhľadenia. Vplyv jadrových odhadov na jede typ dát, typ a), sme si už ukázali, na ďalších obrázkoch, si ukážeme vyhľadené aj ostatné typy dát. Pre vyhľadenie na obrázku 4.5 bola



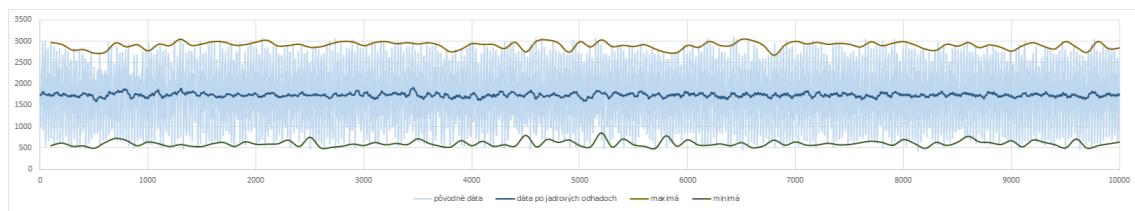
Obr. 4.5: Typy dát v grafoch s jadrovými odhadmi



Obr. 4.6: Typy dát v grafoch s jadrovými odhadmi

zvolená dĺžka okna 30 a na obrázku 4.6 dĺžka okna 15.

Pri type c) sa hodnoty menia veľmi rýchlo a bez akéhokoľvek trendu, pripomínajú biely šum. Pri takýchto dátach, by vplyvom vyhladenia mohlo dôjsť k strate informácie o veľkosti rozptylu dát. Samozrejme to závisí na testovanej vlastnosti a type softvéra. Obvykle sa pri testovaní použítej zameriava na hľadanie “únikov pamäte” (skôr známe pod anglickým pojmom - *memory leak*). V tomto prípade je takéto vyhladenie viac ako prospešné, pretože ukazuje, že nenastali žiadne výkyvy. Naopak ak by sa pri testovaní zameriaval na rozptyl hodnôt použitej pamäte, poprípade na nejaké lokálne maximá, vyhladenie by takéto informácie zničilo. Pre takéto prípady môžeme pridať možnosť ohraňčujúcich kriviek, ktoré doplnia stratené informácie. Použitie týchto kriviek je na obrázku 4.7.



Obr. 4.7: Ohraničujúce krivky

Dáta, z ktorých sú krivky vykreslené sú tvorené maximami a minimami z každých 100 vzoriek. To znamená, že pri 10080 záznamov potrebujeme ďalších 202 záznamov pre tieto krivky. Vykreslovanie ohraničujúcich kriviek, ktoré ukazujú akési lokálne minimá a maximá, bude ďalší konfigurovateľný parameter pre výsledný optimalizačný algoritmus.

Kapitola 5

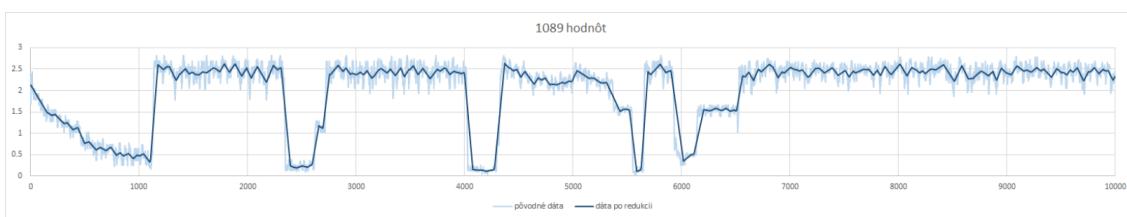
Redukcia dát

V tejto kapitole aplikujeme na už vyhadené dátá redukciu. V každej podkapitole si popíšeme a použijeme iný redukčný algoritmus.

5.1 Redukcia pomocou smerodajnej odchýlky a priemeru

Pri aplikovaní tejto redukcie je hlavná myšlienka taká, že ak sa dátá veľmi nemenia môžme ich redukovať. Na túto redukciu nám poslúži smerodajná odchýlka.

Algoritmus pre každú hodnotu spočíta smerodajnú odchýlku z posledných 10 hodnôt včetne danej hodnoty, ak je smerodajná odchýlka spočítaná s danou niekoľko krát väčšia ako smerodajná odchýlka spočítaná bez tejto hodnoty pre predchádzajúci záznam, hodnota sa uloží. Číslo, ktoré určuje koľkokrát sa musí zmeniť smerodajná odchýlka aby sa daná hodnota uložila, bude ďalším voliteľným parametrom, ktorý riadi silu redukcie (ďalej len parameter redukcie). Vplyv tohto parametra na dátá ukazuje obrázok 5.1, redukcia bola aplikovaná na dátá vyhadené oknom o šírke 30.



(a) Redukcia pomocou smerodajnej odchýlky, parameter = 1,2



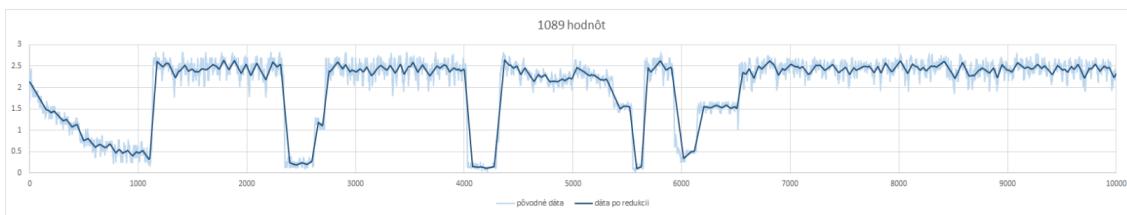
(b) Redukcia pomocou smerodajnej odchýlky, parameter = 1,3



(c) Redukcia pomocou smerodajnej odchýlky, parameter = 1,4

Obr. 5.1: Redukcia pomocou smerodajnej odchýlky, použitá pôvodná hodnota

Malou obmenou tohto algoritmu môže byť, ukladanie priemeru všetkých 50 záznamov namiesto danej hodnoty, ktorá rozhoduje o uložení respektíve vyhodení záznamu. Nasledujúci obrázok 5.2 ukazuje túto obmenu na rovnakých dátach ako obrázok 5.1.



(a) Redukcia pomocou smerodajnej odchýlky, parameter = 1,2



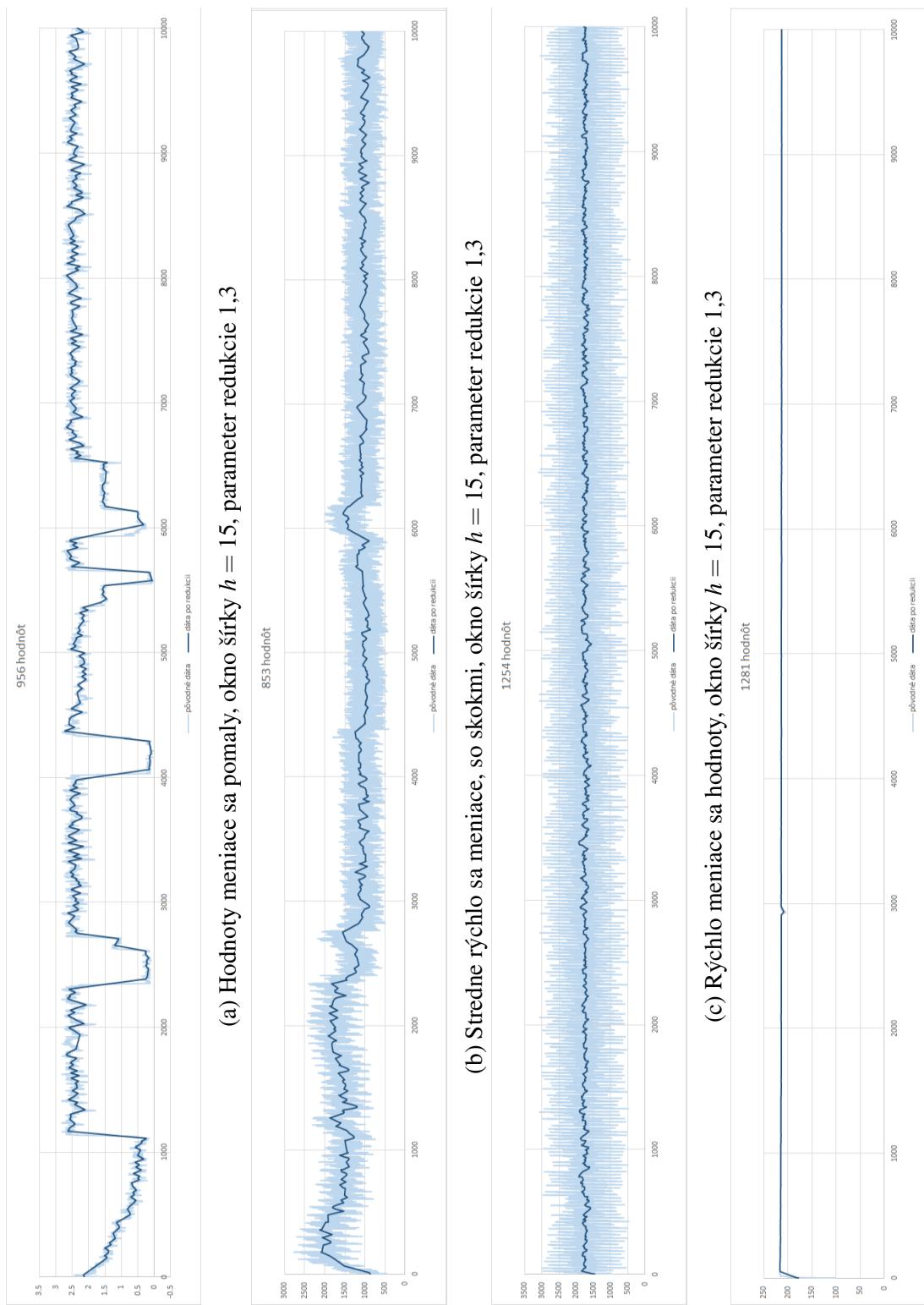
(b) Redukcia pomocou smerodajnej odchýlky, parameter = 1,3



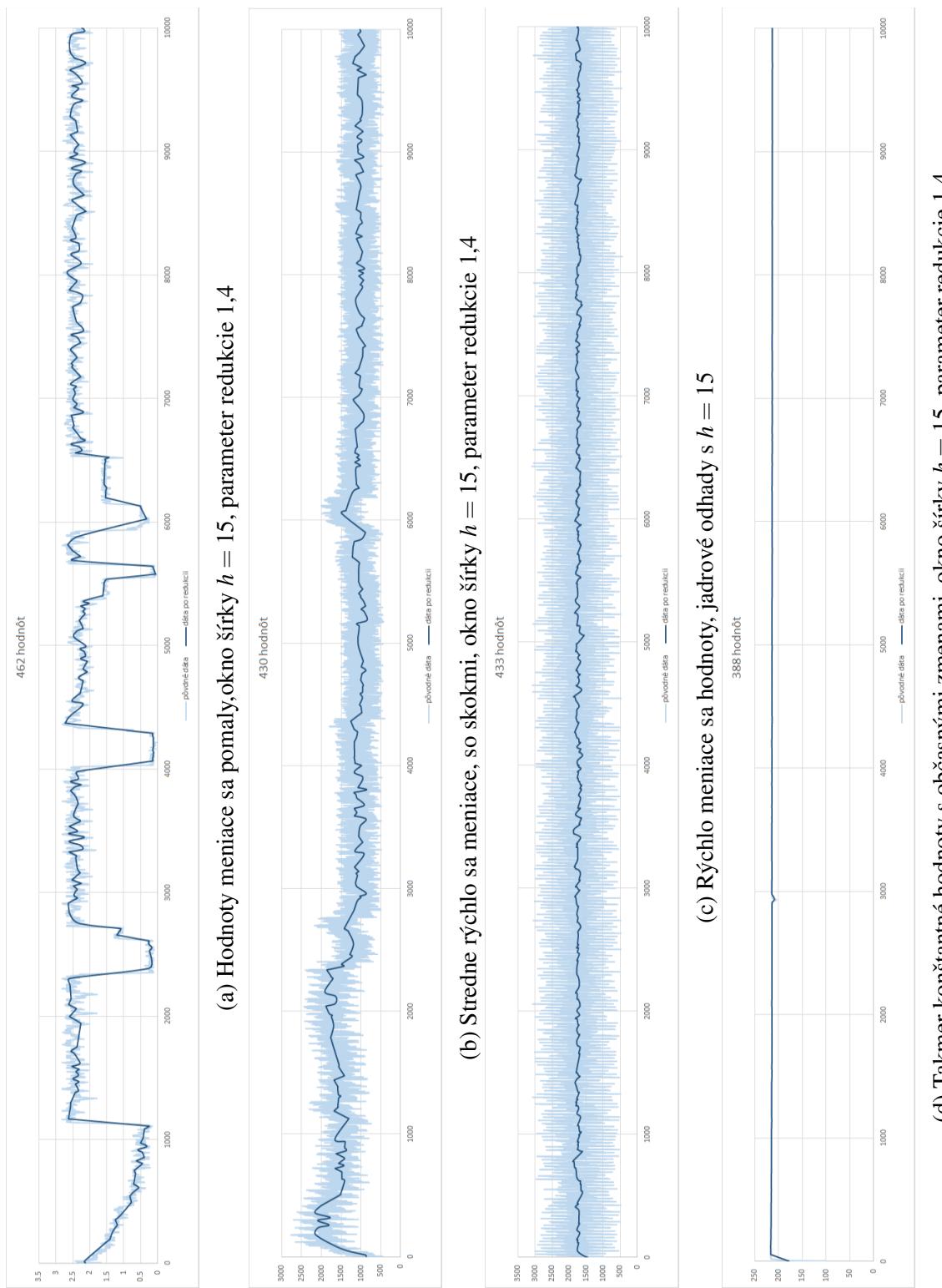
(c) Redukcia pomocou smerodajnej odchýlky, parameter = 1,4

Obr. 5.2: Redukcia pomocou smerodajnej odchýlky, použitý priemer hodnôt

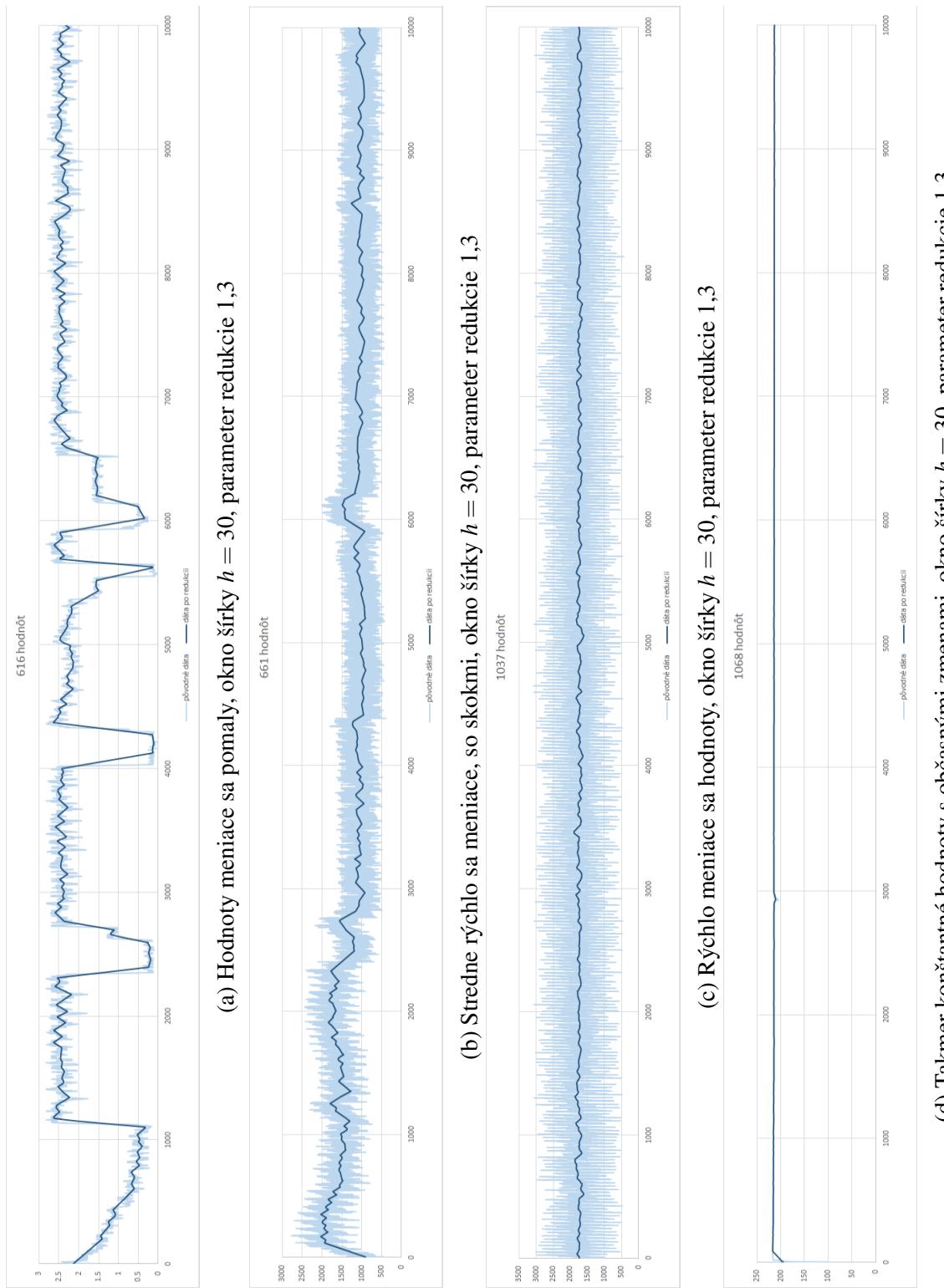
Takto upravený algoritmus nezahadzuje informáciu z vylúčených záznamov, pretože sa použije v najbližšej uloženej hodnote v podobe priemeru. Preto tento algoritmus s používaním priemeru dáva reálnejšie výsledky. Na nasledujúcich obrázkoch si ukážeme



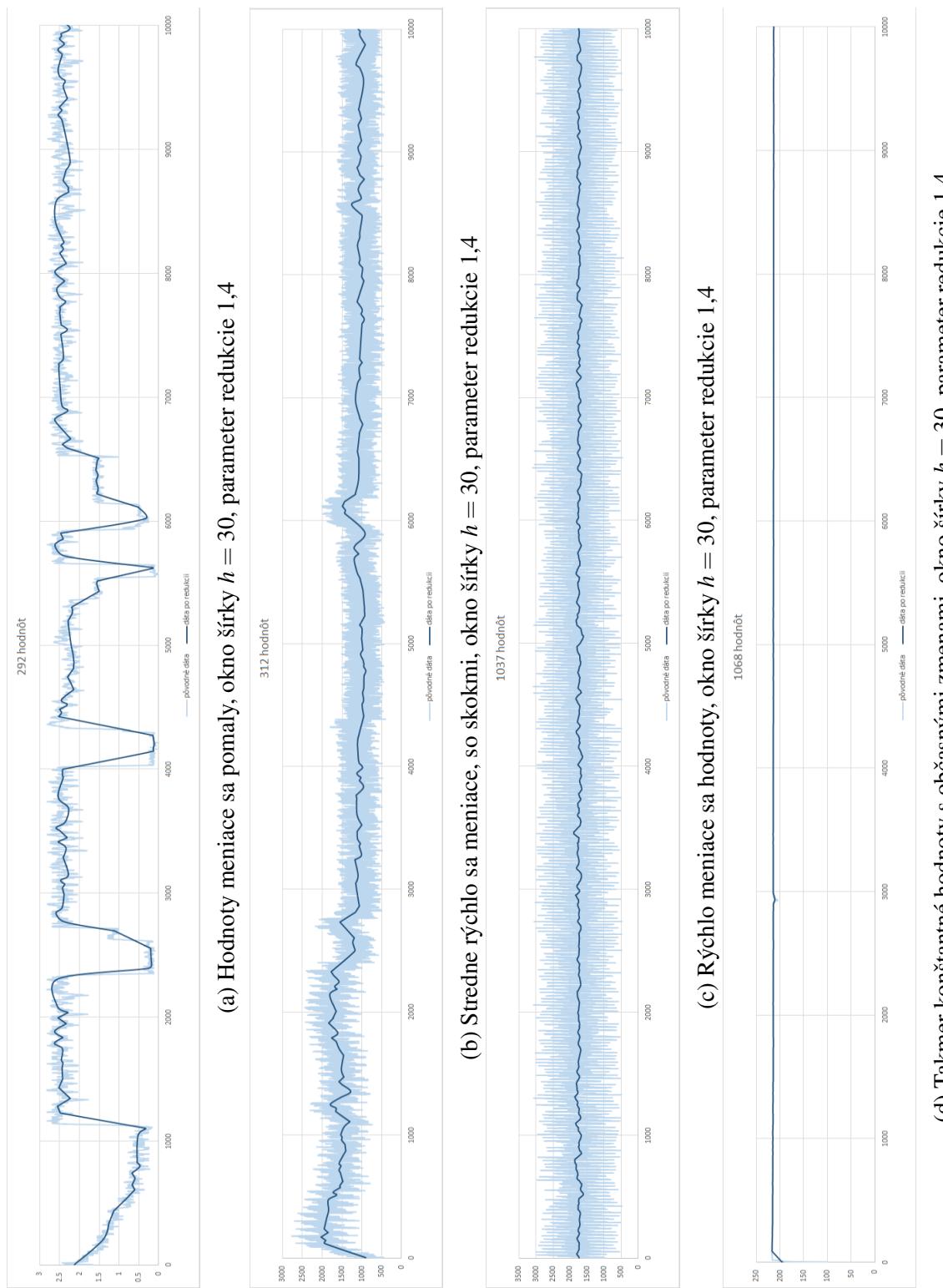
Obr. 5.3: Typy dát v grafoch po redukcii pomocou smerodajnej odchýlky



Obr. 5.4: Typy dát v grafoch po redukcii pomocou smerodajnej odchýlky



Obr. 5.5: Typy dát v grafoch po redukcii pomocou smerodajnej odchýlky



Obr. 5.6: Typy dát v grafoch po redukcii pomocou smerodajnej odchýlky

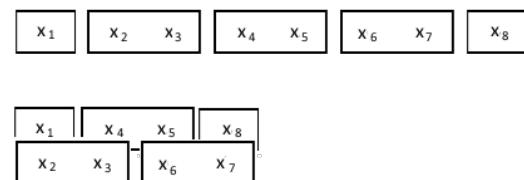
aplikovanie algoritmu s použitím priemeru na všetkých typoch dát. Obrázok 5.3 ukazuje dátá po vyhladení oknom šírky 15 a po redukcii pomocou smerodajnej odchýlky s parametrom redukcie 1,3. Obrázok 5.4, okno šírky 15 a parameter redukcie 1,4, obrázok 5.5 okno šírky 30 a parameter redukcie 1,3 a obrázok 5.6 okno šírky 30 a parameter redukcie 1,4.

5.2 Redukcia podobná redukcii pri zrýchľovaní zvuku

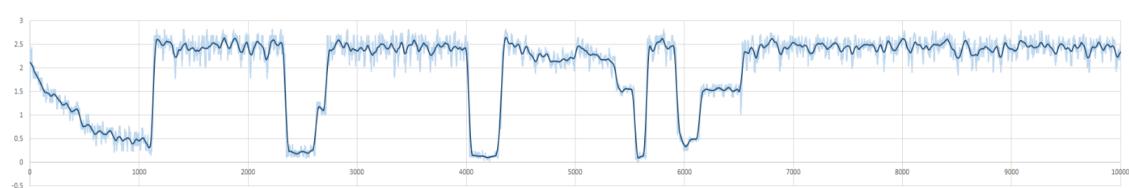
Pri zrýchľovaní zvuku je potrebné modifikovať časovú os, aby sme zachovali výšku zvukového signálu. Zjednodušene sa takáto modifikácia aplikuje tak, že sa signál rozdelí na rovnako dlhé úseky, ktoré sa následne prekryjú na dosiahnutie požadovanej dĺžky signálu. Tam kde sa úseky prekryjú sa dané vzorky zpriemerujú. Táto modifikácia sa používa napríklad v prehliadači VLC. [zdroják VLC]

My na dátá použijeme rovnaký algoritmus, aby sa ale priemerovali čo najbližšie záznamy, rozdelíme dátá na úseky o dĺžke 2 záznamy (okrem prvého a posledného, tie majú dĺžku 1 záznam). Tieto cez seba prekryjeme, toto rozdelenie a prekrytie je znázornené na obrázku 5.7. Tie záznamy, ktoré sa prekryjú priemerujeme. Takto zredukujeme dátá na polovicu. Aplikovaním rovnakej modifikácie napríklad 4-krát, môžeme dátá zredukovať na šestnásťinu.

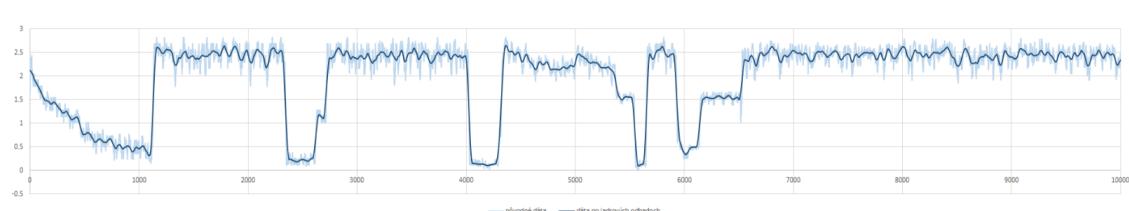
Na nasledujúcim obrázku 5.8 si ukážeme opísaný algoritmus použitý na dátá vyhladené šírkou okna 30, raz, 2-krát, 3-krát, 4-krát, 5-krát.



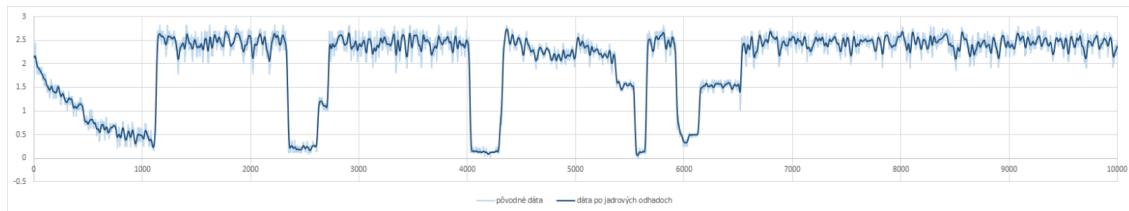
Obr. 5.7: Rozdelenie a prekrytie dát



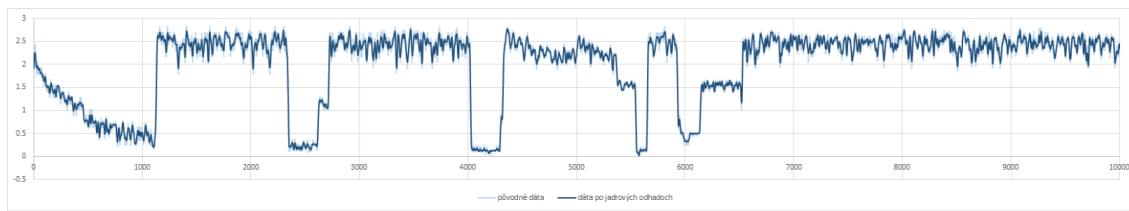
(a) Upravené jadrové odhady s dĺžkou okna 30



(b) Upravené jadrové odhady s dĺžkou okna 30



(c) Upravené jadrové odhady s dĺžkou okna 15

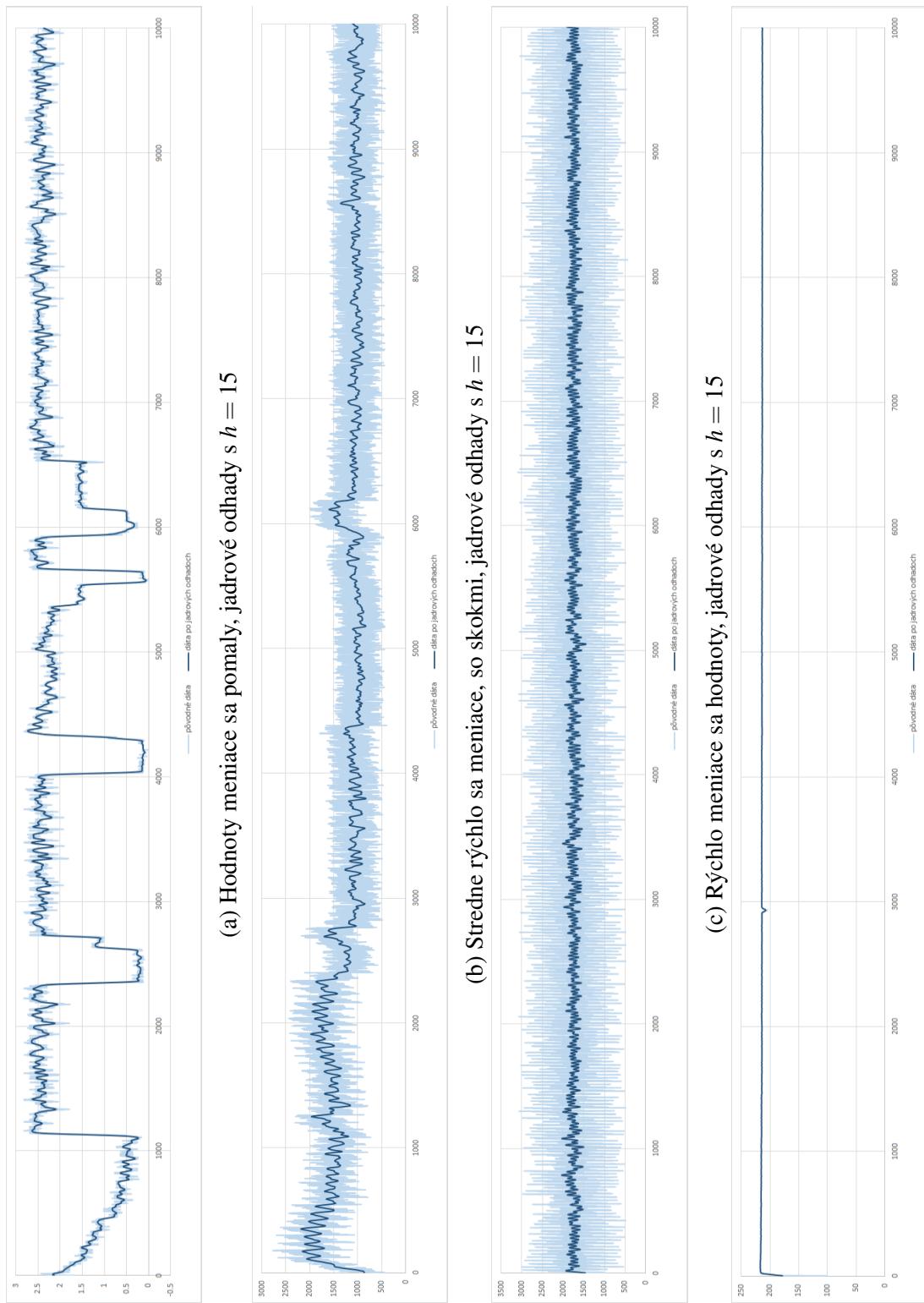


(d) Upravené jadrové odhady s dĺžkou okna 5

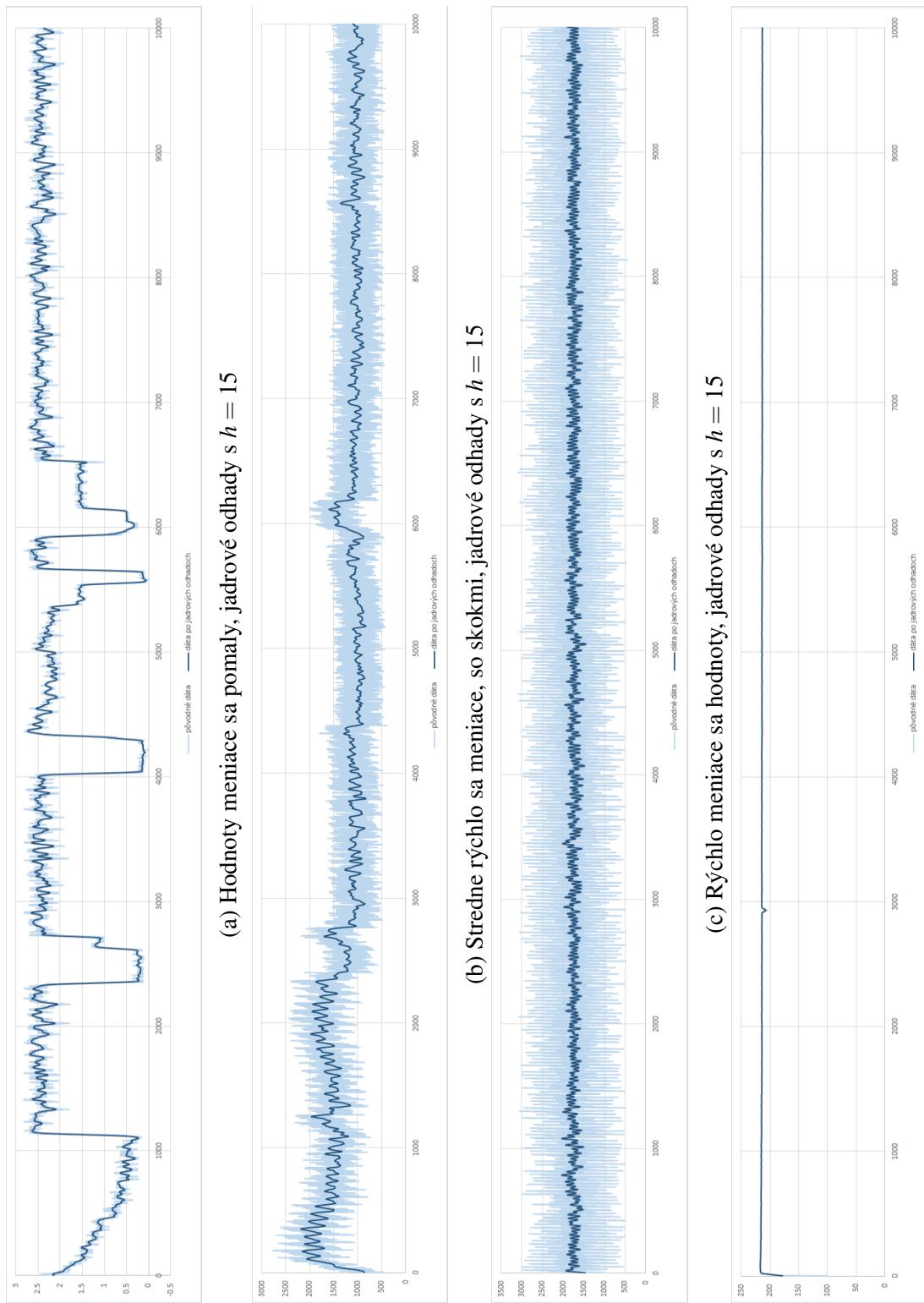
Obr. 5.8: Jadrové odhady s rôznymi hodnotami vychladzovacieho parametra

Na zvyšné typy dát použijeme algoritmus 3-krát a 4-krát a použité šírky okien sú 15 a 30. Výsledky sú znázornené na obrázkoch [5.9](#), [5.10](#), [5.11](#) a [5.12](#).

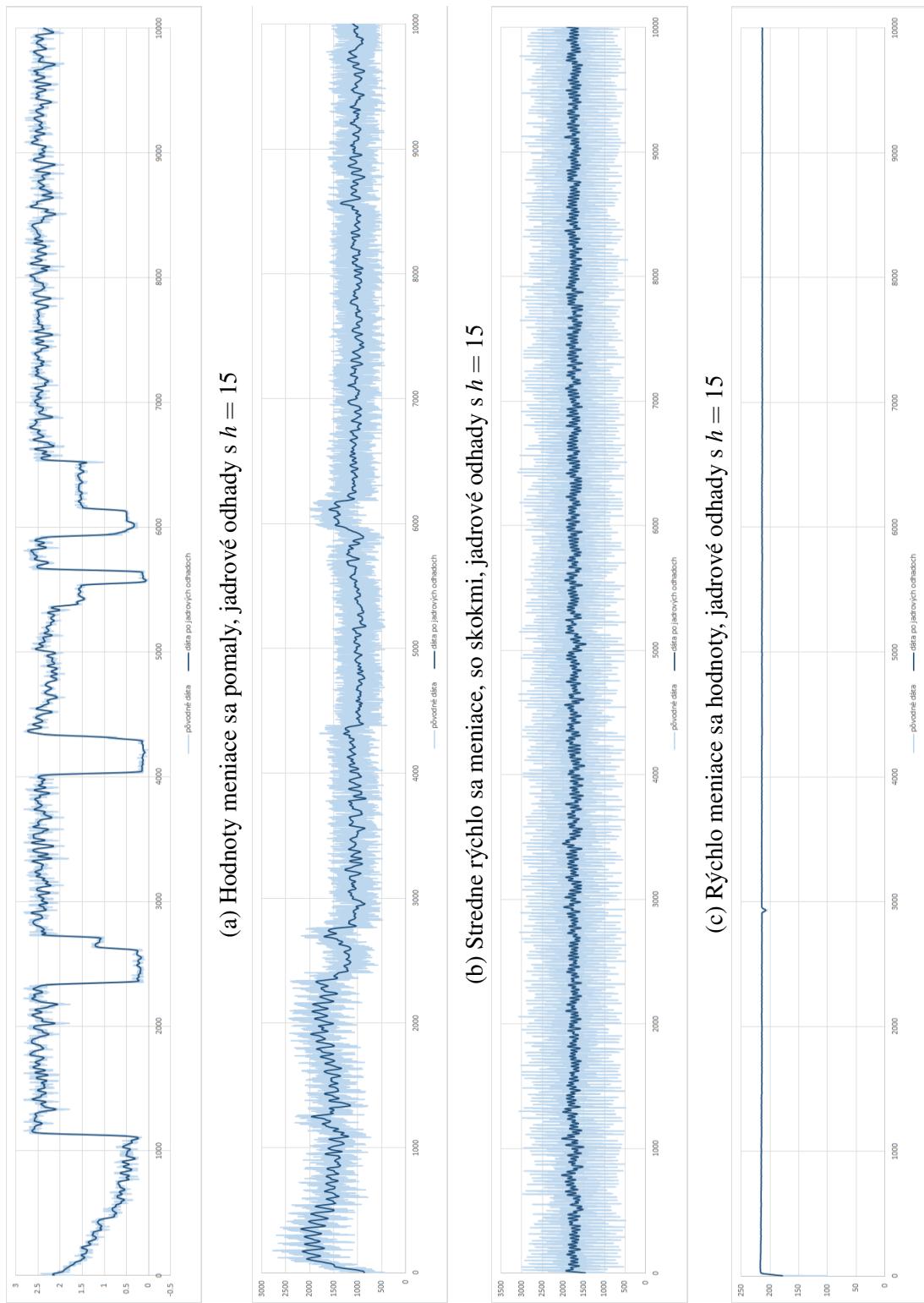
5.3 Redukcia pomocou jadrových odhadov



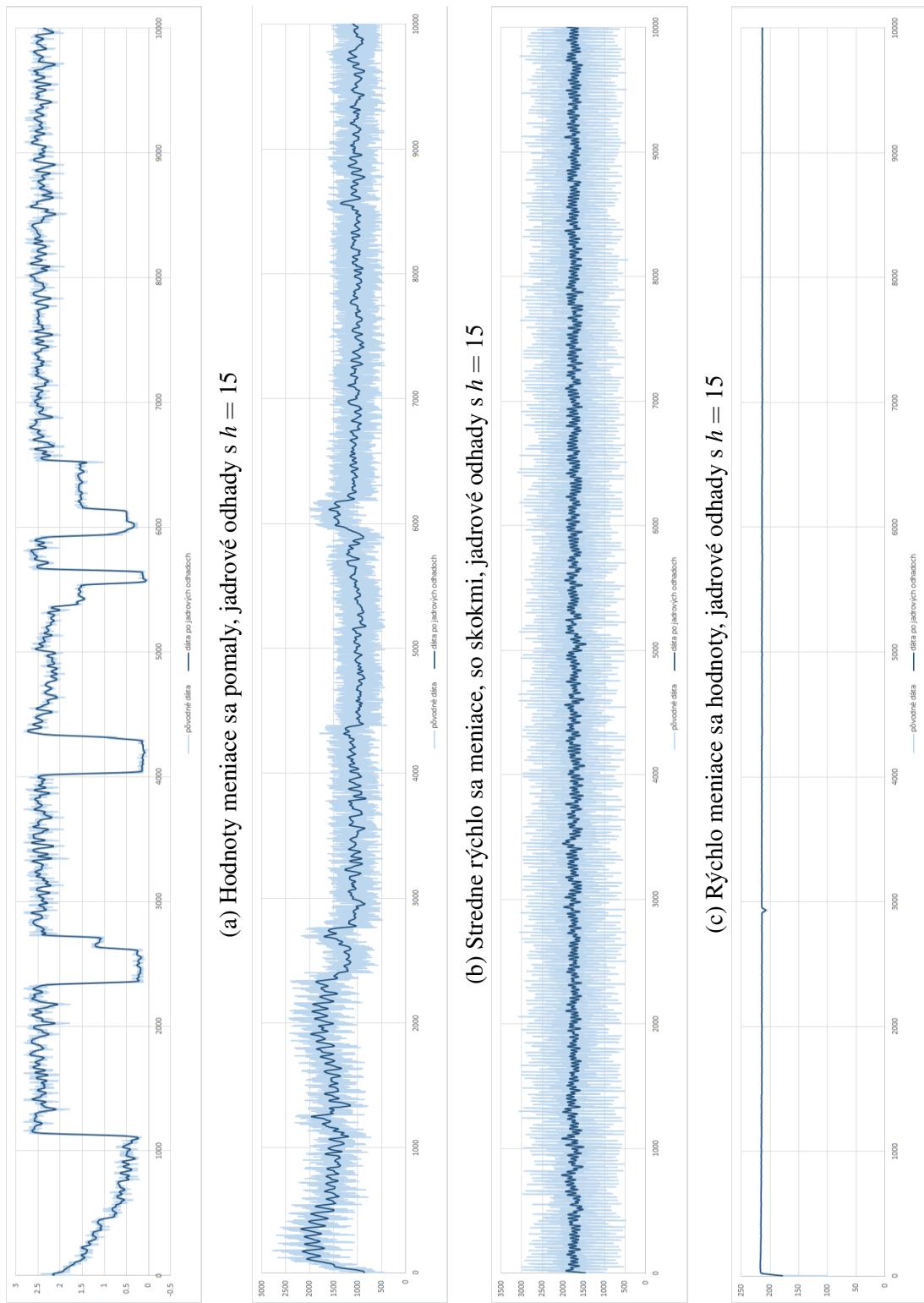
Obr. 5.9: Typy dát v grafoch s jadrovými odhadmi



Obr. 5.10: Typy dát v grafoch s jadrovými odhadmi



Obr. 5.11: Typy dát v grafoch s jadrovými odhadmi



Obr. 5.12: Typy dát v grafoch s jadrovými odhadmi

Závěr

Zde můžete napsat závěr. Zde můžete napsat závěr.

Zde můžete napsat závěr. Zde můžete napsat závěr. Zde můžete napsat závěr. Zde můžete napsat závěr. Zde můžete napsat závěr. Zde můžete napsat závěr. Zde můžete napsat závěr. Zde můžete napsat závěr. Zde můžete napsat závěr. Zde můžete napsat závěr. Zde můžete napsat závěr. Zde můžete napsat závěr. Zde můžete napsat závěr. Zde můžete napsat závěr. Zde můžete napsat závěr. Zde můžete napsat závěr. Zde můžete napsat závěr. Zde můžete napsat závěr. Zde můžete napsat závěr.

Quisque facilisis auctor sapien. Pellentesque gravida hendrerit lectus. Mauris rutrum sodales sapien. Fusce hendrerit sem vel lorem. Integer pellentesque massa vel augue. Integer elit tortor, feugiat quis, sagittis et, ornare non, lacus. Vestibulum posuere pellentesque eros. Quisque venenatis ipsum dictum nulla. Aliquam quis quam non metus eleifend interdum. Nam eget sapien ac mauris malesuada adipiscing. Etiam eleifend neque sed quam. Nulla facilisi. Proin a ligula. Sed id dui eu nibh egestas tincidunt. Suspendisse arcu.

Maecenas duis. Aliquam volutpat auctor lorem. Cras placerat est vitae lectus. Curabitur massa lectus, rutrum euismod, dignissim ut, dapibus a, odio. Ut eros erat, vulputate ut, interdum non, porta eu, erat. Cras fermentum, felis in porta congue, velit leo facilisis odio, vitae consectetur lorem quam vitae orci. Sed ultrices, pede eu placerat auctor, ante ligula rutrum tellus, vel posuere nibh lacus nec nibh. Maecenas laoreet dolor at enim. Donec molestie dolor nec metus. Vestibulum libero. Sed quis erat. Sed tristique. Duis pede leo, fermentum quis, consectetur eget, vulputate sit amet, erat.

Donec vitae velit. Suspendisse porta fermentum mauris. Ut vel nunc non mauris pharetra varius. Duis consequat libero quis urna. Maecenas at ante. Vivamus varius, wisi sed egestas tristique, odio wisi luctus nulla, lobortis dictum dolor ligula in lacus. Vivamus aliquam, urna sed interdum porttitor, metus orci interdum odio, sit amet euismod lectus felis et leo. Praesent ac wisi. Nam suscipit vestibulum sem. Praesent eu ipsum vitae pede cursus venenatis. Duis sed odio. Vestibulum eleifend. Nulla ut massa. Proin rutrum mattis sapien. Curabitur dictum gravida ante.

Phasellus placerat vulputate quam. Maecenas at tellus. Pellentesque neque diam, dignissim ac, venenatis vitae, consequat ut, lacus. Nam nibh. Vestibulum fringilla arcu mollis arcu. Sed et turpis. Donec sem tellus, volutpat et, varius eu, commodo sed, lectus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque enim arcu, suscipit nec, tempus at, imperdiet vel, metus. Morbi volutpat purus at erat. Donec dignissim, sem id semper

tempus, nibh massa eleifend turpis, sed pellentesque wisi purus sed libero. Nullam lobortis tortor vel risus. Pellentesque consequat nulla eu tellus. Donec velit. Aliquam fermentum, wisi ac rhoncus iaculis, tellus nunc malesuada orci, quis volutpat dui magna id mi. Nunc vel ante. Duis vitae lacus. Cras nec ipsum.

Morbi nunc. Aliquam consectetur varius nulla. Phasellus eros. Cras dapibus porttitor risus. Maecenas ultrices mi sed diam. Praesent gravida velit at elit vehicula porttitor. Phasellus nisl mi, sagittis ac, pulvinar id, gravida sit amet, erat. Vestibulum est. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur id sem elementum leo rutrum hendrerit. Ut at mi. Donec tincidunt faucibus massa. Sed turpis quam, sollicitudin a, hendrerit eget, pretium ut, nisl. Duis hendrerit ligula. Nunc pulvinar congue urna.

Nunc velit. Nullam elit sapien, eleifend eu, commodo nec, semper sit amet, elit. Nulla lectus risus, condimentum ut, laoreet eget, viverra nec, odio. Proin lobortis. Curabitur dictum arcu vel wisi. Cras id nulla venenatis tortor congue ultrices. Pellentesque eget pede. Sed eleifend sagittis elit. Nam sed tellus sit amet lectus ullamcorper tristique. Mauris enim sem, tristique eu, accumsan at, scelerisque vulputate, neque. Quisque lacus. Donec et ipsum sit amet elit nonummy aliquet. Sed viverra nisl at sem. Nam diam. Mauris ut dolor. Curabitur ornare tortor cursus velit.

Morbi tincidunt posuere arcu. Cras venenatis est vitae dolor. Vivamus scelerisque semper mi. Donec ipsum arcu, consequat scelerisque, viverra id, dictum at, metus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut pede sem, tempus ut, porttitor bibendum, molestie eu, elit. Suspendisse potenti. Sed id lectus sit amet purus faucibus vehicula. Praesent sed sem non dui pharetra interdum. Nam viverra ultrices magna.

Aenean laoreet aliquam orci. Nunc interdum elementum urna. Quisque erat. Nullam tempor neque. Maecenas velit nibh, scelerisque a, consequat ut, viverra in, enim. Duis magna. Donec odio neque, tristique et, tincidunt eu, rhoncus ac, nunc. Mauris malesuada malesuada elit. Etiam lacus mauris, pretium vel, blandit in, ultricies id, libero. Phasellus bibendum erat ut diam. In congue imperdiet lectus.

Aenean scelerisque. Fusce pretium porttitor lorem. In hac habitasse platea dictumst. Nulla sit amet nisl at sapien egestas pretium. Nunc non tellus. Vivamus aliquet. Nam adipiscing euismod dolor. Aliquam erat volutpat. Nulla ut ipsum. Quisque tincidunt auctor augue. Nunc imperdiet ipsum eget elit. Aliquam quam leo, consectetur non, ornare sit amet, tristique quis, felis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque interdum quam sit amet mi. Pellentesque mauris dui, dictum a, adipiscing ac, fermentum sit amet, lorem.

Ut quis wisi. Praesent quis massa. Vivamus egestas risus eget lacus. Nunc tincidunt, risus quis bibendum facilisis, lorem purus rutrum neque, nec porta tortor urna quis orci. Aenean aliquet, libero semper volutpat luctus, pede erat lacinia augue, quis rutrum sem ipsum sit amet pede. Vestibulum aliquet, nibh sed iaculis sagittis, odio dolor blandit augue, eget mollis urna tellus id tellus. Aenean aliquet aliquam nunc. Nulla ultricies justo eget orci. Phasellus tristique fermentum leo. Sed massa metus, sagittis ut, semper ut, pharetra vel, erat. Aliquam quam turpis, egestas vel, elementum in, egestas sit amet, lorem. Duis convallis, wisi sit amet mollis molestie, libero mauris porta dui, vitae aliquam arcu turpis ac sem. Aliquam aliquet dapibus metus.

Vivamus commodo eros eleifend dui. Vestibulum in leo eu erat tristique mattis. Cras at elit. Cras pellentesque. Nullam id lacus sit amet libero aliquet hendrerit. Proin placerat,

mi non elementum laoreet, eros elit tincidunt magna, a rhoncus sem arcu id odio. Nulla eget leo a leo egestas facilisis. Curabitur quis velit. Phasellus aliquam, tortor nec ornare rhoncus, purus urna posuere velit, et commodo risus tellus quis tellus. Vivamus leo turpis, tempus sit amet, tristique vitae, laoreet quis, odio. Proin scelerisque bibendum ipsum. Etiam nisl. Praesent vel dolor. Pellentesque vel magna. Curabitur urna. Vivamus congue urna in velit. Etiam ullamcorper elementum dui. Praesent non urna. Sed placerat quam non mi. Pellentesque diam magna, ultricies eget, ultrices placerat, adipiscing rutrum, sem.

Morbi sem. Nulla facilisi. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Nulla facilisi. Morbi sagittis ultrices libero. Praesent eu ligula sed sapien auctor sagittis. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Donec vel nunc. Nunc fermentum, lacus id aliquam porta, dui tortor euismod eros, vel molestie ipsum purus eu lacus. Vivamus pede arcu, euismod ac, tempus id, pretium et, lacus. Curabitur sodales dapibus urna. Nunc eu sapien. Donec eget nunc a pede dictum pretium. Proin mauris. Vivamus luctus libero vel nibh.

Fusce tristique risus id wisi. Integer molestie massa id sem. Vestibulum vel dolor. Pellentesque vel urna vel risus ultricies elementum. Quisque sapien urna, blandit nec, iaculis ac, viverra in, odio. In hac habitasse platea dictumst. Morbi neque lacus, convallis vitae, commodo ac, fermentum eu, velit. Sed in orci. In fringilla turpis non arcu. Donec in ante. Phasellus tempor feugiat velit. Aenean varius massa non turpis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae;

Aliquam tortor. Morbi ipsum massa, imperdiet non, consectetuer vel, feugiat vel, lorem. Quisque eget lorem nec elit malesuada vestibulum. Quisque sollicitudin ipsum vel sem. Nulla enim. Proin nonummy felis vitae felis. Nullam pellentesque. Duis rutrum feugiat felis. Mauris vel pede sed libero tincidunt mollis. Phasellus sed urna rhoncus diam euismod bibendum. Phasellus sed nisl. Integer condimentum justo id orci iaculis varius. Quisque et lacus. Phasellus elementum, justo at dignissim auctor, wisi odio lobortis arcu, sed sollicitudin felis felis eu neque. Praesent at lacus.

Vivamus sit amet pede. Duis interdum, nunc eget rutrum dignissim, nisl diam luctus leo, et tincidunt velit nisl id tellus. In lorem tellus, aliquet vitae, porta in, aliquet sed, lectus. Phasellus sodales. Ut varius scelerisque erat. In vel nibh eu eros imperdiet rutrum. Donec ac odio nec neque vulputate suscipit. Nam nec magna. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Nullam porta, odio et sagittis iaculis, wisi neque fringilla sapien, vel commodo lorem lorem id elit. Ut sem lectus, scelerisque eget, placerat et, tincidunt scelerisque, ligula. Pellentesque non orci.

Etiam vel ipsum. Morbi facilisis vestibulum nisl. Praesent cursus laoreet felis. Integer adipiscing pretium orci. Nulla facilisi. Quisque posuere bibendum purus. Nulla quam mauris, cursus eget, convallis ac, molestie non, enim. Aliquam congue. Quisque sagittis nonummy sapien. Proin molestie sem vitae urna. Maecenas lorem. Vivamus viverra consequat enim.

Seznam použité literatury

- [1] S. J. Monaquel a K. M. Schmidt, *On M-functions and operator theory for non-self-adjoint discrete Hamiltonian systems*, v „Special Issue: 65th birthday of Prof. Desmond Evans“, J.Comput. Appl. Math. **208** (2007), č. 1, 82–101.
- [2] M. Murata, *Positive solutions and large time behaviors of Schrödinger semigroups, Simon's problem*, J. Funct. Anal. **56** (1984), č. 3, 300–310.
- [3] J. Qi a S. Chen, *Strong limit-point classification of singular Hamiltonian expressions*, Proc. Amer. Math. Soc. **132** (2004), č. 6, 1667–1674 (elektronicky).
- [4] Z. Pospíšil, *An inverse problem for matrix trigonometric and hyperbolic functions on measure chains*, v „Colloquium on Differential and Difference Equations — CDDE 2002“ (Brno, 2002), Folia Fac. Sci. Natur. Univ. Masaryk. Brun. Math. **13**, str. 205–211, Masarykova univerzita, Brno, 2003.
- [5] R. Šimon Hilscher a P. Zemánek, *Friedrichs extension of operators defined by linear Hamiltonian systems on unbounded interval*, v „Equadiff 12“, Proceedings of the Conference on Differential Equations and their Applications (Brno, 2009), J. Diblík, O. Došlý, P. Drábek a E. Feistauer, editoři, Math. Bohem. **135** (2010), č. 2, 209–222.
- [6] W. T. Reid, *Sturmian Theory for Ordinary Differential Equations*, Applied Mathematical Sciences, Springer-Verlag, New York, 1980. ISBN 0-387-90542-1.
- [7] W. T. Reid, *Sturmian Theory for Ordinary Differential Equations*, Applied Mathematical Sciences, Springer-Verlag, New York, 1980. ISBN 0-387-90542-1.
- [8] W. T. Reid, *Sturmian Theory for Ordinary Differential Equations*, Applied Mathematical Sciences, Springer-Verlag, New York, 1980. ISBN 0-387-90542-1.
- [9] W. T. Reid, *Sturmian Theory for Ordinary Differential Equations*, Applied Mathematical Sciences, Springer-Verlag, New York, 1980. ISBN 0-387-90542-1.
- [10] W. T. Reid, *Sturmian Theory for Ordinary Differential Equations*, Applied Mathematical Sciences, Springer-Verlag, New York, 1980. ISBN 0-387-90542-1.
- [11] W. T. Reid, *Sturmian Theory for Ordinary Differential Equations*, Applied Mathematical Sciences, Springer-Verlag, New York, 1980. ISBN 0-387-90542-1.
- [12] W. T. Reid, *Sturmian Theory for Ordinary Differential Equations*, Applied Mathematical Sciences, Springer-Verlag, New York, 1980. ISBN 0-387-90542-1.

- [13] W. T. Reid, *Sturmian Theory for Ordinary Differential Equations*, Applied Mathematical Sciences, Springer-Verlag, New York, 1980. ISBN 0-387-90542-1.
- [14] W. T. Reid, *Sturmian Theory for Ordinary Differential Equations*, Applied Mathematical Sciences, Springer-Verlag, New York, 1980. ISBN 0-387-90542-1.

