

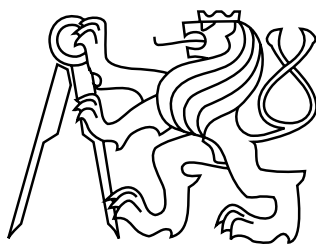
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
FAKULTA STAVEBNÍ

DIZERTAČNÁ PRÁCA

PRAHA 2018

Ing. Gabriel GYÖRI

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
FAKULTA STAVEBNÍ  
OBOR GEODÉZIE A KARTOGRAFIE



DIZERTAČNÁ PRÁCA

LMS V GEODETICKÝCH APLIKÁCIACH  
VYROVNÁVACIEHO POČTU

Školiteľ: Prof. Ing. Aleš Čepek, CSc.  
Školiteľ špecialista: Doc. RNDr. Petr Mayer, Dr.  
Katedra geomatiky

## **ABSTRAKT**

Témou dizertačnej práce je implementácia metódy mediánov najmenších štvorcov (LMS – Least Median of Squares) ako robustnej metódy vyrovnania geodetických meraní. Metóda LMS je založená na výbere vzoriek pozorovaní a následnom riešení lineárnej kombinácie neznámych parametrov v závislosti na meraných veličinách bez nutnosti explicitného zostavovania normálnych rovníc. Metóda nieje v každodennej praxi používaná kvôli jej náročnosti na výpočet a neexistujúcemu riešeniu na uzavretom intervale. Riešenie neznámych parametrov je vykonané pomocou algoritmu združených gradientov. Aj napriek tomu, že spomínaný algoritmus je z hľadiska numerického výpočtu optimálny, v geodézii sa veľmi nevyužíva z dôvodu, že pre výpočet neznámych parametrov neurčuje inverznú maticu, ktorá obsahuje v geodézii neodmysliteľné presnosti vyrovnávaných veličín, ktoré však pre konkrétne riešenie LMS nie sú potrebné. V riešenej implementácii algoritmus združených gradientov využíva výhodu riedkosti vstupnej matice plánu. Dátové štruktúry sú kompatibilné s dátovými štruktúrami projektu GNU Gama pre prípadnú implementáciu do projektu.

## **KĹÚČOVÉ SLOVA**

LMS, robustné metódy, metóda združených gradientov, riedke matice, vyrovnávací počet

## **ABSTRACT**

The topic of the doctoral thesis is implementation of the least median of squares (LMS) as robust method used in geodesy adjustment. This method is based on choosing samples of observations which are used for solving linear combination of unknown parameters. LMS method is not usually used in geodesy because of its difficult computation process and also exact solution in finite range is not known. Solving of unknown parameters is provided by method of conjugate gradients. Although it is optimal algorithm for solving of linear algebraic equations the conjugate gradients method is not used in geodesy because there is any possibility to reach elements of inverse matrix in effective way. Elements of inverse matrix are used for computing of the standard deviations which are necessary to evaluate adjustment results. LMS method does not require inversion of design matrix itself for obtaining of correct solution. Effective solution can be reached by using combination of the conjugate gradient method and sparse matrices. It could lead to less time and store consuming computation. Data structures of algorithms are compatible with algorithms of GNU Gama project.

## **KEYWORDS**

LMS, robust methods, conjugate gradient method, sparse matrices, adjustment theory

## PREHLÁSENIE

Prehlasujem že dizertačnú prácu na tému „LMS v geodetických aplikáciach vyrovnávacieho počtu“ som vypracoval samostatne. Použitú literatúru a podkladové materiály uvádzam v zozname zdrojov.

V Prahe dňa .....

.....

(podpis autora)

## POĎAKOVANIE

Na tomto mieste by som chcel v prvom rade poďakovať môjmu školiteľovi prof. Ing. Alešovi Čepkovi, CSc. za cenné rady, pripomienky a odborné konzultácie v oblasti geodézie a vyrovnávacieho počtu a za vždy pozitívny a trpezlivý prístup. Vedenie prof. Čepka bolo vždy na špičkovej profesionálnej úrovni ako po stránke odbornej, tak aj ľudskej a bez jeho vedenia si nedokážem predstaviť svoje doktorské štúdium a odborné pôsobenie. Vďaka taktiež patrí školiteľovi špecialistovi doc. RNDr. Petrovi Mayerovi, Dr. za čas venovaný konzultáciám v obore numerickej matematiky a programovania, ktoré presahujú obor geodézie a geoinformatiky. Získané znalosti sú pre mňa veľmi vzácne a nevyhnutné nielen pre vypracovanie tejto dizertačnej práce, ale mnohokrát našli využitie aj v mojom odbornom živote a v nemalej miere ich využívam dodnes. V tomto odstavci nesmiem zabudnúť na Ing. Jana Doušu, Ph.D., vedúceho projektu grantovej akadémie ČR, na ktorom som s ním pôsobil vo Výskumnom ústave geodetickom, topografickom a kartografickom a s ktorým som spoluautor niekoľkých publikácií ohľadom GPS a určovania troposféry. Vďaka Ing. Janovi Doušovi, Ph.D. som mal možnosť pracovať na veľmi zaujímavých medzinárodných projektoch, ktoré v nemalej miere prispeli k môjmu odbornému rastu. Taktiež by som rád poďakoval svojim spolužiakom z katedry špeciálnej geodézie za poskytnutie terestrických meraní geodetických sietí určených v praxi, ktoré boli následne objektom skúmania popísaného v tejto práci. V neposlednej rade musím poďakovať svojej manželke, rodičom a blízkym za neustálu podporu pri štúdiu. Dizertačná práca je jedným z výsledkov niekoľkoročného pôsobenia v oblasti aplikovaného výskumu, behom ktorého som mal možnosť poznať a pracovať s mnohými profesionálmi vo svojom obore, za čo som veľmi vďačný.

# Obsah

|   |           |
|---|-----------|
| Úvod  | 17        |
| <b>1 Ciele</b>  | <b>19</b> |
| <b>2 Stav riešenej problematiky</b>   | <b>20</b> |
| 2.1 Popis robustných metod . . . . .  | 20        |
| 2.2 Voľba numerických metod . . . . .   | 21        |
| 2.3 Voľba knižnice riedkych matíc . . . . .   | 23        |
| <b>3 Metóda minimalizácie mediánov štvorcov reziduí</b>                               | <b>24</b> |
| 3.1 Algoritmy pre výpočet LMS . . . . .   | 26        |
| 3.1.1 Algoritmy pre výpočet odhadu LMS v rovine . . . . .                             | 27        |
| 3.1.2 Algoritmy pre exaktný výpočet odhadu LMS v<br>d-rozmernom priestore . . . . .   | 31        |
| 3.1.3 Algoritmy pre približný výpočet odhadu LMS v<br>d-rozmernom priestore . . . . . | 32        |
| 3.1.4 Paralelne algoritmy pre výpočet odhadu LMS . . . . .                            | 38        |
| 3.2 Prípady zlyhania LMS odhadu . . . . .   | 40        |
| <b>4 Robustné metódy vyrovnania meraných veličín</b>                                  | <b>42</b> |
| 4.1 Teoretický základ . . . . .   | 42        |
| 4.2 $L_p$ -norma odhady . . . . .   | 42        |
| 4.3 M-odhady . . . . .  | 43        |
| 4.3.1 Určenie váhových koeficientov . . . . .   | 45        |
| 4.4 Least Trimmed Squares - LTS . . . . .   | 47        |
| 4.5 Metóda RANSAC . . . . .   | 48        |
| 4.6 Odľahlé body a ich vplyv na odhad neznámych parametrov . . . . .                  | 49        |
| 4.6.1 Identifikácia vplyvových bodov . . . . .  | 50        |
| <b>5 Určenie spoľahlivosti vyrovnania<br/>geodetických meraní metódou LMS</b>         | <b>53</b> |
| 5.1 Vzťah medzi reziduami a hrubými chybami . . . . .                                 | 53        |
| 5.2 Globálny test smerodajnej odchýlky . . . . .                                      | 54        |
| 5.3 Data snooping . . . . .   | 56        |
| 5.4 Tau-test . . . . .  | 58        |
| 5.5 Test kvadratickej formy reziduí . . . . .   | 58        |

|          |  |           |
|----------|--|-----------|
| <b>6</b> | <b>Implementačné aspekty numerických operácii riedkych matíc</b>         | <b>61</b> |
| 6.1      | Graf riedkych matíc . . . . .  | 61        |
| 6.2      | Štruktúra riedkych matíc . . . . .                                       | 61        |
| 6.2.1    | Dynamicky re-alokovaná štruktúra riedkej matice . . . . .                | 63        |
| 6.3      | Násobenie riedkej matice a vektora . . . . .                             | 66        |
| 6.4      | Implementácia násobenia riedkej matice a vektora . . . . .               | 67        |
| 6.5      | Implementácia sčítania matíc . . . . .                                   | 68        |
| 6.6      | Násobenie riedkych matíc . . . . .                                       | 70        |
| 6.7      | Implementácia vonkajších súčinov matíc . . . . .                         | 70        |
| <b>7</b> | <b>Numerické metódy riešenia sústavy lineárnych rovníc</b>               | <b>72</b> |
| 7.1      | Gramm-Schmidtova ortogonalizácia . . . . .                               | 72        |
| 7.1.1    | Zovšeobecnená ortogonalizácia . . . . .                                  | 73        |
| 7.2      | Singulárny rozklad SVD . . . . .   | 74        |
| 7.3      | Metóda združených gradientov . . . . .                                   | 75        |
| 7.3.1    | Zastavenie výpočtu . . . . .   | 77        |
| 7.3.2    | Metóda predpodmienených združených gradientov . . . . .                  | 78        |
| 7.3.3    | Neúplný Choleskýho rozklad s riedkou maticou . . . . .                   | 79        |
| 7.3.4    | Riešenie lineárnych rovníc trojuholníkových systémov . . . . .           | 80        |
| <b>8</b> | <b>Implementačné detaily LMS a vybraných robustných metód.</b>           | <b>82</b> |
| 8.1      | Generovanie vstupných dát . . . . .                                      | 82        |
| 8.1.1    | Vygenerovanie vstupných parametrov sústavy lineárnych rovníc             | 82        |
| 8.1.2    | Vygenerovanie a rozdelenie porúch meraní . . . . .                       | 83        |
| 8.1.3    | Implementácia generátoru v C++11 . . . . .                               | 83        |
| 8.2      | Exaktný výpočet metódy LMS . . . . .                                     | 84        |
| 8.3      | Implementácia vybraných M-odhadov . . . . .                              | 84        |
| 8.4      | Implementácia metódy LTS . . . . .                                       | 86        |
| 8.5      | Implementácia metódy RANSAC . . . . .                                    | 87        |
| 8.6      | Výpočet LMS využitím viacjadrových procesorov . . . . .                  | 89        |
| 8.6.1    | Výpočet skalárnych súčinov s využitím multi-thread technológie . . . . . | 89        |
| 8.6.2    | Výpočet bez použitia exkluzívnych zámkov . . . . .                       | 91        |
| 8.6.3    | Paralelizácia určenia minimálneho mediánu štvorcov reziduí . . . . .     | 92        |

|           |   |            |
|-----------|---|------------|
| <b>9</b>  | <b>Porovnanie výsledkov vyrovnania simulovaných dát určených metódou LMS a vybranými robustnými metódami.</b> | <b>94</b>  |
| 9.1       | Porovnanie výsledkov vyrovnania simulovaných dát metódou LMS s M-odhadmi . . . . .                            | 94         |
| 9.2       | Porovnanie výsledkov vyrovnania simulovaných dát metódou LMS s metódou LTS . . . . .                          | 108        |
| 9.3       | Porovnanie výsledkov vyrovnania simulovaných dát metódou LMS s metódou RANSAC . . . . .                       | 113        |
| 9.4       | Zhodnotenie efektivity návrhu paralelného výpočtu LMS . . . . .   | 118        |
| 9.5       | Dielčí záver . . . . .  | 120        |
| <b>10</b> | <b>Použitie metód v geodetických aplikáciach</b>  | <b>121</b> |
| 10.1      | Vyrovnanie geodetickej siete . . . . .  | 121        |
| 10.1.1    | Vyrovnanie s podmienkami . . . . .  | 124        |
| 10.2      | Zostavenie normálnych rovníc . . . . .  | 128        |
| 10.3      | Riedke matice v geodézii . . . . .  | 129        |
| 10.4      | Riedke matice pri určovaní presnej polohy bodu pomocou GPS . . . . .  | 129        |
| 10.4.1    | Rovnice pozorovaní . . . . .  | 130        |
| 10.4.2    | Sekvenčné vyrovnanie . . . . .  | 131        |
| 10.4.3    | Analýza matice plánu GPS meraní . . . . .   | 131        |
| 10.4.4    | Choleského rozklad a minimum degree permutácia . . . . .  | 133        |
| 10.4.5    | Výsledky observácii stanice NICO . . . . .  | 133        |
| <b>11</b> | <b>Určovanie odľahlých geodetických meraní metódou LMS</b>  | <b>135</b> |
| 11.1      | Postup určenia množiny meraní podozrivých z odľahlosti . . . . .  | 135        |
| 11.2      | Aplikovanie postupu na terestrických geodetických meraniach . . . . .   | 137        |
| 11.3      | Zhodnotenie výsledkov . . . . .   | 138        |
| <b>12</b> | <b>Diskrétne rozdelenie pravdepodobnosti generovania náhodného vzoru geodetických meraní</b>                  | <b>142</b> |
| 12.1      | Rozdelenie pravdepodobnosti diskkrétnej náhodnej veličiny . . . . .   | 143        |
| 12.2      | Algoritmus diskkrétneho rozdelenia pravdepodobností v závislosti na veľkosti vyrovnaných opráv . . . . .      | 144        |
| 12.3      | Stratégie rozdelenia pravdepodobnosti . . . . .   | 144        |
| 12.4      | Implementácia diskkrétneho rozdelenia pravdepodobnosti v C++11 . . . . .                                      | 146        |
| 12.5      | Možnosti paralelizácie výpočtu . . . . .  | 148        |
| 12.6      | Zhodnotenie výsledkov . . . . .   | 148        |



|  |            |
|--|------------|
| <b>13 Výpočet geodetickej siete metódou LMS</b>                        |            |
| <b>použitím metódy zmeny váh meraní</b>                                | <b>151</b> |
| 13.1 Homogenizácia meraní . . . . .                                    | 151        |
| 13.2 Stratégia výpočtu LMS . . . . .                                   | 152        |
| 13.2.1 Vyhodenie vzoru matice plánu . . . . .                          | 152        |
| 13.2.2 Stratégie váhovania . . . . .                                   | 153        |
| 13.2.3 Generovanie náhodného vzoru matice plánu . . . . .              | 154        |
| 13.2.4 Algoritmus výpočtu . . . . .                                    | 154        |
| 13.3 Určovanie súvislosti geodetickej siete . . . . .                  | 155        |
| 13.3.1 Implementácia určenia súvislosti geodetickej siete. . . . .     | 156        |
| 13.4 Perzistentné uloženie výsledkov do databáze . . . . .             | 159        |
| 13.4.1 Databáza PostgreSQL . . . . .                                   | 159        |
| 13.4.2 Inštalácia a konfigurácia databáze na platforme Linux . . . . . | 159        |
| 13.4.3 Založenie databáze a užívateľa . . . . .                        | 161        |
| 13.4.4 Návrh databázových objektov . . . . .                           | 161        |
| 13.5 Implementácia komunikácie aplikácie LMS s databázou . . . . .     | 163        |
| 13.5.1 Naviazanie konexie . . . . .                                    | 164        |
| 13.5.2 Vytvorenie a odstránenie objektov databáze . . . . .            | 165        |
| 13.5.3 Zápis hodnôt do databázových objektov . . . . .                 | 165        |
| 13.6 Použitie aplikácie LMS . . . . .                                  | 167        |
| 13.6.1 Konfiguračné súbory aplikácie LMS . . . . .                     | 167        |
| 13.6.2 Inštalácia a spustenie aplikácie . . . . .                      | 168        |
| 13.7 Technológia hromadného spracovania dát . . . . .                  | 169        |
| 13.7.1 Vyrovnávanie sietí . . . . .                                    | 170        |
| 13.7.2 Separácia vstupných dát . . . . .                               | 170        |
| 13.7.3 Zostavenie konfiguračných súborov . . . . .                     | 171        |
| 13.7.4 Vytvorenie kovariančnej matice meraní . . . . .                 | 172        |
| 13.7.5 Výpočet LMS . . . . .   | 173        |
| 13.8 Posúdenie kvality algoritmov . . . . .                            | 174        |
| 13.8.1 Overenie metódy združených gradientov . . . . .                 | 174        |
| 13.8.2 Overenie algoritmov LMS . . . . .                               | 174        |
| 13.9 Zhodnotenie výsledkov vyrovnania geodetických sietí . . . . .     | 182        |
| 13.9.1 Vyrovnávané geodetické siete . . . . .                          | 182        |
| 13.9.2 Overenie metódy LMS na vybraných geodetických sieťach . . . . . | 183        |
| <b>Záver</b>   | <b>194</b> |
| Literatura . . . . .   | 198        |
| <b>Referencie</b>  | <b>198</b> |

|  |            |
|--|------------|
| <b>Zoznam skratiek</b>   | <b>204</b> |
| <b>Zoznam príloh</b>   | <b>206</b> |
| <b>A Zápisky geodetických meraní</b>                             | <b>207</b> |
| <b>B Vybrané zdrojové kódy</b>                                   | <b>209</b> |
| B.1 Algoritmus združených gradientov s predpodmienením . . . . . | 209        |
| B.2 Algoritmus ILU0 . . . . .                                    | 213        |
| B.3 Funkcie súborových operácií . . . . .                        | 215        |

# Zoznam obrázkov

|      |   |     |
|------|---|-----|
| 3.1  | Zmena odhadu spôsobená malým rozdielom v centre dát . . . . .   | 41  |
| 4.1  | Znázornenie odľahlého bodu v ose Y [Outliers-53] . . . . .  | 50  |
| 4.2  | Znázornenie odľahlého bodu v ose X [Outliers-53] . . . . .  | 51  |
| 4.3  | Bod ovplyvňujúci určený odhad [Outliers-53] . . . . .   | 52  |
| 6.1  | Graf riedkej matice . . . . .   | 62  |
| 6.2  | Štruktúra riedkej matice . . . . .  | 62  |
| 6.3  | Štruktúra dynamicky re-alokovanej riedkej matice A . . . . .  | 64  |
| 6.4  | Realokácia pamäti . . . . .   | 64  |
| 6.5  | Funkcia garbage collector . . . . .   | 65  |
| 6.6  | Vloženie prvku do štruktúry riedkej matice . . . . .  | 67  |
| 6.7  | Sčítanie riedkych matíc . . . . .   | 69  |
| 7.1  | Poloha vlastných čísel . . . . .  | 78  |
| 9.1  | Odchýlky výpočtu neznámych parametrov od očakávaných hodnôt<br><i>Príklad 1</i> . . . . .             | 99  |
| 9.2  | Graf hodnôt vyrovnaných reziduí <i>Príklad 1</i> . . . . .  | 100 |
| 9.3  | Odchýlky výpočtu neznámych parametrov od očakávaných hodnôt<br><i>Príklad 2</i> . . . . .             | 100 |
| 9.4  | Graf hodnôt vyrovnaných reziduí <i>Príklad 2</i> . . . . .  | 101 |
| 9.5  | Odchýlky výpočtu neznámych parametrov od očakávaných hodnôt<br><i>Príklad 3</i> . . . . .             | 101 |
| 9.6  | Graf hodnôt vyrovnaných reziduí <i>Príklad 3</i> . . . . .  | 102 |
| 9.7  | Závislosť hodnoty mediánu na početnosti meraných hodnôt vzoru sú-<br>stavy <i>Príklad 1</i> . . . . . | 103 |
| 9.8  | Závislosť hodnoty mediánu na početnosti meraných hodnôt vzoru sú-<br>stavy <i>Príklad 2</i> . . . . . | 103 |
| 9.9  | Odchýlky výpočtu neznámych parametrov od očakávaných hodnôt<br><i>Príklad 4</i> . . . . .             | 104 |
| 9.10 | Graf hodnôt vyrovnaných reziduí <i>Príklad 2</i> . . . . .  | 104 |
| 9.11 | Odchýlky výpočtu neznámych parametrov od očakávaných hodnôt<br><i>Príklad 1</i> . . . . .             | 111 |
| 9.12 | Graf hodnôt vyrovnaných reziduí <i>Príklad 1</i> . . . . .  | 111 |
| 9.13 | Odchýlky výpočtu neznámych parametrov od očakávaných hodnôt<br><i>Príklad 2</i> . . . . .             | 112 |
| 9.14 | Graf hodnôt vyrovnaných reziduí <i>Príklad 2</i> . . . . .  | 112 |
| 9.15 | Odchýlky výpočtu neznámych parametrov od očakávaných hodnôt<br><i>Príklad 1</i> . . . . .             | 116 |
| 9.16 | Graf hodnôt vyrovnaných reziduí <i>Príklad 1</i> . . . . .  | 117 |

|       |  |     |
|-------|--|-----|
| 9.17  | Odchýlky výpočtu neznámych parametrov od očakávaných hodnôt<br><i>Príklad 2</i> . . . . .                              | 117 |
| 9.18  | Graf hodnôt vyrovnaných reziduí <i>Príklad 2</i> . . . . .   | 117 |
| 9.19  | Doba výpočtu v závislosti na počte použitých vlákien a počte kombinácií . . . . .                                      | 119 |
| 10.1  | Matica plánu . . . . .   | 132 |
| 10.2  | Matice pre 10 min pozorovania . . . . .  | 134 |
| 10.3  | Matice Choleského rozkladu . . . . .   | 134 |
| 11.1  | Súradnice bodov geodetických meraní . . . . .  | 137 |
| 11.2  | Hodnoty vyrovnaných opráv v závislosti na počte kontaminovaných meraní . . . . .                                       | 140 |
| 11.3  | Histogram početnosti meraní rozpoznaných metódou LMS - 2 hrubé chyby . . . . .   | 140 |
| 11.4  | Histogram početnosti meraní rozpoznaných metódou LMS - 8 hrubých chýb . . . . .  | 140 |
| 11.5  | Histogram početnosti meraní rozpoznaných metódou LMS - 28 hrubých chýb . . . . .                                       | 141 |
| 12.1  | Rozdelenie pravdepodobnosti meraniam . . . . .   | 146 |
| 12.2  | Priebeh zmeny mediánu v závislosti na vybranej stratégii diskrétného rozdelenia pravdepodobnosti - Príklad 1 . . . . . | 150 |
| 12.3  | Priebeh zmeny mediánu v závislosti na vybranej stratégii diskrétného rozdelenia pravdepodobnosti - Príklad 2 . . . . . | 150 |
| 13.1  | Meranie v geodetickej sieti [Gama–29] . . . . .  | 151 |
| 13.2  | Súvislosť siete [Connectivity–54] . . . . .  | 157 |
| 13.3  | Databázové schéma . . . . .  | 163 |
| 13.4  | Automatizácia výpočtu sietí . . . . .  | 169 |
| 13.5  | Graf matice 1033x320 vľavo a 1850x712 vpravo . . . . .   | 176 |
| 13.6  | Graf znázornenia konvergencie mediánu pre jeden náhodný vzor . . . . .   | 177 |
| 13.7  | Graf matice 1033x320 vľavo a 1850x712 vpravo . . . . .   | 178 |
| 13.8  | Graf matice 1033x320 vľavo a 1850x712 vpravo . . . . .   | 181 |
| 13.9  | Graf matice 1033x320 vľavo a 1850x712 vpravo . . . . .   | 181 |
| 13.10 | Grafy riedkych matíc geodetických sietí . . . . .  | 189 |
| 13.11 | Početnosť odstránení meraní podozrivých z odľahlosti mostnej konštrukcii v obci Klášterec nad Ohří . . . . .           | 190 |
| 13.12 | Početnosť odstránení meraní podozrivých z odľahlosti mostnej konštrukcii v obci Saný . . . . .                         | 190 |
| 13.13 | Hodnoty priemerného mediánu LMS mostnej konštrukcii v obci Klášterec nad Ohří pre 10 vylúčených meraní . . . . .       | 190 |

|       |   |     |
|-------|---|-----|
| 13.14 | Hodnoty minimalného mediánu metódy LMS mostnej konštrukcii v obci Klášterec nad Ohří pre 10 vylúčených meraní . . . . .           | 191 |
| 13.15 | Hodnoty priemerného mediánu metódy LMS mostnej konštrukcii v obci Saný pre 10 vylúčených meraní . . . . .                         | 191 |
| 13.16 | Hodnoty minimalného mediánu metódy LMS mostnej konštrukcii v obci Saný pre 10 vylúčených meraní . . . . .                         | 192 |
| 13.17 | Hodnoty minimalného a priemerného mediánu metódy LMS mostnej konštrukcii v obci Klášterec nad Ohří pre 40 vylúčených meraní . . . | 192 |
| 13.18 | Priebeh poklesu mediánu metódy LMS aplikovanej pri výpočte mostnej konštrukcii v obci Klášterec nad Ohří . . . . .                | 193 |

# Zoznam tabuliek

|      |   |     |
|------|---|-----|
| 8.1  | Voľba konštánt vybraných robustných odhadov . . . . .   | 85  |
| 9.1  | Tabuľka parametrov overovaných sústav . . . . .   | 96  |
| 9.2  | Tabuľka rozdelenia chyb medzi observácie . . . . .  | 96  |
| 9.3  | Tabuľka rozdelenia chyb medzi observácie . . . . .  | 97  |
| 9.4  | Tabuľka indexov kombinácie observácii použitej pre výpočet nezná-<br>mých parametrov . . . . .                  | 97  |
| 9.5  | Tabuľka minimálnych mediánov . . . . .  | 98  |
| 9.6  | Tabuľka odchýlok vyrovnaných neznámych parametrov od očakáva-<br>ných hodnôt 1.časť. <i>Príklad 1</i> . . . . . | 98  |
| 9.7  | Tabuľka odchýlok vyrovnaných neznámych parametrov od očakáva-<br>ných hodnôt 2.časť. <i>Príklad 1</i> . . . . . | 99  |
| 9.8  | Tabuľka odchýlok vyrovnaných neznámych parametrov od očakáva-<br>ných hodnôt <i>Príklad 2</i> . . . . .         | 105 |
| 9.9  | Tabuľka odchýlok vyrovnaných neznámych parametrov od očakáva-<br>ných hodnôt <i>Príklad 3</i> . . . . .         | 106 |
| 9.10 | Tabuľka odchýlok vyrovnaných neznámych parametrov od očakáva-<br>ných hodnôt <i>Príklad 4</i> . . . . .         | 107 |
| 9.11 | Tabuľka parametrov sústav porovnania metód LMS a LTS . . . . .  | 108 |
| 9.12 | Tabuľka rozdelenia chyb medzi observácie . . . . .  | 109 |
| 9.13 | Tabuľka indexov kombinácie observácii použitej pre výpočet nezná-<br>mých parametrov . . . . .                  | 109 |
| 9.14 | Tabuľka minimálnych mediánov a minimálnych sum štvorcov . . . . .   | 110 |
| 9.15 | Tabuľka odchýlok vyrovnaných neznámych parametrov od očakáva-<br>ných hodnôt 1.časť. <i>Príklad 1</i> . . . . . | 110 |
| 9.16 | Tabuľka odchýlok vyrovnaných neznámych parametrov od očakáva-<br>ných hodnôt 2.časť. <i>Príklad 1</i> . . . . . | 110 |
| 9.17 | Tabuľka odchýlok vyrovnaných neznámych parametrov od očakáva-<br>ných hodnôt <i>Príklad 2</i> . . . . .         | 110 |
| 9.18 | Tabuľka parametrov sústav porovnania metód LMS a RANSAC . . . . .   | 113 |
| 9.19 | Tabuľka rozdelenia chyb medzi observácie . . . . .  | 114 |
| 9.20 | Tabuľka indexov kombinácie observácii použitej pre výpočet nezná-<br>mých parametrov . . . . .                  | 115 |
| 9.21 | Tabuľka minimálnych mediánov a minimálnych sum štvorcov . . . . .   | 115 |
| 9.22 | Tabuľka odchýlok vyrovnaných neznámych parametrov od očakáva-<br>ných hodnôt <i>Príklad 1</i> . . . . .         | 115 |
| 9.23 | Tabuľka odchýlok vyrovnaných neznámych parametrov od očakáva-<br>ných hodnôt <i>Príklad 1</i> . . . . .         | 115 |

|      |  |     |
|------|--|-----|
| 9.24 | Tabuľka odchýlok vyrovnaných neznámych parametrov od očakávaných hodnôt <i>Príklad 2</i> . . . . .                 | 116 |
| 9.25 | Tabuľka odchýlok vyrovnaných neznámych parametrov od očakávaných hodnôt <i>Príklad 2</i> . . . . .                 | 116 |
| 9.26 | Tabuľka doby výpočtu LMS . . . . .   | 119 |
| 11.1 | Rozdelenie hrubých chýb . . . . .  | 138 |
| 11.2 | Určenie prírastkov vyrovnávaných súradníc v závislosti na použitej metóde a počte kontaminovaných meraní . . . . . | 139 |
| 11.3 | Test kvadratickej formy . . . . .  | 139 |
| 12.1 | Tabuľka parametrov generovaných sústav . . . . .   | 149 |
| 12.2 | Tabuľka rozmerov určených sústav . . . . .   | 149 |
| 13.1 | Porovnanie rýchlosti rôznych typov algoritmov združených gradientov  | 175 |
| 13.2 | Závislosť hodnoty minimálneho a mediánu na vybranej stratégii váhovania. Matica 1033x320 . . . . .                 | 178 |
| 13.3 | Závislosť hodnoty minimálneho a mediánu na vybranej stratégii váhovania. Matica 1850x712 . . . . .                 | 179 |
| 13.4 | Závislosť hodnoty minimalného mediánu na veľkosti množiny preváhaných riadkov . . . . .                            | 180 |
| 13.5 | Vyrovňované siete . . . . .  | 183 |
| 13.6 | Tabuľka spôsobu rozdelenia porúch meraniam . . . . .   | 184 |
| 13.7 | Tabuľka riadkov s odľahlými meraniami . . . . .  | 186 |
| 13.8 | Tabuľka konfigurácií metódy LMS . . . . .  | 187 |

# Zoznam zdrojových kódov

|       |   |     |
|-------|---|-----|
| 3.1   | Symbolicky zápis algoritmu určenia odhadu LMS minimalizáciou funkcie iteratívnym spresňovaním . . . . . | 37  |
| 3.2   | Roussew LMS algoritmus určený pre paralelizáciu . . . . .   | 38  |
| 6.1   | Algoritmus násobenia riedkej matice a vektoru . . . . .   | 67  |
| 6.2   | Algoritmus násobenia riedkej transponovanej matice a vektoru . . . . .                                  | 68  |
| 6.3   | Algoritmus násobenia riedkej matice a vektoru . . . . .   | 69  |
| 6.4   | Algoritmus vonkajšieho súčinu dvoch vektorov . . . . .  | 70  |
| 7.1   | Algoritmus stĺpcovej verzie priamej substitúcie . . . . .   | 80  |
| 7.2   | Algoritmus spätnej substitúcie . . . . .  | 81  |
| 8.1   | Generátor náhodných čísel . . . . .   | 83  |
| 8.2   | Definícia riešiča robustných odhadov . . . . .  | 85  |
| 8.3   | Volanie funkcií robustných odhadov . . . . .  | 86  |
| 8.4   | Hlavný výpočet LTS . . . . .  | 87  |
| 8.5   | Hlavný výpočet RANSAC . . . . .   | 87  |
| 8.6   | Implementácia skalarného súčinu bez využitia vlákien . . . . .  | 89  |
| 8.7   | Funkcia pre výpočet skalarného súčinu dvoch vektorov pomocou viacerých jadier . . . . .                 | 90  |
| 8.8   | Funkcia pre určenie hraníc rozdeleného vektoru . . . . .  | 90  |
| 8.9   | Vykonanie úlohy sklarného súčinu viacerými jadrami . . . . .  | 91  |
| 8.10  | Výpočet bez použitia exkluzívnych zámkov . . . . .  | 91  |
| 8.11  | Výpočet mediánu . . . . .   | 92  |
| 8.12  | Výpočet LMS n-vláknami . . . . .  | 93  |
| 8.13  | Výpis minimálnych mediánov pre konkrétne submatice . . . . .  | 93  |
| 9.1   | Určenie času výpočtu danej úlohy . . . . .  | 118 |
| 12.1  | Algoritmus generovania diskretných náhodných veličín . . . . .  | 146 |
| 12.2  | Funktor linearnarného diskretného rozdelenia . . . . .  | 147 |
| 12.3  | Volanie funkcie solve() vo funkcii main() . . . . .   | 147 |
| 13.1  | Algoritmus prehľadávania do hĺbky . . . . .   | 157 |
| 13.2  | Určenie počtu komponent slabej súvislosti . . . . .   | 158 |
| 13.3  | Metoda vytvorenia databázových schém . . . . .  | 158 |
| 13.4  | Inštalácia databáze . . . . .   | 159 |
| 13.5  | Inicializácia databázového clusteru . . . . .   | 160 |
| 13.6  | Konfigurácia rozhraní . . . . .   | 160 |
| 13.7  | Povolenie prístupu . . . . .  | 160 |
| 13.8  | Spustenie databázového serveru . . . . .  | 160 |
| 13.9  | Založenie databáze a užívateľa . . . . .  | 161 |
| 13.10 | Metoda naviazania spojenia s databázovým serverom . . . . .   | 164 |



|       |  |     |
|-------|--|-----|
| 13.11 | Metoda vytvorenia databázových schém . . . . .               | 165 |
| 13.12 | Metoda zápisu do relácia settings . . . . .                  | 166 |
| 13.13 | Hromadné vyrovnanie sietí . . . . .                          | 168 |
| 13.14 | Hromadné vyrovnanie sietí . . . . .                          | 168 |
| 13.15 | Hromadné vyrovnanie sietí . . . . .                          | 170 |
| 13.16 | Separácia vstupných dát . . . . .                            | 171 |
| 13.17 | Vygenerovanie nastavení . . . . .                            | 171 |
| 13.18 | Vytvorenie kovariančnej matice meraní . . . . .              | 173 |
| 13.19 | Vyrovnanie dát aplikáciou LMS . . . . .                      | 173 |
| A.1   | Merané hodnoty . . . . .                                     | 207 |
| A.2   | Súradnice bodov . . . . .                                    | 208 |
| B.1   | Algoritmus združených gradientov s predpodmienením . . . . . | 209 |
| B.2   | Algoritmus združených gradientov s predpodmienením . . . . . | 213 |
| B.3   | Funkcie súborových oprácií . . . . .                         | 215 |

# Úvod

Dizertačná práca spojuje oblasť numerickej matematiky, informatiky a štatistiky aplikovanej v obore geodézie, preto je v tomto zmysle interdisciplinárna. Vlastný prínos dizertačnej práce spočíva v podobe návrhu, implementácie a zhodnotenia získaných výsledkov použitím klasických metód numerickej matematiky a robustných metód používaných v štatistike, aplikovaných v oblasti geodézie a oboroch jej príbuzných. Pôvodnosť práce spočíva v návrhu stratégií a optimalizácie numerických výpočtov, ktoré v kombinácii so vstupnými predpokladmi príznačnými pre geodetické úlohy môžu dosahovať vysokú efektivitu výpočtu a presnosť hľadaných neznámych parametrov. Súčasťou je štúdia venovaná popisu súčasného stavu riešenej problematiky, porovnaniu rôznych algoritmov z hľadiska efektivity a presnosti riešenia sústavy lineárnych rovníc a implementácii algoritmov pre výpočet mediánu.

V úvode je práca zameraná na hlbšie teoretické základy vybraných numerických a štatistických metód, možnosť ich použitia v geodetických aplikáciách, spôsoby ich implementácie a návrh predpokladaného výpočtu. Jedným z hlavných subjektov výskumu je opakované riešenie sústavy lineárnych algebraických rovníc používaných v geodézii metódou minimalizácie mediánu štvorcov opráv meraných veličín (LMS). Matica, ktorá je súčasťou lineárnych rovníc sa v geodézii nazýva prvá matica plánu a má v mnohých prípadoch riedku štruktúru. Práve kvôli tejto vlastnosti je dôraz kladený na to, aby vo vybranom algoritme bola možnosť použitia výpočtu s riedkymi maticami. Správna implementácia riedkych matíc môže v mnohých prípadoch značne zrýchliť čas výpočtu a zredukovať veľkosť alokovanej pamäti výpočtovej jednotky. Ďalšou súčasťou práce je teoretický popis robustných metód vyrovnania meraných veličín v porovnaní s klasickými metódami, ktoré sú v geodézii bežne používané.

Metóda minimalizácie mediánu štvorcov reziduí patrí medzi NP-náročné úlohy, z čoho vyplýva, že nemá riešenie na uzavretom intervale a to vo väčšine prípadoch, avšak s výnimkou situácií, kde je možné vytvoriť všetky kombinácie definovanej množiny meraní v konečnom čase. Pokiaľ opomenieme prípad exaktného určenia odhadu neznámych parametrov, tak po ukončení každého výpočtu LMS je získaný ich jedinečný odhad. Základom čo najlepšieho odhadu by mohla byť vhodná voľba stratégie postupného váhovania odľahlých meraní. Metóda LMS by mala byť primárne určená na detekciu skupiny odľahlých meraní, ktoré vzhľadom k množstvu meraní nieje možné v momente akvizície rozpoznať. Typickým prípadom sú dáta GPS meraní, laserového skenovania či obrazových dát diaľkového prieskumu zeme. V neposlednej rade je možné metódu LMS využiť pre výpočet neznámych parametrov geodetických sietí určených terestrickým meraním. V jednom z riešení LMS sú k množine meraní podozrivých z odľahlosti priradené váhy v závislosti na použitej stratégii, ktoré je potom možno aplikovať pri výpočte sietí klasickou metódou naj-

menších štvorcov. Jedným z problémov výpočtu LMS je určenie veľkosti množiny vylúčených meraní podozrivých z odľahlosti tak, aby nenastala zásadná strata pôvodnej informácie získanej meraním, ale na druhu stranu, aby bola efektívne využitá vlasnosť vysokého bodu zvratu príznačného pre túto metódu. Vylúčenie niektorých observácií môže viesť k singularite sústavy, ktorej výpočet je vo väčšine prípadoch značne nepresný. Tieto prípady je nutné nejakým spôsobom rozpoznať a vylúčiť zo súboru výsledkov vyrovnania.

V predkladanej práci sú zhrnuté výsledky evalvácie algoritmov numerických výpočtov a algoritmov LMS na vzorových dátach, dátach generovaných výpočtovými prostriedkami a dátach geodetických sietí meraných v praxi. Vyrovnanie geodetických sietí metódou LMS je vykonané s rôznymi konfiguráciami výpočtu. Vzhľadom k variabilite výsledkov je nutné každú sieť spočítať niekoľko krát a výsledky následne zhodnotiť. Proces výpočtu sietí je automatizovaný pomocou skriptov interpretovateľného jazyka *bash*. Výsledky vyrovnania sú ukladané do databáze a spracovávané jazykom SQL.

# 1 Ciele

Cieľmi dizertačnej práce s názvom LMS v geodetických aplikáciach vyrovnávacieho počtu sú:

- Analýza metódy LMS a návrh jej použitia v geodetických aplikáciach vyrovnávacieho počtu. Tento bod zahŕňa vytypovanie rôznych stratégií určenia neznámych parametrov vhodných pre použitie v geodetických úlohách. Cieľom je nájsť vhodnú veľkosť množiny meraní tak, aby bol maximalizovaný bod zvratu s ohľadom na reálne odhadovaný počet meraní kontaminovaných odľahlými hodnotami.
- Analýza numerických algoritmov pre opakovaný výpočet sústavy lineárnych algebraických rovníc s ohľadom na špecifika vyplývajúce z riešenia geodetických úloh. Do úvahy by sa mala brať presnosť a rýchlosť výpočtu neznámych parametrov. Algoritmus by mal byť navyše pripravený na riešenie sústav lineárnych rovníc s riedkou maticou. Analyzované budú ako algoritmy pre finitný výpočet určovaného vektoru, tak algoritmy s iteračným spresňovaním výsledku.
- Implementácia rôznych možností určenia neznámych parametrov metódou LMS. Na základe analýzy z predošlých bodov budú implementované rôzne mechanizmy minimalizácie mediánu štvorcov reziduí. Pri implementácii bude braný ohľad na vysokú efektivitu a presnosť výpočtu.
- Porovnanie metódy LMS s vybranými robustnými metódami. Súčasťou je zhodnotiť prípady, kedy je výhodnejšie dať prednosť metóde LMS pred ostatnými robustnými metódami.
- Overenie algoritmov numerických výpočtov. Overenie bude spočívať v porovnaní efektivity a presnosti výpočtu vybraných numerických metód.
- Vyrovnávanie geodetických meraní metódou LMS a určenie množiny odľahlých pozorovaní. Cieľom je vyrovnávanie homogenizovaných meraní hodnôt určených v praxi s rôznym počtom odľahlých meraní za použitia rozličných konfigurácií výpočtu. Konfiguráciami výpočtu sa rozumie voľba veľkosti vzoru matice plánu a redukovaných meraní, stratégia váhovania, zabezpečenie konvergencie riešenia či počet opakovaní generovania náhodného vzoru matice plánu.
- Zhodnotenie postupov vyrovnávaní geodetických observácií na základe výsledkov vykonaných experimentov v podobe tabuliek a grafov.

## 2 Stav riešenej problematiky

Základy metódy najmenších štvorcov popísal v roku 1795 Carl Friedrich Gauss. Prvýkrát však metódu najmenších štvorcov zmienil Adrien-Marie Legendre vo svojej práci *Nouvelles méthodes pour la détermination des orbites des comètes*, teda nové metódy odhadov orbít komét, ako dodatok s názvom *Sur la Méthode des moindres quarrés*. Metódou najmenších štvorcov (MNC) je možné získať odhady neznámych parametrov minimalizáciou štvorcov opráv meraných parametrov. Metóda najmenších štvorcov je v obore geodézie najčastejšie používanou metódou pre vyrovnanie neznámych parametrov vďaka jej jednoduchosti na implementáciu do rôznych výpočtových prostredí a taktiež kvôli tomu, že pre jednu úlohu existuje práve jedno riešenie. Hlavným problémom metódy je fakt, že odhad neznámych parametrov môže byť značne ovplyvnený už jedným odľahlým meraním, ktoré môže výrazne znehodnotiť celý výpočet. Existuje preto mnoho metód slúžiacich k vylúčeniu týchto meraní zo súboru dát. Metódy, ktoré produkujú správny odhad neznámych parametrov aj v prípade väčšieho množstva odľahlých meraní, sa nazývajú robustné.

### 2.1 Popis robustných metod

Iný spôsob ako získať odhady neznámych parametrov, ktoré sú potenciálne porušené vplyvom odľahlými meraniami, ponúkajú robustné metódy. Všeobecne sa predpokladá, že pri vyrovnaní meraní robustnými metódami je výsledok v menšej miere ovplyvnený hrubými nepresnosťami pozorovaných veličín. Jedným z kritérií kvality robustnej metódy je odhad veľkosti množiny odľahlých meraní, pri ktorej je možné potenciálne získať reálny odhad hľadaných parametrov. V tomto spojení zaviedol Hampel takzvaný bod zvratu, ktorý percentuálne vyjadruje maximálne možné množstvo odľahlých pozorovaní obsiahnutých v množine meraní bez následku znehodnotenia celého výsledku. Ako bude ukázané neskôr, bod zvratu metódy najmenších štvorcov je rovný nule. Jednou zo známejších robustných metód sú tzv. M-odhady, ktorých základnou myšlienkou je nahradiť štvorce rezidui symetrickou funkciou rezidui a tú minimalizovať. Najstí riešenie tejto úlohy je v praxi pomerne triviálne. Riešením je iteračné preváhovanie najmenších štvorcov opráv. V priebehu zdokonaľovania robustných metód sa podarilo zvýšiť bod zvratu na 30%. Otázkou ale ostávalo zvýšenie bodu zvratu na čo najväčšiu možnú hranicu. Odpoveď na ňu poskytol Siegel v roku 1982, ktorý navrhol metódu riešenia neznámych parametrov opakovaným mediánom s bodom zvratu 50%. Na podobnú myšlienku prišiel Hampel a Rousseeuw a to nahradením minimalizácie sumy štvorcov rezidui minimalizáciou ich mediánu. Pre túto metódu je predpokladaný bod zvratu taktiež 50% [Rousseeuw–16]. Ťažisko

dizertačnej práce spočíva práve v metóde minimalizácie mediánu najmenších štvorcov(LMS). Metóda sa v geodetickej praxi bežne nepoužíva z niekoľkých dôvodov. Prvým z nich je pomerne náročná implementácia do výpočtového prostredia. Ďalším dôvodom je vysoká časová náročnosť výpočtu, pretože na exaktné riešenie je potrebné použiť vzory (podmnožiny) sústavy obsahujúce všetky kombinácie definovaného počtu meraní. Pre veľké sústavy to nieje možné zrealizovať, takže riešenie bude s každým priebehom výpočtu odlišné. Majme  $n$  observácii a  $p$  veľkosť množiny vylúčených meraní, výsledný počet kombinácií pre exaktné riešenie je práve  $\binom{n}{p}$ . V prípade, že máme v podstate jednoduchú úlohu, ktorá obsahuje 40 pozorovaní a 4 vylúčené merania je počet kombinácií  $\binom{40}{4} = 91390$ , čo znamená vyriešiť 91390 sústav lineárnych rovníc. Pri takom množstve výpočtov je veľmi dôležité zvažovať a vybrať správne numerické prostriedky k riešeniu tohto problému. Výber numerických metód je popísaný v kapitole 2.2. Robustnými metódami sa v geodézii zaoberalo mnoho autorov a je na ne vyhotovených rada prací. Väčšina z nich sa zamerala na jednoduchšie implementovateľné M-odhady. Konkrétne na univerzite ČVUT boli robustné metódy popísané v dizertačnej práci P. Třasáka [Trasak–30] z katedry špeciálnej geodézie, ktorý sa mimo iné zaoberal robustnými M-odhadmi a M. Jeřábka [Jerabek–23], ktorý popisoval paralelne spracovanie robustnej metódy Least Trimmed Squares (LTS). Metódu mediánov najmenších štvorcov (LMS) o niečo detailnejšie popísala v svojej dizertačnej práci E. Kanani [Kanani–31] s názvom Robust Estimators for Geodetic Transformations and GIS. Možnosť použitia LMS v globálnych navigačných systémoch bola spomenutá v článku [Yang–32].

## 2.2 Voľba numerických metód

Riešenie sústavy lineárnych rovníc  $\mathbf{Ax} = \mathbf{b}$  je všeobecne jedným z hlavných problémov vedeckých výpočtov. Algoritmy, ktoré sa používajú na riešenie sústavy lineárnych rovníc, je možné rozdeliť do dvoch základných kategórií. Do prvej kategórie patria priame metódy, kde je presné riešenie získané v konečnom počte kórkov. Do druhej kategórie spadajú metódy iteračné, u ktorých je možnosť explicitného nastavenia presnosti výpočtu. Po dosiahnutí požadovanej presnosti je výpočet ukončený. Voľba algoritmu závisí na štruktúre a type vstupnej matice  $\mathbf{A}$ . K riešeniu sústav algebraických lineárnych rovníc so všeobecnou štvorcovou plnou maticou je možné použiť Gaussovu elimináciu [Golub–18]. Gaussova eliminácia zahŕňa prevod sústavy zadaného štvorcového systému do trojuholníkového systému s rovnakým riešením. V geodézii by tento prípad so štvorcovou maticou nastal vo chvíli, keď by bol počet meraní zhodný s počtom neznámych parametrov. Všeobecne sa však predpokladá určitý počet nadbytočných meraní, čo ma za následok, že vstupná matica, taktiež na-

zývaná prvá matica plánu alebo matica diferenciálnych prírastkov,  $\mathbf{A}$  je obdĺžniková  $\mathbf{A} \in R^{m \times n}$ , kde  $m > n$ . V tomto prípade pripadá do úvahy možnosť voľby medzi vytvorením normálnych rovníc alebo pozitív algoritmu, ktorý dokáže riešiť sústavu s obdĺžnikovou maticou. Vytvorenie normálnych rovníc znamená vynásobiť sústavu lineárnych rovníc transpozíciou matice plánu  $\mathbf{A}$  čiže  $\mathbf{A}^T \mathbf{A} \mathbf{x} = \mathbf{A}^T \mathbf{b}$ . Toto riešenie je v mnohých prípadoch používané pretože po násobení matice  $\mathbf{A}^T \mathbf{A}$  je výsledkom matica štvorcová, symetrická, pozitívne semi-definitná. Semidefinitná matica vznikne za predpokladu, že vstupná matica  $\mathbf{A}$  je singulárna, teda jej determinant je rovný nule, čo znamená, že obsahuje lineárne závislé riadky resp. stĺpce. Vytvorenie normálnych rovníc však so sebou prináša taktiež niekoľko nevýhod. Hlavnou nevýhodou je vysoká časová náročnosť zostavenia normálnych rovníc s počtom operácií násobenia matíc  $n^3$ . Teoreticky sa časová náročnosť dá znížiť použitím Strassenovho algoritmu na  $n^{2.807}$  popísaného v [Golub–18]. Druhou nevýhodou zostavenia normálnych rovníc je potenciálna degradácia výhody riedkosti vstupnej matice nahradením nulových prvkov matice novovzniknutými nenulovými prvkami. Jednou z priamych metód riešenia systému lineárnych rovníc bez explicitného zostavenia normálnych rovníc je metóda singularného rozkladu. Singularný rozklad je však nevýhodné použiť v prípadoch riedkej matice pretože počas výpočtu vytvára ortonormálnu plnú maticu, ktorá môže zaberáť podstatné množstvo pamäti počítača a v konečnom dôsledku znamená stratu počítačovej výhody riedkosti vstupnej matice.

Výpočty veľkých sústav boli v tejto práci vykonané iteračnou metódou združených gradientov, ktorá sa aj napriek jej optimálnej časovej náročnosti výpočtu v geodézii takmer nepoužíva. Dôvodom je fakt, že k riešeniu neznámych parametrov vyrovnania neodmysliteľne patria odhady ich presnosti. Tieto odhady sú prvkami inverznej matice, ktorá v prípade združených gradientov nieje vytvorená. Tento problém je však vyriešený tým, že algoritmus bude využívaný robustnou metódou LMS, ktorá pre vlastné riešenie nevyžaduje prvky inverznej matice. Združené gradienty disponujú ďalšími značnými výhodami. Prvou z nich je možnosť voľby presnosti výpočtu, pretože pri výpočte veľkosti mediánu je postačujúce poznať medián na niekoľko málo platných cifier. Ďalšou výhodou je pomerne jednoduchá implementácia riedkych matíc do pôvodného algoritmu. Metóda taktiež nevyžaduje explicitné zostavenie normálnych rovníc, čo je v prípade riedkej vstupnej matice nezanedbateľná prednosť. Jednoduchou modifikáciou algoritmu je teda možné pracovať priamo s obdĺžnikovou maticou [Saad–17]. Rýchlosť konvergenie je v určitých prípadoch možné zvýšiť použitím vhodného prepodmieňovača. Asi najpoužívanejším prepodmieňovačom je neúplný Choleskýho rozklad. Prepodmieňovač však nieje možné použiť v momente, keď nie sú zostavené normálne rovnice pretože vstupnou maticou do choleskýho rozkladu musí byť štvorcová, symetrická, pozitívne definitná matica. Kompromisom by mohol byť výber bloku vstupnej matice, ktorý je vý-

znamný z hľadiska konverencie, z tohto bloku zhotoviť normálne rovnice, vyhotoviť neúplný Choleskýho rozklad a prepodmieňovač použiť do výpočtu. Tento prípad bude predmetom ďalšieho skúmania. Značnou výhodou prepodmienenia je fakt, že takmer určite zvyšuje presnosť výpočtu.

## 2.3 Voľba knižnice riedkych matíc

Ako už bolo spomínané vstupná matica plánu má veľmi často riedku štruktúru. Pri časovo náročných výpočtoch by bola chyba tento predpoklad nevyužiť. Na výpočty s riedkymi maticami existuje pre jazyk C++ mnoho sofistikovaných knižníc, z ktorých najznámejšie sú SparseLib++ a Eigen. SparseLib++ obsahuje funkciu na riešenie metódy združených gradientov s možnosťami predpodmienenia. Pre efektívne matematické operácie používa SparseLib++ knižnicu Sparse BLAS programovanú v jazyku Fortran. Dátové štruktúry a základné operácie s riedkymi maticami obsahuje taktiež projekt GNU Gama. Obsah knižnice SparseLib++ zahŕňa niekoľko rokov vývoja algoritmov riedkych matíc, takže je možné očakávať optimálne postupy riešenia rôznych výpočtov numerickej matematiky. Aj napriek vysokej efektivite algoritmov knižnice SparseLib++ boli pre túto prácu zvolené dátové štruktúry programu GNU Gama, ktoré boli vyvinuté podľa návrhu práce matematikov George a Liu [Liu–21]. Dôvodom je, že algoritmy obsiahnuté v dizertačnej práci naväzujú na dátové štruktúry programu GNU Gama a využívajú niektoré jeho časti, napríklad zostavenie matice plánu, vektoru redukovaných meraní a kovariančnej matice meraní. V ďalšom prípade algoritmus združených gradientov vyžaduje veľké množstvo úprav závislých na vlastnostiach geodetických úloh. Všeobecne sa nedoporučuje algoritmy komplexných knižníc upravovať z dôvodu možnosti porušenia vnútorných štruktúr, či ovplyvnenia funkčnosti upravených objektov, ktoré závisia na iných častiach knižnice.



### 3 Metóda minimalizácie mediánov štvorcov reziduí

Každé meranie akýchkoľvek veličín je zaťažené skutočnou chybou neznámej veľkosti. Na základe opakovania merania a známych vzťahov je možné určiť odhady týchto chýb a v závislosti na odhade chýb určiť odhad meraných veličín. Lineárny model odhadu meraných veličín je zostavený tak, že očakávané hodnoty meraní sú vyjadrené ako lineárne funkcie neznámych parametrov so známymi koeficientami. Lineárny vzťah medzi parametrami a observáciami vznikne linearizáciou fyzikálnych, či matematických vzťahov. Existuje mnoho spôsobov ako odhadnúť merané veličiny resp. ich skutočné chyby. V tejto práci je však dôraz kladený na robustné odhady. Množina observácii môže byť znehodnotená meraním so značnou chybou. Toto meranie označujeme ako odľahlé a spôsobuje zmenu náhodného rozdelenia meraných veličín. Pokiaľ metóda nieje citlivá na zmenu tohto rozdelenia, je možné nazvať ju robustnou. Ináč povedané robustný odhad parametrov nieje citlivý na odľahlé merania. Metóda najmenších štvorcov, metóda maximum-likelihood, či metóda najlepšieho nestranného odhadu nepatria medzi robustné metódy, avšak dokážu poskytnúť správny odhad pokiaľ obsahujú nejakú z metod testovania odľahlých meraní. Robustné metódy poskytnú optimálny odhad aj bez explicitného testu odľahlých meraní. Ako už bolo spomenuté, metóda najmenších štvorcov nepatrí medzi robustné metódy. Aby ju ale bolo možné do tejto kategórie zaradiť, mnoho autorov sa snažilo nahradiť minimalizáciu sumy štvorcov reziduí minimalizáciou niečoho iného, napríklad sumy absolútnych hodnôt reziduí, či v prípade LMS minimalizáciou mediánu štvorcov reziduí, ktorú predstavil Rousseeuw roku 1984. LMS vracia správny odhad neznámych parametrov aj v prípade pomerne veľkého počtu kontaminovaných meraní a to do 50%. Ako už z názvu vyplýva, metóda spočíva v nahradení sumy senzibilnej na odľahlé merania mediánom, ktorý je veľmi robustný.

Majme klasický lineárny model daný ako

$$\ell = \mathbf{A}\mathbf{x} + \epsilon \quad (3.1)$$

kde  $\ell \in R^{n \times 1}$  je vektor meraných veličín,  $\mathbf{x} \in R^{d \times 1}$  je vektor neznámych parametrov,  $\mathbf{A} \in R^{n \times d}$  matica diferenciálnych prírastkov a  $\epsilon \in R^{n \times 1}$  vektor skutočných chýb meraní. Predpokladajme, že matica  $\mathbf{A}$  ma hodnosť  $d$  a všetky jej sub-matice rozmeru  $d \times d$  sú nesingulárne. Najznámejší odhad  $L_2$ -normy, teda metóda najmenších štvorcov, ktorá minimalizuje štvorce reziduí, je v mnohých prípadoch optimálny avšak príliš citlivý na odľahlé merania a taktiež na body, ktoré ovplyvňujú výsledok odhadu neznámych parametrov (*leverages*). Vplyvové body, teda body nejakým spôsobom ovplyvňujúce odhad neznámych parametrov, môžeme rozdeliť na odľahlé

body tzv. *outliers*, ktoré niesu vzdialené od regresnej priamky, ale sú vzdialené od skupiny meraní, na vyššie spomenuté vertikálne odľahlé merania (leverages) a všeobecne odľahlé merania v oboch osách. Tieto body sú podrobne popísané v kap. 4.6

Minimalizácia mediánov štvorcov reziduí je daná vzťahom

$$\text{med}(v_i^2) \rightarrow \min \quad (3.2)$$

kde  $v$  tzv. reziduá alebo taktiež opravy sú rozdiely medzi observovanými hodnotami a aktuálnym odhadom. Majme  $n$  observácii  $(a_i, l_i) = (a_{i1} \dots a_{id}, l_i)$ , ktoré patria do lineárneho priestoru riadkov vektoru dimenzie  $d + 1$  a neznáme parametre  $x$  prináležia stĺpcovému vektoru dimenzie  $d$ . Všetky observácie, pre ktoré platí, že  $x_i = 0$  môžu byť z výpočtu odstránené, pretože neobsahujú informácie o neznámych parametroch. Metóda LMS má pre konečný počet observácií vždy riešenie a pri akejkolvek zmene  $n$  získame jedinečný odhad  $\mathbf{x}$ .

Pomocou metódy LMS sa určuje odhad riešenia  $\mathbf{x}$  tak, že sa zo súboru pozorovaných veličín vyberie práve taký súbor  $V$ , ktorý ma najmenší medián štvorcov rezidui  $v$ . Súbor meraní  $V$  obmedzený o určitú konečnú množinu meraní nazývame vzor sústavy. Pre akýkoľvek vzor  $V$  o  $m$  - observáciach a regresný odhad  $T$ , pre ktorý platí, že  $\beta(p; T, V)$  je supremum  $\|T(Vt) - T(V)\|$  pre všetky porušené vzory  $Vt$  a nejaký ľubovoľný počet dát  $p$  nahradený inými ľubovoľnými dátami, je bod zvratu

$$\tau(T, V) = \min\{p/m; \beta(p; T, V) \text{ je } \infty\} \quad (3.3)$$

Ináč povedané je to najmenšie množstvo kontaminovaných meraní, ktoré spôsobí odľahlosť odhadu  $T(Vt)$  od odhadu  $T(V)$ . V prípade MNČ je  $\tau(T, V) = 1/m$ , pretože už jedno odľahlé meranie ovplyvní celý odhad. Pre LMS platí, že pokiaľ je  $p > 1$  tak bod zvratu je  $([n/2] - p + 2)/n$ , čo znamená, že bod zvratu je mierne závislý na rozmere vzoru  $V$ . Položením  $n \rightarrow \infty$  a pre fixný počet  $p$  je zrejme, že teoretický bod zvratu v prípade MNČ je 0% naproti tomu LMS 50%. Pri kontaminácii nad 50% už nieje možné rozlíšiť správny a nesprávny súbor dát. To znamená, že hodnota 50% je maximálna hranica pre počet kontaminovaných meraní.

Určenie, či veľkosť rezidua  $v_i$  presiahlo veľkosť tolerancie je možné pomocou odhadu veľkosti skutočnej chyby meraní  $\hat{\sigma}$ :

$$\hat{\sigma} = C \sqrt{\text{med}(v_i^2)} \quad (3.4)$$

kde  $C$  je korekčný faktor použitý pre dosiahnutie konzistencie rozdelenia chýb. Robustný odhad štandardnej smerodajnej odchýlky meraní je podľa [Rousseeuw–48]

$$\sigma_v^2 = 1,4826 \left( 1 + \frac{5}{n-d-1} \right) \sqrt{\text{med}(v_i^2)} \quad (3.5)$$

Odhad veľkosti skutočnej chyby je možné aplikovať ako váhy pri odhade neznámych parametrov v oboch prípadoch a to tak, že

$$w_i = \begin{cases} 1 & \text{pre } |v_i/\hat{\sigma}| \leq 2,5 \\ 0 & \text{pre } |v_i/\hat{\sigma}| > 2,5 \end{cases} \quad (3.6)$$

Takáto definícia rozdelenia váh je nazývaná ako prísne zamietnutie. Druhou možnosťou je rozdeliť váhy plynule pomocou nejakej bližšie špecifikovanej funkcie. Plymulé rozdelenie váh je použité pri vyrovnávaní geodetických sietí v kapitole 13.2.2. Odhad toho ako dobre vyjadruje LMS odhad variabilitu meraní určuje koeficient odhadu  $R^2$

$$R^2 = 1 - \left( \frac{\text{med } |v_i|}{\text{med } |\ell_i|} \right)^2 \quad (3.7)$$

Čím vyššia je hodnota  $R^2$ , tým lepší je regresný odhad.

Reziuda získané z klasickej metódy najmenších štvorcov nie sú veľmi vhodné pre diagnostiku odľahlých meraní. Naproti tomu vysoko robustné odhady zaisťujú pomerne presnú informáciu určenú pre detekciu všetkých odľahlých meraní v observovaných dátach. Bezdimenzionálne štandardizované odhady opráv sú vyjadrené ako  $v_i/\hat{\sigma}$ . Kvantitatívna informácia sa získa zobrazením týchto štandardizovaných opráv a odhadovaných neznámych parametrov alebo indexov meraní. Analýzou tejto informácie je možné detekovať odľahlé hodnoty meraní.

### 3.1 Algoritmy pre výpočet LMS

Skôr než prejdeme k popisu samotných algoritmov predstavme si pojem časová a priestorová náročnosť algoritmu. Časová náročnosť, niekedy nazývana taktiež ako asymptotická zložitosť  $O(n)$ , je maximálny počet krokov, ktoré algoritmus vykoná na vstupnej množine dát veľkosti  $n$ . Priestorová náročnosť algoritmu je minimálne nutné množstvo alokácie pamäti počítača pre vykonanie výpočtu. Nevýhodou metódy LMS je, že vlastný výpočet metódy je časovo pomerne náročný. Úlohu je možné zaradiť do  $NP$  ťažkých úloh. Podľa [Rousseeuw–16] metóda LMS konverguje oproti iným robustným metódam omnoho pomalšie a to s mierou  $n^{1/3}$  oproti bežnej konvergencii s mierou  $n^{1/2}$ . Jednoznačný odhad neznámych parametrov v závislosti na minimálnom mediáne je možné získať vyriešením všetkých možných kombinácií meraní.

Existuje preto mnoho prístupov ako tento odhad LMS získať. Menšie sústavy je možné riešiť exaktne a pre sústavy veľkých rozmerov je v závislosti na časovej náročnosti nutné použiť metódy aproximujúce odhad LMS. V praxi sa pre aproximáciu LMS používa systém PROGRESS. Tento výpočtový prostriedok má však isté obmedzenia popísane v nižšie uvedených sekciách. Vzory veľkých sústav sú vyberané náhodne a časová náročnosť algoritmov systému PROGRESS je  $O(n \log n)$  pre fixnú dimenziu. V nasledujúcich kapitolách budú predstavené ďalšie doterajšie algoritmy pre určenie riešenia LMS.

### 3.1.1 Algoritmy pre výpočet odhadu LMS v rovine

Geometrická reprezentácia metódy LMS spočíva v určení pásu ohraničeného dvoma rovnobežnými priamkami s minimálnou šírkou, pričom tento pás musí obsahovať aspoň polovicu observovaných veličín. Majme v rovine  $n$  bodov lokalizovaných súradnicami  $x_i$  a  $y_i$  pre každý  $i$ -tý bod. Odhad LMS v rovine je daný priamkou

$$y = a + bx \tag{3.8}$$

Funkcia minimalizácie mediánu štvorcov reziduí je po častiach lineárna a môže mať kvadratický počet lokálnych extrémov  $O(n^2)$ . Existuje niekoľko algoritmov, ktoré dokážu lokalizovať minimum. Najlepší z nich má komplexitu  $O(n^3)$ . Algoritmus, ktorý sa v praxi najčastejšie vyskytuje, je jeden z najmenej efektívnych s časovou náročnosťou  $O(n^3 \log(n))$ . [Steele–1] navrhuje zvýšenie efektivity s použitím heuristických algoritmov s očakávanou časovou komplexitou  $O((n \log(n))^2)$ . Popis charakteristiky lokálneho minima stanovuje podmienku pre určenie minima globálneho. [Steele–1] uvažuje body v dvojrozmernom kartézskom priestore. Pre každú trojicu bodov sú určené práve tri bisektujúce priamky. Algoritmus, ktorý určuje exaktné riešenie LMS v konečnom čase hrubou silou má časovú náročnosť  $O(n^4)$ . V tomto algoritme sa pre každú z  $n(n-1)(n-2)/6$  priamok určí medián. S týmto algoritmom bolo možné LMS prakticky exaktne vyriešiť v rovine obsahujúcej približne 50 bodov<sup>1</sup>. Operácie pôvodného algoritmu, nazvime ho *brute-force*, je možné zredukovať zafixovaním  $b_{ij} = (y_i - y_j)/(x_i - x_j)$ , kde hodnoty  $x$  a  $y$  reprezentujú kartézke súradnice bodov a tým sa zredukuje problém na  $g(a) = \text{median}(|z_i - a|)$ , kde  $z_i = y_i - bx_i$  a  $g(a)$  je minimum rovný dĺžke najmenšieho intervalu obsahujúceho polovicu  $z_i$ .  $\mu(n)$  je medián určený z bodov  $1, \dots, n$ . Pokiaľ platí, že  $z_p + \mu(n) - z_p = \min[(z_j + \mu(n) - z_j), j = 1, \dots, \mu(n)]$ , tak potom musí platiť, že  $\min(g(a)) = z_p + \mu(n) - z_p$  a  $a = (z_p + \mu(n) + z_p)/2$ . Algoritmus teda iteruje cez všetky páry bodov  $u$  a  $v$  ale mnoho z týchto iterácií je možné zredukovať z času

<sup>1</sup>Počet platil pre výpočtovú techniku použitú v roku 1986. Dnes je toto číslo mnohonásobne vyššie

$O(n \log(n))$  na  $O(\log(n))$ . Lepšie výsledky je možné získať modifikáciou algoritmu *brute-force* nazývanou ako algoritmus  $W$  (wedge - klín). Algoritmus je založený na určení  $W = [b_L, b_U]$  klínu tvoreného dvoma priamkami a to tak, že sa vezmú parametre priamky  $a$  a  $b$  lepšieho z riešenia klínu a eliminujú sa všetky  $u$  a  $v$ , pre ktoré platí  $b_{uv} \in W$ . Klín je možné určiť v lineárnom čase vyriešením  $n - 2$  jednoduchých rovníc, pre ktoré platí, že  $|u_i|$  je zhodná z  $|u_p|$ , kde  $i$  je rôzne od  $p$  a  $q$ . Na elimináciu niektorých  $b_{uv}$  je potrebný výpočet a triedenie všetkých  $b_{uv}$  v čase  $O(n^2 \log(n))$ . Pre určení hodnotu klínu  $W$  je možné eliminovať  $b_{uv} \in W$  binárnym vyhľadávaním v čase  $O(\log(n))$ . Optimálny prienik pre  $b_{uv}$  a generovaný klín je možné získať v čase  $O(n \log(n))$ . Celkový čas je teda  $K O(n \log(n)) + (n^2 - K) \log(n) + O(n^2 \log(n))$ , kde  $K$  je celkový počet párov bodov. Algoritmus  $W$  má priestorovú náročnosť  $O(n^2)$ . V prípade aproximácie riešenia, je pre 1000 bodov potreba analyzovať  $\approx 100$  vzorov. Pre veľké sústavy je potom nutné použiť heuristický algoritmus  $R$  založený na algoritme  $W$ , ktorý ale oproti algoritmu  $W$  nekontroluje všetky  $b_{uv}$ , ale vyberá  $b_{uv}$  len náhodne.

Riešením LMS v rovine sa taktiež intenzívne zaoberal D.L. Souvaine, ktorý predstavil jednu z najznámejších prác týkajúcej sa riešenia výpočtu LMS v rovine popísanej v [Souvaine-2]. V práci je prezentovaný *sweep-line* algoritmus s časovou zložitou  $O(n^2 \log(n))$ , ale iba s využitím  $O(n)$  priestoru. Ďalšie vylepšenie algoritmu poskytol vývoj dátových štruktúr, ktorý umožnil riešiť LMS problém s časovou zložitou  $O(n^2)$ , ale s priestorovou náročnosťou taktiež  $O(n^2)$ . Algoritmus využíva dve v tom čase pomerne nové vlastnosti z tzv. výpočtovej geometrie a to afinnú dualitu a dátové štruktúry, ktoré zefektívňujú vyhľadávacie a zoraďovací proces. Výhodou týchto dvoch vlastností je možnosť ich vzájomnej kombinácie a s jej pomocou je možné riešiť štatistické úlohy, ktoré sa zaoberajú bodmi a priamkami v rovine. V štatistike sa dualita bodu a priamky vyskytuje už pomerne dlhú dobu a využíva sa v algoritmoch pre rôzne riešenia robustných metód. Na pochopenie *sweep-line* algoritmu je nutné vysvetlenie niekoľkých pojmov.

### Dualita bodu a priamky:

Transformácia  $T$  sa aplikuje tak, že sa bod  $(a, b)$  vloží do priamky  $y = ax + b$  a priamka  $y = ax + b$  sa vloží do bodu  $(-a, b)$ . Bod  $P$  transformácie  $T$  je určený dvoma priamkami  $L_1$  a  $L_2$  a namapovaný do priamky  $TP$  ktorá je určená dvoma bodmi  $TL_1$  a  $TL_2$ . Rovnako priamka  $L$  určená bodmi  $P_1$  a  $P_2$  je namapovaná do bodu  $TL$  určeného priamkami  $TP_1$  a  $TP_2$ . Transformácia má niekoľko dôležitých vlastností. Prvou z nich je dodržanie zoradenia *nad* a *pod* bodov a priamok. To znamená, že pokiaľ je bod  $P$  nad  $L$ , tak aj priamka  $TP$  bude nad priamkou  $TL$ . Druhou vlastnosťou je zachovanie zvislej vzdialenosti medzi bodmi a priamkami, tzn. že  $TL$  a  $TP$  sú nad sebou v rovnakej vzdialenosti ako  $L$  nad  $P$ . Táto vlastnosť

je veľmi dôležitá pre určenie mediánu štvorcov reziduí výberom priamok. A treťou z vlastostí je, že pokiaľ je sklon  $L$  väčší ako sklon  $L'$ , tak aj x-ová súradnica priamky  $TL$  je menšia než x-ová súradnica  $TL'$ .

### Ekvioscilácia a jej dualita:

Priamka  $l(a, b) = (x, y) : y = a + bx$  s reziduími  $v_i(a, b)$  "bisektuje" trojicu bodov tak, že všetky reziduí majú rovnakú veľkosť, ale opačné znamienka. Takže pokiaľ  $x_{i1} < x_{i2} < x_{i3}$  a zároveň  $r_{i1} = -r_{i2} = r_{i3}$ , tak hovoríme, že priamka ekviosciluje vzhľadom k týmto bodom.

Algoritmus navrhovaný v [Souvaine-2] má časovú náročnosť  $O(n^2)$ . Ekviosciláčna priamka  $l(a, b)$  má dve príbuzne priamky so sklonom  $a$ . Priamka  $L_1$  je určená bodmi  $P_1(x_{i1}, y_{i1})$  a  $P_3(x_{i3}, y_{i3})$  a  $L_2$  prechádzajúca bodom  $P_2(x_{i2}, y_{i2})$ . Kľúčová vlastnosť LMS je, že počet bodov  $K$  medzi priamkami musí spĺňať  $K = (n - 4)/2$  pre párny počet bodov a  $K = (n - 5)/2$  pre počet nepárny. LMS sa určí tak, že pre každú trojicu bodov  $P_1, P_2$  a  $P_3$  so zoradenými x-ovými súradnicami sa určia priamky  $L_1$  a  $L_2$  a následne sa nájde trojica s počtom  $K$ -bodov. Výsledná vzdialenosť medzi priamkami je minimálna a riešenie LMS prechádza stredom tohto pruhu. Redefiníciou problému je možné zmeniť hľadanie  $K$ -bodov medzi priamkami  $L_1$  a  $L_2$  na  $K$  priamok medzi bodmi  $TL_1$  a  $TL_2$ , kde  $TL_1$  je bod určený priamkami  $TP_1$  a  $TP_3$ .

### Sweep-line algoritmus a jeho priestorová náročnosť $O(n)$ :

*Sweep-line* algoritmus používa dve dátové štruktúry. Prvou z nich je jednoduchý spojový zoznam, v ktorom sú z geometrického hľadiska zoradené a uložené priamky. Priamky sú zoradené podľa ich sklonu. S každou priamkou je uložený pointer na priamky 1 nad  $L$ ,  $K$  nad  $L$ , 1 pod  $L$  a  $K$  pod  $L$ . Pointre priamok vyskytujúcich sa na začiatku a na konci zoznamu ukazujú na hodnotu NULL. V druhej dátovej štruktúre je uložená podmnožina  $n - 1$  z množiny  $n(n - 1)/2$  bodov  $P_{ij}$ , cez ktoré priamky prechádzajú. Dátová štruktúra umožňuje prístup k najmenšiemu prvku v čase  $O(1)$ . Vkladanie a mazanie prvku je vykonané v čase  $O(\log(n))$ . Body sú zoradené v závislosti na ich x-ových súradniciach. Ako druhú dátovú štruktúru zvolil autor haldu. V prvom kroku, je nutné vyhľadať priamku najviac vľavo od všetkých bodov  $P_{ij}$ . Aby ju bolo možné nájsť je nutné lokalizovať priesečník  $P_{ij}$  ležiaci najviac vľavo a to v závislosti na zoradení podľa x-ových súradníc. Pre interakciu medzi oboma dátovými štruktúrami je vytvorená množina pointrov a to tak, že pointer bodu  $P_{ij}$  ukazuje na príslušné priamky  $L_i$  a  $L_j$  zo zoznamu a naopak pointer priamok ukazuje na im príslušný bod. Pre priamku  $L_0 : x = h$  platí, že  $h < x_{min}$ . Ďalšou priamkou je tzv. *sweep* priamka  $L$  v prvom kroku totožná s  $L_0$ . V inicializačnom kroku je časová

náročnosť potrebná pre zoradenie priamok v zozname rovná  $O(n \log(n))$  s priestorom  $O(n)$  a časová náročnosť vloženia bodov do haldy  $O(n)$  s priestorom taktiež  $O(n)$ . Algoritmus pre výpočet LMS je s použitím týchto dátových štruktúr takmer triviálny.

- Priamka  $L$  iteruje cez všetky body haldy.
- Pritom je ale nutné:
  - udržiavať zoznam zoradený v vzostupnom poradí
  - udržiavať pointre v zozname tak aby splňovali počiatočnú podmienku
  - halda vždy obsahuje bod s najmenšou súradnicou  $x$  všetkých bodov na pravo od priamky  $L$
- Keď  $L$  dosiahne nový bod  $P_{ij}$ , tak sa vypočíta vertikálna vzdialenosť medzi hornou a dolnou priamkou  $K$  a tá sa porovná s predošlým optimom.
- Následne sa vymení relatívna pozícia  $L_i$  a  $L_j$  v zozname.

Z celého zoznamu bodov  $n(n-1)/2$  je nutné nájsť priamku  $L_0$  nimi prechádzajúcu tak, aby bola najbližšie *sweep* priamke  $L$  určenou priamkami  $L_i$  a  $L_j$ . Pre každú z  $O(n^2)$  pozícií priamky  $L$  je potreba aktualizovať tri body zo zoznamu v čase  $O(\log(n))$ . Celkový čas algoritmu je teda  $O(n^2 \log(n))$ .

V prípade viacrozmernej regresie je možné algoritmus rozšíriť s očakávaným značným nárastom režijných nákladov. Existujú nástroje vyhľadávania bodov v  $d$ -rozmernom priestore s princípom duality. Tie však neposkytnú riešenie pre LMS v čase lepšom než  $O(n^d)$ .

[Massart–11] publikovali algoritmus LMS s odlišným prístupom určenia odhadu. Algoritmus určuje sklon priamky, pre ktorú je medián štvorcov reziduí minimálny skenovaním všetkých uhlov s určitou veľkosťou. Po tom, čo je vybraná priamka vykazujúca najlepší výsledok, je táto procedúra opakovaná s malým inkrementovaním uhla blízkeho optimu. Majme špeciálny prípad minimalizácie  $med_i(z_i - b)^2$ , kde sa snažíme pomocou LMS odhadnúť hodnotu  $b$ . Prvým krokom je zoradenie  $n$ -observácií  $z$  podľa veľkosti od najmenšieho po najväčší. Potom najmenší rozdiel je hľadaný  $z(z(h) - z(1) \dots z(n) - z(n - h + 1))$ , kde  $h$  je najväčšie celé číslo, ktoré je ale menšie nanajvýš rovné  $(n/2) + 1$ . Pokiaľ je  $z(k) - z(j)$  najmenší rozdiel potom je  $b = 1/2(z(j) + z(k))$ . Pre jednoduchú lineárnu regresiu minimalizácie  $med_i((y_i - ax_i) - b)^2$  je pre každé  $a$  aplikovaný algoritmus uvedený vyššie. Kde  $z_i = y_i - ax_i$ . Výsledkom je priamka, pre ktorú platí, že medián štvorca rezidua je minimálny. Aplikácia na  $d$ -rozmerný problém je obdobná. V geodézii si s výpočtom LMS v rovine nevystačíme, preto je nutné zamerať sa na  $d$ -rozmerný problém.

### 3.1.2 Algoritmy pre exaktný výpočet odhadu LMS v $d$ -rozmernom priestore

V článku [Dallal–9] predstavil program *LMSMVE*, ktorý vykonáva robustnú regresiu s použitím metódy LMS. Program taktiež určuje vzdialenosti medzi odhadovaným riešením a bodmi pre stanovenie množiny odľahlých meraní. Program bol vytvorený v jazyku Fortran. *LMSMVE* dokázal spracovať sústavy do 300 meraní a 10 nezávislých neznámych parametrov. Regresná plocha LMS leží v prostriedku medzi dvoma hyper-rovinami a obsahuje množinu polovice meraní. *LMSMVE* rieši úlohu so všetkými možnými vzormi sústavy veľkosti  $p + 1$ , kde  $p$  je veľkosť množiny vylúčených meraní. V dobe predstavenia algoritmu nebolo možné týmto prístupom realizovať výpočet pre sústavy, ktorých kombinácie  $nC_p + 1$  presahujú dve stovky. V dnešnej dobe je možné počet kombinácii zvýšiť. Pokiaľ je ale počet kombinácii príliš vysoký, tak je použitý náhodný výber vzorov sústavy. Pomocou štandardnej odchýlky pre robustný odhad definovaný rovnicou 3.5 sa potom indentifikujú merania s odľahlými meraniami. Za odľahlé meranie je považované také meranie, ktorého smerodajná odchýlka z robustného vyrovnania presahuje 2,5 násobok štandardnej odchýlky. [Dallal–9] navrhuje pre každý vzor určiť kovariančnú maticu, z ktorej sa potom určí elipsoid obsahujúci odpovedajúce merania. Robustné vzdialenosti sú potom získané z elipsoidu s minimálnym objemom. Tento prístup nieje možné v disertačnej práci použiť a to z dôvodu, že pre výpočet sústavy lienárnych rovníc je použitá metóda združených gradientov bez vyhotovenia normálnych rovníc. Tým pádom nieje časovo optimálne určovať kovariančnú maticu v každom kroku výpočtu.

V článku [Erickson–5] bol predstavený algoritmus pre exaktný výpočet LMS v  $d$ -rozmernom priestore s výpočtovým časom  $O(n^d \log(n))$ . Autori popísali redukciiu v lienárnom čase založenú na tzv. *affine degeneracy* probléme.  $h(P, k)$  je najmenšia ortogonálna vzdialenosť medzi dvoma rovinami obsahujúcimi  $k$ -bodov množiny  $P$ . Exaktný algoritmus minimálnej výšky pre viacrozmerný priestor vylepšuje algoritmus topologického prečesávania o faktor  $O(n)$  a je založený na *sample-sweep* prístupe. V prvom kroku sa vyberie náhodný vzor s kritickou vertikálnou vzdialenosťou, na ktorý sa aplikuje binárne vyhľadávanie pre identifikáciu malého intervalu obsahujúceho optimálnu vzdialenosť a následne sa aplikuje *sweep* algoritmus na nájdenie optimálnej vzdialenosti.

Ďalším z algoritmov určujúcich exaktne riešenie navrhol [Stromberg–15] a je založený na prístupe rozdelenia dát na dve polovice a to na tzv. vonakjšiu a vnútornú. Vonakjšia polovica obsahuje dáta s počtom  $h$ -reziduami väčšími než vnútorna polovica. LMS riešenie sa určí ako Chebyshevov minmax odhad vnútornej polovice dát. Chebyshevov odhad je známy ako minimalizácia absolútných odchýlok  $d + 1$  bodov z čoho vyplýva, že LMS je špeciálny prípad Chebyshevoava odhadu. Exakt-



ným riešením je otestovanie všetkých vzorov obsahujúcich  $d + 1$  bodov a pre každú vnútornu polovicu vyriešiť Chebyshevov odhad a overiť, či odhad pokrýva aspoň  $h$  prípadov. Tento exaktný algoritmus je použiteľný pre pomerne malý počet dát a neznámych parametrov. Pre  $n = 30$  a  $d = 4$  je nutné vyriešiť a následne overiť 30 000 Chebyshevových odhadov. Pre  $n = 40$  a  $d = 4$  je to už približne 4mil.

### 3.1.3 Algoritmy pre približný výpočet odhadu LMS v $d$ -rozmernom priestore

Pretože exaktné určenie LMS veľkých sústav je náročné a v reálnom čase prakticky nemožné, ponúka sa možnosť aproximácie riešenia na základe heuristických algoritmov. Tri vylepšené varianty LMS algoritmu založeného na náhodnom výbere vzorov matice  $d$ -rozmerného vektorového priestoru s ohľadom na hyper-rovinu prechádzajúcu bodmi  $p$  navrhli autori v článku [Joss-3]. Autori využili geometrickú definíciu problému určenia riešenia LMS, teda vyhľadanie najužšieho pásu rozmeru  $2\rho$  určeného dvoma rovnobežnými priamkami, ktorý obsahuje  $\lfloor n/2 \rfloor + 1$  bodov z množiny  $p$ . V *brute-force* algoritme je medián určený so zložitosťou  $O(n)$ . Aj napriek tomu, že je možné v niektorých krokoch určenie preskočiť a tým podstatne zredukovať čas výpočtu, tak to nezníži celkovú náročnosť výpočtu, kde je nutné rešiť sústavu lineárnych rovníc všetkých vzorov sústavy. Algoritmus použitý v programe PROGRESS s časovou náročnosťou  $O(n \log n)$  operácii pre každý vzor dokáže nájsť exaktné riešenie pre  $d = 1$  v jednom kroku a pre  $d = 2$  v  $\binom{n}{2}$  krokoch. Algoritmus však nieje použiteľný pre  $d > 2$  sustavy. Pre tieto sústavy je implementovaný algoritmus uvedený v [Joss-3]. Posledný z popisovaných algoritmov zlepšuje predchádzajúci algoritmus tým, že sa vykonáva lokálna minimalizácia susedných vzorov aktuálneho náhodného vzoru. Ideou je generovanie "lepších" vzorov, ktoré zahŕňajú  $d+1$  observácií s najväčšími reziduami obsiahnutými v aktuálnom optimálnom páse.

V ďalšom článku [Hawkins-6] autor navrhuje heuristický *feasible set* algoritmus založený na Chebyshevovom minmax odhade, ktorého výsledkom sú hodnoty spĺňajúce podiemku pre vyhľadanie optima, ktoré poskytuje určenie exaktného riešenia  $d$ -dimenzionálneho problému s pravdepodobnosťou  $P = 1$  pri zvyšujúcom sa počte iterácii. Prezintovaný algoritmus má tú vlastnosť, že pri opakovaných iteráciách nad náhodnými vzormi výpočet konverguje ku globálnemu minimu teda riešeniu LMS. Doterajšie rozsiahle výsledky nasvedčujú, že počet vzorkov matice musí byť dostatočne veľký, aby bolo možné dosiahnuť spoľahlivú aproximáciu. [Portnoy-39] ukázal na príklade, že pre nejakú veľkosť sústavy môže byť počet vzorov  $N = 10000$  nedostačujúci, ale  $N = 50000$  už pre spoľahlivé určenie minima môže postačovať. Menší počet preskúmaných vzorkov môže dávať dobré výsledky, ale zároveň môže úloha vyžadovať preskúmanie vzorkov veľkej množiny. V [Ruppert-38] autori navrhli

aproximáciu, ktorá spresňuje získaný výsledok tak, že pri náhodnom výbere vzoriek sa predchádzajúce riešenie nezhodí, ale použije sa k spresneniu odhadu prechodu na ďalší vzor, ktorý môže potenciálne zlepšiť dočasné riešenie. V publikácii bolo ukázané, že spresňovaním dočasného riešenia je možné získať zrovnateľne presný výsledok ako pri náhodnom výbere, ale v celkovo kratšom čase. Ďalší z návrhov bol skombinovať Strombergovo riešenie uvedené v kapitole 3.1.2 pomocou Chebyshevových odhadov s Ruppertovým spresňovaním vzorov.

### Chebyshevov odhad:

Odhad je riešený ako duálna úloha lineárneho programovania aplikovaním simplexového algoritmu minimalizácie hodnoty  $z$ .

$$Y - X\beta \leq zY - X\beta \geq -z \quad (3.9)$$

Efektívnejšie riešenie tejto úlohy je možné prepísať ako minimalizáciu

$$G = \sum_{i=1}^n (c_i^+ + c_i^-) \quad (3.10)$$

kde  $c_i^+$  a  $c_i^-$  reprezentujú kladnú a zápornú cenu obmedzujúcich podmienok

$$Y_i - \sum x_{ij}\beta_j \leq z \quad a \quad Y_i - \sum x_{ij}\beta_j \geq -z \quad (3.11)$$

Algoritmus riešenia úlohy pomocou Chebyshevovho odhadu:

1. Inicializačný krok Chebyshevovho odhadu podmnožiny  $d + 1$ .
2. Určenie cien pomocou výberu pivotov simplexovou metódou.
3. Pokiaľ sú všetky ceny kladné tak je nájdené riešenie a podmnožina je optimálna. Pokiaľ nie, tak sa vyberie stĺpec s najmenšou negatívnou cenou a vloží sa do pôvodnej podmnožiny.
4. Vykoná sa znova výber pivotov simplexovou metódou a opakujú sa kroky 2-4.
5. Inverziou minimalizačného kritéria  $G$  je minimalizované  $z$ .

Modifikáciou kroku 3 je možné získať riešenie LMS a to tak, že:

pokiaľ je aspoň  $h$  cien pozitívnych tak LMS riešenie je optimálne. Pokiaľ tomu tak nieje, vyberie sa stĺpec s algebraicky minimálnou negatívnou cenou.

Podľa [Hawkins–6] algoritmus konverguje od akejkoľvek počiatočnej množiny k množine optimálnej. Táto optimálna množina sa nazýva *feasible* riešenie. Opakovaním algoritmu s rôznymi počiatočnými množinami je možné získať  $n$ -feasible riešenia a jedno z nich, ktoré splňuje podmienku, že  $G$  je maximálne je riešením LMS. Označme podmnožinu s  $d + 1$  prvkami indexom  $J$ . Určme  $S$  ako maximálne reziduum podmnožiny  $J$ .

$$S = \frac{\sum_J v_{iJ}^2}{\sum_J |v_{iJ}|} \quad (3.12)$$

Označme počet absolútnych hodnôt reziduí menších alebo nanajvýš rovných  $S$  ako  $m$ . Potom môžu nastať tieto situácie:

- $m < h$  - riešenie nieje optimálne. Chebyshevov odhad nezahŕňa dostatok prípadov. Algoritmus zahodí aktuálne riešenie. Do pôvodnej množiny sa vloží prípad s  $h$ -tou najväčšou absolútnou hodnotou rezidua a pomocou simplexovej metódy sa odstraní nevyhovujúci prípad. Táto výmena zaistí, že sa hodnota  $S$  zvýši. Z tejto podmnožiny sa znovu určí počet reziduí a prepočíta sa  $m$ .
- $m = h$  - riešenie splňuje podmienku optimálneho LMS a to je *feasible* riešenie. *Feasible* riešenie obsahuje minimálnu hodnotu  $S$  a to je požadovaný odhad LMS.
- $m > h$  - až na niektoré špeciálne prípady to nieje požadovaný LMS odhad.

Feasible algoritmus:

1. Vygenerovanie náhodného vzoru matice veľkosti  $p + 1$  a jeho použitie ho ako počiatočný vzor.
2. Aplikácia metódy dualného simplexového algoritmu s modifikovanou pivotážou k dosiahnutiu *feasible* riešenia
3. Pokiaľ ma feasible riešenie väčšiu hodnotu kritéria ako predošlé *feasible* riešenie, tak sa použije toto riešenie ako aktuálne optimálne.
4. Kroky 1-3 sa opakujú, pokiaľ nieje použitý dostatočný počet náhodných vzorov.

Vyčerpaním všetkých vzorov je možné dospieť k hodnote globalného minima LMS odhadu. Pravdepodobnosť nájdenia globalného optima je daná

$$P = 1 - [1 - \pi]^N \quad (3.13)$$

kde  $\pi$  je pomer všetkých *feasible* riešení a priemerného počtu pivotov  $E(P)$  na dosiahnutie *feasible* riešenia od počiatočného vzoru a  $N$  je počet náhodných vzorov.

Počet otestovaných vzorov je:

$$E = N(1 + E(P)) \quad (3.14)$$

[Hawkins–6] ukázal, že pre sústavu  $n = 40$  a  $p = 6$  pri odhade  $E(P) = 9$  a uvažovaní priaznivého pomeru  $\pi = 0.75$  je pravdepodobnosť nájdenia globálneho minima LMS rovná jednej (100%) už pri prehľadani  $N = 750$  náhodných vzorov. Pre nepriaznivý pomer  $\pi = 0.25$  je pre nájdenie globálneho minima potreba približne 1500 náhodných vzorov.

Ďalším z algoritmov aproximujúcich riešenie LMS popísali autori v [Olson–7]. Bol to algoritmus pre LMS regresiu s podmienkami založeným na náhodnom výbere vzorov sústavy s časovou zložitou  $O(n \log^2 n)$  pre 2-rozmerné sústavy a  $O(n^{d-1} \log^2(n))$  pre  $d$ -rozmerné sústavy, ktorý úspešne nájde odhad LMS s vysokou pravdepodobnosťou. Algoritmus pre dvojicu bodov s rôznymi  $x$ -ovými súradnicami a reziduom  $\mu$  majú priamky, ktoré náležia týmto bodom sklon  $\alpha$  v určitom rozmedzí.

$$\alpha \in \left( \frac{y_j - y_i - 2\mu}{x_j - x_i}, \frac{y_j - y_i + 2\mu}{x_j - x_i} \right) \quad (3.15)$$

K riešeniu LMS problému s obmedzujúcimi podmienkami je nutné nájsť minimálnu hodnotu rezidua mediánu  $\mu$  nájdením najnižšieho bodu v hladine množiny  $d - 1$  v  $n - 1$  kúžeľoch. Cieľom je určiť, či konkrétne  $\mu$  leží nad alebo pod správnou hodnotu. Výpočet je vykonaný s časovou náročnosťou  $O(n \log(n))$ . Pre nájdenie bodu je nutné určiť  $2n - 2$  hraníc kúžeľov, ktoré je potrebné zotriediť a prehľadať, tak aby bolo možné určiť počet sklonov prekrývajúcich každý region. Na začiatku výpočtu je definovaný čítač, ktorý sa inkrementuje v momente, keď sa dosiahne začiatok intervalu sklonu a dekrementuje pri dosiahnutí konca sklonu. Autor zvolil prístup náhodného výberu  $n$  vzorov z celkového počtu  $(n - 1)(n - 2)/2$  priesečiek kúžeľov, kde dve z týchto priesečiek ohraničujú správnu hodnotu. Priesečiky sú zoradené v závislosti na  $\mu$  v čase  $O(n \log^2(n))$  a v každom kroku je medián  $\mu$  testovaný v danom intervale a určí sa, či sa správna hodnota nachádza nad, pod alebo na testovanej hodnote. Následne sú určené všetky ostatné priesečiky ležiace medzi určenými hranicami. Výsledne vyhľadavanie je vykonané nad týmito priesečikmi a určí odhad LMS. Podľa [Olson–7] je pravdepodobnosť určenia aspoň jedného dobrého vzoru v  $t$  pokusoch

$$P = 1 - \left( \frac{n - m}{n} \right) \quad (3.16)$$

Metódu je možné rozšíriť pre  $d$ -rozmerný priestor, kde pravdepodobnosť nájdenia správneho riešenia je

$$P = \prod_{i=0}^{d-2} \left( \frac{m-i}{n} \right) \quad (3.17)$$

Ďalší zaujímavý spôsob optimalizácie určenia riešenia úlohy LMS navrhli [Karr–10]. Úloha spočívala v optimalizácii výberu vzorov s použitím genetického algoritmu. Genetický algoritmus je využívaný pre optimalizáciu problémov založených na prirodzenom výbere. Algoritmus opakovane modifikuje populáciu na základe individuálnych riešení. Je to vlastne vyhľadávacia technika, ktorá sa používa na modeloch vyskytujúcich sa v prírode. Genetický algoritmus sa používa pri optimalizačných úlohách, ktorých *objective* funkcia je nediferencovateľná, stochastická alebo nelineárna. Odhad LMS môže byť s použitím genetického algoritmu dosiahnutý omnoho rýchlejšie s pomerne vysokou presnosťou. Genetický algoritmus generuje populáciu ako reťazce, zhodnotí ich a následne použije toto zhodnotenie pre generovanie presnejšieho reťazca. Pre množinu dát veľkosti 900 bodov v dvoj-rozmernom priestore je pri použití genetického algoritmu vyhľadanie odhadu blízkeho optimu potrebné zhodnotenie 13 000 funkcií. Podľa autora dokáže genetický algoritmus nájsť presnejšie riešenie než algoritmus PROGRESS za rýchlejší čas.

[Vandev–12] navrhuje riešenie LMS pomocou minimalizácie funkcie iteratívnym spresňovaním neznámeho parametru v každom kroku. Extrém bude určovaný s použitím metódy minimalizácie gradientu. Gradient funkcie, ktorá ma byť minimalizovaná, je rovný gradientu funkcie  $f(x, l_k)$  resp.  $F(x)$  pre nejaké  $k$ .  $k$ -té poradie funkcie v zoradenej množine ale nieje známe. Preto musia byť určené všetky funkcie a výsledky uložené pre ďalší výpočet. Ďalším problémom je, že globálna funkcia nieje konvexná a má mnoho lokálnych extrémov, čo nezaručuje konvergenciu ku globálnemu minimu. Autor preto navrhuje určiť gradient približne a to vyhľadaním v náhodnom výbere funkcií. Algoritmus preto nezaručuje určenie globálneho minima.

Gradient vektora neznámych parametrov je vo viacnásobnej regresii daný ako

$$\frac{df(x, l)}{dx} = -\text{sign}((\ell^0 - \ell l)(x - x^0))(x l, 1) \quad (3.18)$$

kde  $x_i$  je vektor neznámych parametrov v  $i$ -tom kroku

$$x_{i+1} = x_i - \gamma_i \left( \frac{\nabla(F(x_i))}{\|\nabla(F(x_i))\|} \right) \quad (3.19)$$

kde  $\gamma_i$  splňuje podmienku  $\sum_{i=1}^{\infty} \gamma_i^2 < \infty$  a  $\sum_{i=1}^{\infty} \gamma_i = \infty$

```

1 n = A_rows
2 m = A_cols
3 x[i] = 0 for i=1...m
4 prop = 6
5 delta = 10
6 iter = 100
7
8 for( k=1; k<iter; k++ )
9 {
10  for( i=1; i<10; i++ )
11  {
12    J(i) = rand(1...n) // 10 - size of random subsample
13  }
14
15  for( o=1; o<10; o++ )
16  {
17    res = l[J] - A[J,o] * b
18    ares = abs(res)
19    sort(&ares)
20
21    for( i=1; i<10; i++ )
22    {
23      ti = trans(A[J[i],o]) * sign(res[i])
24      theta = sqrt(trans(ti)*ti)
25      gamma = delta/k
26      x[i] = x[i] + gamma * ti[i]/theta
27    }
28 }

```

Kód 3.1: Symbolicky zápis algoritmu určenia odhadu LMS minimalizáciou funkcie iteratívnym spresňovaním

Pre jednoduchú lineárnu regresiu s 1000 observáciami porušenými z 30-tich% odľahlými meraniami bol rozumný odhad určený po 150 iteráciach pre  $\delta = 10$ . Algoritmus je veľmi efektívny, zachováva hodnotu bodu zvratu a je možné ho použiť k prvému kroku robustného odhadu v multi-dimenzionálnych prípadoch. V tomto kroku je detekovaná väčšina odľahlých meraní.

Na princípe iteratívneho určovania gradientu je v [Rusiecki–13] popísany algoritmus, ktorý rieši úlohu LMS pomocou stratégie spätného rozdeľovania zmien *performance* funkcie s ohľadom na váhy. Problém je riešený algoritmom RBP (*resilient backpropagation*), ktorý eliminuje efekt pomalej konverencie pre veľmi nízke hodnoty gradientu spôsobené plochou oblasťou. Autor navrhuje určiť zmeny rovnako ako pre metódu MSE (*least mean squared error*), ktorá je optimálna pre dáta, ktoré neobsahujú odľahlé merania s normálnym rozdelením chýb a určené zmeny potom použiť k minimalizácii reziduí LMS.

$$E_{MSE} = \frac{1}{n} \sum_{i=1}^n v_i^2 \quad (3.20)$$

Každý krok obsahuje výpočet gradientu pomocou RBP algoritmu a následne zhodnotenie zmeny odhadu LMS. Váhy sú určené ako

$$w(k) = \delta(k) \times \text{sign}(\nabla(E_{MSE})) \quad (3.21)$$

kde  $k$  je aktuálny krok výpočtu a  $\delta$  je hodnota, ktorá sa mení v každom kroku výpočtu  $\delta(k+1) = \frac{\delta(k)}{a}$ , kde  $a$  je tzv. klesajúci faktor. Kroky výpočtu LMS použitím RBP algoritmu sú teda:

- použi spätné rozdelenie k výpočtu zmien *performance* funkcie odhadu MSE
- aktualizuj váhy
- modifikuj delta
- určí hodnotu mediánu v aktuálnom kroku
- pokiaľ je medián minimalizovaný na určitú hodnotu alebo počet iterácií dosiahol maxima, tak ukonči výpočet. V opačnom prípade pokračuj od začiatku.

Autor ďalej navrhuje v každom kroku redukovať počet dát podozrivých z odľahlosti výpočtom štandardnej robustnej odchýlky 3.5. Práh je zvolený ako  $v_i^2 \geq 2,5\sigma_v^2$ .

### 3.1.4 Paralelné algoritmy pre výpočet odhadu LMS

V súčasnej dobe väčšina počítačov obsahuje procesor s viacnásobným počtom jadier. Paralelizácia časovo náročných výpočtov je pomerne bežná záležitosť. Preto sa ponúka myšlienka, či výpočet LMS nieje možné riešiť na viacerých jadrách súčasne a tým celý výpočet paralelizovať. Jedným z prvých článkov zaoberajúcich sa touto tematikou publikovali autori v [Chong-wei–14]. Autori navrhujú paralelizáciu pôvodného algoritmu navrhnutého podľa Roussew a *sweep-line* algoritmu popísaného v kapitole 3.1.1. Majme teda symbolický zápis pôvodného Roussew algoritmu.

```

1 for( i=1; i<=n-1; i++ )
2 {
3   for( j=i+1; j<n; j++ )
4   {
5     make_a_line(data_ary[i], data_ary[j], &alpha, &beta);
6
7     for( k=1; k<=n; k++ )
8     {
9       med_ary[k] = calc_residual(data_ary[k], alpha, beta);
10      sort(med_ary);
11      median_val = med_ary[mid];
12      optimal_med = find_opt(median_val);

```

```
13     }  
14     }  
15 }
```

### Kód 3.2: Roussew LMS algoritmus určený pre paralelizáciu

Pre paralelizáciu Roussew algoritmu autor navrhuje dve možné spôsoby a to v prvom prípade:

- procesor je rozdelený na jeden host (hlavné) vlákno a  $p$  nodov (vlákien).
- rovnomerne rozdeliť  $n$  dát na  $p$  procesorov resp. jadier  $n/p$
- všetky procesory počítajú nezávisle  $n/p$  rezidií
- triedenie prebieha na  $n/p$  reziduách a následne prebehne operácia triedenia na  $p$  roztriedených množinách
- host spracováva operácie dvoch vonkajších cyklov a vnútorný cyklus sa rozdelí medzi  $p$  nodov

V druhom prípade autor navrhuje:

- rovnomerne rozdeliť  $n$  dát na  $p$  nodov  $n/p$
- vytvoriť  $n/p$  priamok v  $p$  skupinách
- vyriešiť všetky reziduá pre každú skupinu priamok
- roztriediť reziduá pre každú skupinu
- dve vonkajšie cykly sú rozdelené medzi  $p$  nodov
- vnútorný cyklus a triedenie nieje paralelizované, ale vykonáva sa pre každú skupinu

Z toho vyplýva, že v druhom prípade máme po výpočte  $p$  riešení s  $p$  lokálnymi extrémami. Z týchto lokálnych extrémov potom host vyberie extrém globálny tzn. riešenie s minimálnym mediánom. V prvom navrhovanom riešení sa synchronizuje paralelné triedenie pre každú skupinu dát. Režijné náklady na synchronizáciu nodov s tak príliš vysokou granularitou prevyšujú výhodu paralelného výpočtu. Druhý algoritmus znižuje počet synchronizácií a zvyšuje granularitu. Každý z nodov vykonáva rovnaký počet operácií asynchrónne. Tým pádom sa zrýchlenie znásobí proporčne cez všetky nody.

V prípade *sweep-line* algoritmu každý výpočtový krok potrebuje globálnu informáciu pre vykonanie ďalšieho výpočtového kroku. Z toho vyplýva, že je nemožné

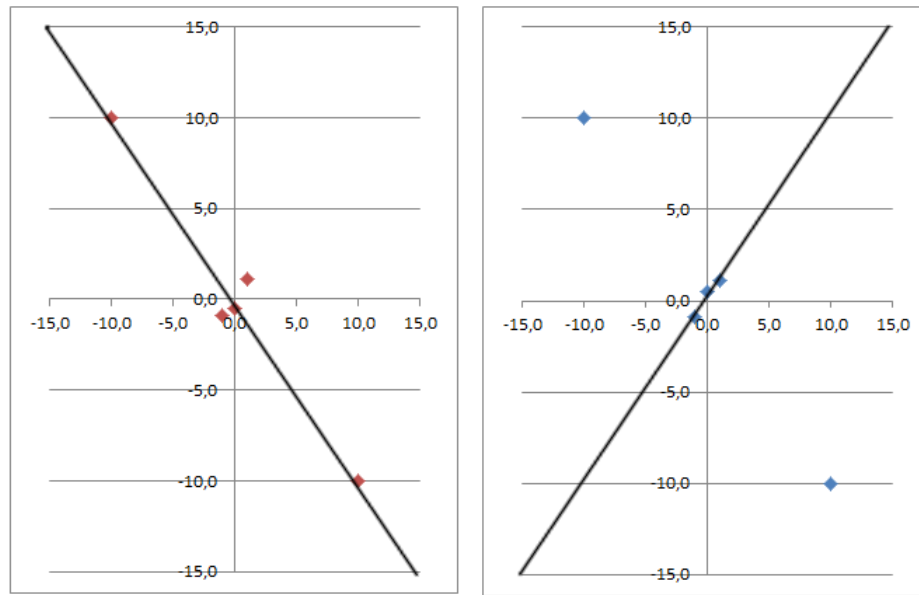


staticky rozdeliť data do  $p$  množín. Riešenie ale spočíva v znalosti dátových štruktúr popísaných v kap. 3.1.1. Jedna dátová štruktúra ukláda  $x$ -ové súradnice a druhá dátová štruktúra súradnice  $y$ . Kedykoľvek nájdeme priesečník dvoch priamok, tak sa priamky vymenia. Operácia udržiava priamky zoradené podľa ich súradnic  $y$ . Pokiaľ je teda rovina rozdelená do niekoľkých skupín, tak môžeme použiť  $x$ -ovú súradnicu na hranici vľavo každej skupiny k nájdeniu  $y$ -zoradených priamok na tejto hranici. Potom má každá skupina svoj priesečník na hranici vľavo a celá skupina zľava môže byť vymenená so skupinou na hranici vpravo. Toto je základná myšlienka paralelizácie *sweep-line* algoritmu. Výkon paralelného *sweep-line* algoritmus je zrovnateľný s paralelným Roussew algoritmom. Sweep-line algoritmus sa používa pre riešenie LMS v rovine, a preto nieje použiteľný pre účely v tejto disertačnej práci. Podrobný popis paralelného sweep-line algoritmu je uvedený v [Chong-wei-14].

## 3.2 Prípady zlyhania LMS odhadu

V práci [Shather-4] autor ukázal, že metóda LMS je síce veľmi odolná proti hrubým chybám, ale na druhú stranu v niektorých prípadoch môžu malé zmeny lokalizované v centre dát spôsobiť výraznú zmenu odhadu. LMS kritérijna funkcia môže byť počítaná zo vzorov, ktoré majú polovičnú veľkosť, teda ktoré obsahujú  $(n/2) + 1$  bodov. Táto metóda sa nazýva *half-sample selection* a je často používaná pri konštrukcii odhadu s 50% bodom zvratu. LMS odhad je v tomto prípade priamka rozdeľujúca pás ohraničený dvoma najbližšími priamkami v ktorom leží 50% bodov. Pokiaľ budeme mať dve polovičné vzory so zhruba rovnakou hodnotou kritérijnej funkcie, tak i malé zmeny v niektorých dátach môžu spôsobiť zásadnú zmenu odhadu. To nastane v prípade, že dve páry rovnobežných priamok obsahujú každý z nich polovicu dát a šírky týchto dvoch pásov sú takmer totožné. Problém nieje v odľahlých hodnotách ani vplyvových bodoch uvedených v kapitole 4.6, ale naopak v malých posunoch lokalizovaných v centre dát. Problém je ukazaný na jednoduchom príklade.

[Cook-8] ukázali, že použitie robustných metód v prípade absencie odľahlých meraní môže byť skôr prekážkou než výhodou pri identifikácii odchýlok od linearity. Výhoda vysokého bodu zvratu pri použití robustných metód môže byť za cenu potenciálnej straty časti pôvodnej informácie o investigovaných dátach. V prípade, že sa veľká časť informácie nachádza v dátach s veľkými reziduami a tieto dáta metóda LMS a taktiež napr. LTS ignoruje, tak nieje možné očakávať, že táto metóda bude dobre fungovať. V [Cook-8] je dokázané, že existujú prípady kde LMS odhad zlyhá a to práve vtedy, keď porušuje Li-Duanov teorem. To znamená, že merané hodnoty nemajú eliptický symetrické rozdelenie. Pre metódy s vysokým bodom zvratu môže byť v týchto prípadoch test odchýlok od linearity výrazne horší než u klasického



Obrázok 3.1: Zmena odhadu spôsobená malým rozdielom v centre dát

MNČ a to z dôvodu, že LMS nieje tzv. Fisher-konzistentné. LMS odhad v týchto prípadoch môže zlyhať aj pri použití veľkých vzorov. Na elementárnom príklade bolo ukázané, že oproti štandardnému MNČ potom môže mať LMS opačný lineárny trend a taktiež reziduá sú v prípade LMS omnoho väčšie ako u MNČ. Príčinou je, že miesto odčítania reálneho lineárneho trendu z dát pridá LMS odhad irelevantnú lineárnu podmienku, ktorá znehodnotí pôvodnú štruktúru dát. Pri LMS odhade je teda nutné predpokladať, že je splnená podmienka lineariry a normality.

## 4 Robustné metódy vyrovnania meraných veličín

Existuje mnoho ďalších spôsobov odhadu neznámych parametrov potenciálne kontaminovaných väčším množstvom odľahlých meraní. V tejto kapitole sú popísané robustné metódy, ktoré boli použité pre porovnanie výsledkov určených metódou LMS. V prvej sekcii je popísaný  $L_p$ -norma odhad, ktorý nieje citlivý na odľahlé merania v ose X, ale neplatí to pre prípad vplyvových bodov v ose Y.  $L_p$ -norma odhad je robustný len pre niektoré hodnoty  $p$ . Ďalšou metódou sú metódy M-odhadov, ktorých výpočet je oproti metóde LMS omnoho efektívnejší, ale na druhú stranu majú nižší bod zvratu. V nasledujúcich podkapitolách je taktiež popísaná metóda LTS pomerne podobná metóde LMS s tým rozdielom, že sa minimalizuje suma štvorcov reziduí nejakým spôsobom obmedzených. Poslednou s popísaných metód je metóda RANSAC založená obdobne ako LMS na výbere náhodných vzorov observácií. Princíp výpočtu odhadu neznámych parametrov metódou RANSAC je čiastočne podobný metóde LMS.

### 4.1 Teoretický základ

Majme observácie, ktoré obsahujú informácie o neznámych parametroch a sú uložené vo vektore  $\ell$ . Každé meranie reprezentuje vzor z množiny hodnôt priestoru pravdepodobnosti. Meranie patrí do nejakého rozdelenia pravdepodobnosti a parametre definované pre toto rozdelenie sú neznáme parametre obsiahnuté vo vektore  $\mathbf{x} \in n \times 1$ . Matica  $\mathbf{A} \in m \times n$  je matica známych koeficientov. Pridaním náhodného vektora chýb  $\epsilon \in m \times n$  merania  $\ell \in m \times 1$  získame konzistentný systém

$$\ell = \mathbf{A}\mathbf{x} + \epsilon \quad (4.1)$$

a reziduá sú definované ako rozdiel odhadu od skutočných meraných hodnôt

$$v = \hat{\ell} - \ell \quad (4.2)$$

### 4.2 $L_p$ -norma odhady

Máme funkciu  $L_p$ -normového odhadu, ktorá sa bude minimalizovať

$$\|\epsilon\|_p = \left( \sum_{i=1}^n |\epsilon_i|^p \right)^{\frac{1}{p}} \quad (4.3)$$

kde  $\|\epsilon\|_p$  je generalizácia  $L_p$ -normy vektora chýb  $\epsilon$ . Minimum získame deriváciou funkcie podľa neznámych parametrov a položením derivácie rovno 0.

$$\frac{\partial \|\epsilon\|_p}{\partial x_k} = \left( \sum_{i=1}^n |\epsilon_i|^p \right)^{\frac{1}{p}-1} \left( \sum_{i=1}^n |\epsilon_i|^{p-2} \epsilon_i a_{ik} \right) \quad (4.4)$$

Pokiaľ je prvý člen nenulový položíme druhý člen rovný nule

$$\sum_{i=1}^n |v_i|^{p-2} v_i a_{ik} = 0 \quad k \in 1, \dots, d \quad (4.5)$$

kde  $d$  je počet neznámych parametrov. V prípade, že sa  $p = 1$   $L_1$ -norma odhad minimalizuje sumu absolútnych hodnôt rezidui.  $L_1$  odhad bol jedným z prvých robustných odhadov. Bod zvratu metódy  $L_1$  je rovnako ako u klasickej metódy najmenších štvorcov 0%. Metóda  $L_1$  je však veľmi robustná v prípade, že observácie obsahujú odľahlé merania v ose X (outliers). Pokiaľ je ale súbor meraní kontaminovaný vplyvovými bodmi, tak je odhad parametrov znehodnotený obdobne ako u MNČ. Pokiaľ je  $p = 2$ ,  $L_p$ -norma odhad je ekvivalentný metóde najmenších štvorcov [Koch–22].

Vektor neznámych parametrov  $\hat{\mathbf{x}}$  odhadneme pomocou  $d$  nezávislých observácií  $\ell$  pomocou  $L_1$ -normy.

$$\sum_{i=1}^n \frac{\hat{\mathbf{x}} - \ell_i}{|\hat{\mathbf{x}} - \ell_i|} = 0 \quad (4.6)$$

Suma bude rovná nule práve vtedy, keď bude počet kladných a záporných hodnôt (jednotiek) rovnaký. Pokiaľ sú merané dáta zoradené vzostupne  $\ell_1 \leq \ell_2 \leq \dots \leq \ell_n$ ,  $L_1$ -norma odhad sa nazýva medián a odhad je daný  $\hat{\mathbf{x}} = \frac{\ell_{n+1}}{2}$ , teda pre čísla nepárne je riešenie  $\ell_{n/2} \leq \hat{x} \leq \ell_{n/2+1}$  a pre čísla párne platí  $\hat{\mathbf{x}} = \frac{1}{2}(\ell_{n/2} + \ell_{n/2+1})$ . Medián vráti teda správny výsledok aj pri kontaminácii 50% merania. Medián je preto považovaný za obzvlášť robustnú metódu.

### 4.3 M-odhady

Súčasťou tejto práce je porovnanie výsledkov LMS s podobnými robustnými metódami. Za podobné robustné metódy je možné považovať M-odahdy. M-odahdy, rovnako ako metóda LMS, si zachovávajú svoju funkčnosť aj v niektorých prípadoch nesplnenia požiadavku na normálne rozdelenie chýb observácií. M-odhady sú založené na metóde maximálnej vierohodnosti (*maximum likelihood*). Splnenie podmienky funkčnosti aj v prípadoch, keď chyby merania nespádajú do normalného rozdelenia, je dosiahnuté nahradením funkcie normalného rozdelenia vhodnejšou odhadovou funkciou ( $\rho$ ). Minimalizáciou odhadovej funkcie teda jej deriváciou získame

vplyvovú funkciu ( $\psi$ ). M-odhady majú na rozdiel od metódy LMS nižší bod zvratu. To znamená, že presnosť odhadu neznámych parametrov je nižšia resp. odhad neznámych parametrov je znehodnotený pri nižšom počte kontaminovaných meraní. Odhad neznámych parametrov v prípade M-odhadov môže značne ovplyvniť observácia vysokou vplyvovou hodnotou (kap. 4.6.1). Podľa [Ruckstuhl–37] je bod zvratu M-odhadov  $1/n$ , kde  $n$  je počet neznámych parametrov.

Chceme minimalizovať nasledujúcu funkciu maximálnej vierohodnosti *score function*

$$\sum_{i=1}^m -\ln g(\ell_i, \mathbf{x}) \quad (4.7)$$

Pre deriváciu robustných odhadov je nutné nahradiť funkciu uvedenú vyššie funkciou viac vyhovujúcou

$$\sum_{i=1}^m \rho(\ell_i, g_i(\mathbf{x})) \quad (4.8)$$

kde  $g_i(\mathbf{x})$  je funkciou neznámych parametrov. Deriváciou funkcie získame

$$\sum_{i=1}^m \psi(\ell_i, \mathbf{x}) \frac{\partial g_i(\mathbf{x})}{\partial \mathbf{x}}, \quad \text{kde} \quad \psi(\ell_i, \mathbf{x}) = \frac{\partial \rho}{\partial g_i} \quad (4.9)$$

počítame s observáciami, ktoré patria do normálneho rozdelenia

$$f(\ell_i) = \frac{1}{\sigma_0 \sqrt{2\pi}} e^{-(\ell_i - \mathbf{a}_i^T \mathbf{x})^2 / 2\sigma_0^2} \quad (4.10)$$

a score funkcia je teda daná

$$\rho(\ell_i, g_i(\mathbf{x})) = \frac{(\mathbf{a}_i^T \mathbf{x} - \ell_i)^2}{2\sigma_0^2} = \frac{1}{2} \left( \frac{\epsilon_i}{\sigma_0} \right)^2 \quad (4.11)$$

Odhadom skutočných chýb sú normované opravy  $\hat{v}_i = \frac{v_i}{\sigma_i}$ , kde  $\sigma_i$  je smerodajná odchýlka merania. Praktický výpočet M-odhadov je riešený ako štandardná metóda najmenších štvorcov s tým rozdielom, že sú k meraniam priradené váhy  $w$  v závislosti na veľkosti normovanej opravy a odhadovej funkcie. Odhad neznámych parametrov je výsledkom  $n$ -iterácii odhadu MNČ s oakovane určenými váhovými koeficientami.

$$\begin{aligned} \mathbf{x}^{(0)} &= (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \ell \\ &\dots \\ \mathbf{x}^{(j+1)} &= (\mathbf{A}^T \mathbf{W}^{(j)} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{W}^{(j)} \ell \end{aligned} \quad (4.12)$$

kde  $\mathbf{W}$  je diagonálna matica váhových koeficientov.

### 4.3.1 Určenie váhových koeficientov

V tejto podkapitole sú uvedené základné vzťahy pre určenie váhových koeficientov pre vybrané robustné odhady.

#### Huberov odhad:

Voľba konštanty  $c$  závisí na percentuálnom množstve kontaminovaných meraní  $o$  z celkového množstva observácií  $n$  a to

$$c = \begin{cases} 1,5 & \text{pre } o \leq 4\% \\ 2 & \text{pre } o < 1\% \end{cases} \quad (4.13)$$

Definícia odhadovej a vplyvovej funkcie a vzťahy pre určenie váhových koeficientov Huberovho odhadu

$$\rho(\hat{v}) = \begin{cases} \frac{1}{2}\hat{v}^2 \\ c|\hat{v}| - \frac{1}{2}c^2 \end{cases}, \psi(\hat{v}) = \begin{cases} \hat{v} \\ c \operatorname{sign}(\hat{v}) \end{cases}, w(\hat{v}) = \begin{cases} 1 & \text{pre } |\hat{v}| \leq c \\ \frac{c}{|\hat{v}|} & \text{pre } |\hat{v}| > c \end{cases} \quad (4.14)$$

#### Modifikovaný huberov odhad:

Definícia odhadovej a vplyvovej funkcie a vzťahy pre určenie váhových koeficientov modifikovaného Huberovho odhadu

$$\rho(\hat{v}) = \begin{cases} \frac{1}{2}\hat{v}^2 \\ c|\hat{v}| - \frac{1}{2}c^2 \\ cb - \frac{1}{2}b^2 \end{cases}, \psi(\hat{v}) = \begin{cases} \hat{v} \\ c \operatorname{sign}(\hat{v}) \\ 0 \end{cases}, w(\hat{v}) = \begin{cases} 1 & \text{pre } |\hat{v}| \leq c \\ \frac{c}{|\hat{v}|} & \text{pre } c < |\hat{v}| \leq b \\ 0 & \text{pre } |\hat{v}| > b \end{cases} \quad (4.15)$$

kde konštanty sú volené ako  $a = 2, b = 3$ .

#### Hampelov odhad:

Definícia odhadovej a vplyvovej funkcie a vzťahy pre určenie váhových koeficientov Hampelovho odhadu

$$\rho(\hat{v}) = \begin{cases} \frac{1}{2}\hat{v}^2 \\ a|\hat{v}| - \frac{1}{2}a^2 \\ ab - \frac{1}{2}a^2 + \frac{1}{2}a(c-b) \left[ 1 - \left( \frac{c-|\hat{v}|}{c-b} \right)^2 \right] \\ ab - \frac{1}{2}a^2 + \frac{1}{2}a(c-b) \end{cases}, \psi(\hat{v}) = \begin{cases} \hat{v} \\ a \operatorname{sign}(\hat{v}) \\ a \frac{c-|\hat{v}|}{c-b} \operatorname{sign}(\hat{v}) \\ 0 \end{cases},$$

$$w(\hat{v}) = \begin{cases} 1 & \text{pre } |\hat{v}| < a \\ \frac{a}{|\hat{v}|} & \text{pre } a \leq |\hat{v}| < b \\ \frac{c-|\hat{v}|}{c-b} \frac{1}{|\hat{v}|} & \text{pre } b \leq |\hat{v}| < c \\ 0 & \text{pre } |\hat{v}| \geq c \end{cases} \quad (4.16)$$

kde konštanty sú volené ako  $a = 2, b = 4, c = 8$ .

### Talvarov odhad:

Definícia odhadovej a vplyvovej funkcie a vzťahy pre určenie váhových koeficientov Talvarovho odhadu

$$\rho(\hat{v}) = \begin{cases} \frac{1}{2}\hat{v}^2 \\ \frac{1}{2}a^2 \end{cases}, \psi(\hat{v}) = \begin{cases} \hat{v} \\ 0 \end{cases}, w(\hat{v}) = \begin{cases} 1 & \text{pre } |\hat{v}| \leq a \\ 0 & \text{pre } |\hat{v}| > a \end{cases} \quad (4.17)$$

kde konštanta je volená ako  $a = 2,795$ .

### Tukeyho biweight odhad

Definícia odhadovej a vplyvovej funkcie a vzťahy pre určenie váhových koeficientov Tukeyho biweight odhadu

$$\rho(\hat{v}) = \begin{cases} \frac{1}{6}a^2 \left(1 - \left[1 - \left(\frac{|\hat{v}|}{a}\right)^2\right]^3\right) \\ \frac{1}{6}a^2 \end{cases}, \psi(\hat{v}) = \begin{cases} |\hat{v}| \underbrace{\left(1 - \left(\frac{\hat{v}}{a}\right)^2\right)^2}_d \\ 0 \end{cases}, \quad (4.18)$$

$$w(\hat{v}) = \begin{cases} d & \text{pre } |\hat{v}| \leq a \\ 0 & \text{pre } |\hat{v}| > a \end{cases}$$

kde konštanta je volená ako  $a = 4,685$ .

### Gemanov odhad:

Definícia odhadovej a vplyvovej funkcie a vzťahy pre určenie váhových koeficientov Gemanovho odhadu

$$\rho(\hat{v}) = \frac{\hat{v}^2}{2(1 + \hat{v}^2)}, \psi(\hat{v}) = \frac{\hat{v}^2}{(1 + \hat{v}^2)}, w(\hat{v}) = \frac{1}{(1 + \hat{v}^2)} \quad (4.19)$$

### Fair odhad:

Definícia odhadovej a vplyvovej funkcie a vzťahy pre určenie váhových koeficientov Fair odhadu

$$\rho(\hat{v}) = \frac{a^2 |\hat{v}|}{a - \log\left(1 + \frac{|\hat{v}|}{a}\right)}, \psi(\hat{v}) = \frac{|\hat{v}|}{1 + \frac{|\hat{v}|}{a}}, w(\hat{v}) = \frac{1}{1 + \frac{|\hat{v}|}{a}} \quad (4.20)$$

kde konštanta je volená ako  $a = 2,985$ .

#### **$L_1$ norma odhad:**

Definícia odhadovej a vplyvovej funkcie a vzťahy pre určenie váhových koeficientov  $L_1$  norma odhadu

$$\rho(\hat{v}) = |\hat{v}|, \psi(\hat{v}) = \frac{\hat{v}}{|\hat{v}|}, w(\hat{v}) = \frac{1}{|\hat{v}|} \quad (4.21)$$

#### **$L_1L_2$ norma odhad:**

Definícia odhadovej a vplyvovej funkcie a vzťahy pre určenie váhových koeficientov  $L_1L_2$  norma odhadu

$$\rho(\hat{v}) = 2 \left( \sqrt{1 + \frac{\hat{v}^2}{2}} - 1 \right), \psi(\hat{v}) = \frac{\hat{v}}{\sqrt{1 + \frac{\hat{v}^2}{2}}}, w(\hat{v}) = \frac{1}{\sqrt{1 + \frac{\hat{v}^2}{2}}} \quad (4.22)$$

## 4.4 Least Trimmed Squares - LTS

Ďalšia robustná metóda s vysokým bodom zvratu, ktorú predstavil Rousseew v roku 1984, sa nazýva LTS (Least trimmed squares). Obdobne ako metóda LMS, tak aj LTS má bod zvratu 50%. Ako už názov napovedá, metóda LTS minimalizuje sumu štvorcov oprav súboru merania orezaného (trimmed) o nejaký určitý počet dát. Konštanta určujúca množstvo dát, ktoré sa majú z meraní odstrániť, je pevne daná. Opakovaným výpočtom MNČ sa zo súboru meraní odstraní rovnaké množstvo meraní s najväčšími vyrovnanými opravami. Rezidua je teda nutné v každom kroku roztriediť podľa veľkosti. V prípade veľkých množín observácií spočíva vyhľadanie odhadu neznámych parametrov pomocu metódy LTS rovnako ako u metódy LMS vo výbere vozorov z pôvodného súboru meraní. Podľa [Rousseew-48] je riešenie metódy LTS všeobecne nepatrne pomalšie ako riešenie metódy LMS. Najväčšej robustnosti sa dosiahne v prípade, že  $h$  je približne  $n/2$ . Minimalizácia reziduí pomocou metódy LTS je teda daná

$$\sum_{i=1}^h v_{i:n}^2 \rightarrow \min \quad (4.23)$$



pre  $v_{1:n} \leq \dots \leq v_{n:n}$ . Odhad veľkosti skutočnej chyby meraní je daný obdobne ako u LMS

$$\hat{\sigma} = C \sqrt{\frac{1}{n} \sum_{i=1}^h v_{i:n}^2} \quad (4.24)$$

kde  $C$  je korekčný faktor použitý pre dosiahnutie konzistencie rozdelenia chýb. Podľa [Rousseew–48] je rýchlosť konvergencie LTS vyjadrená ako  $n^{-1/2}$  naproti extrémne pomalej konvergencii  $n^{-1/3}$  v prípade metódy LMS. Nevýhodou LTS je ale nutnosť triedenia s časovou náročnosťou  $O(n \log n)$  v prípade použitia qucik sort algoritmu naproti vyhľadaniu mediánu s časovou zložitnosťou  $O(n)$ .

## 4.5 Metóda RANSAC

RANSAC, RANdom SAMple Consensus, je metóda odvodená z modelu založeného na lineárnej regresii, ktorej data obsahujú značné množstvo šumu. Riešenie RANSAC podobne ako LMS, je postavené na iteratívnom výbere vzorov z modelu obsahujúceho značné množstvo odľahlých meraní. Výsledkom metódy je nájdenie takého vzoru, ktorý obsahuje maximálny počet meraní, ktorých vzdialenosť od odhadovaného modelu nepresahuje definovaný prah. Z geometrického hľadiska v dvojrozmernom priestore je možné úlohu definovať ako vyhľadanie maximalného počtu bodov obsiahnutých v páse definovaného priamkou určenou dvoma bodmi a šírkou určenou veľkosťou prahovej hodnoty. Metóda RANSAC sa často používa v obore počítačového videnia, korekcie obrazu či chemii pri určení zloženia materiálu [Kaspi–42]. V obore geodézie je možné ju uplatniť vo fotogrametrii či laserovom skenovaní. Aj keď táto metóda je veľmi rýchla a robustná, tak jej výsledkom je iba získanie globálneho optima [Traa–41].

RANSAC algoritmus sa skladá prakticky z dvoch krokov a to výberu minimalného vzoru a testu v iteratívnom procese. V prvom kroku je teda náhodne vybraný vzor dát s minimálnym počtom meraní tzv. MSS (minimal sample set). MSS je teda celkový počet dát potrebných na určenie parametrov modelu. Podstatou výberu minimalného vzoru je znížiť pravdepodobnosť kontaminácie vzoru odľahlými meraniami. Na tých dátach je určený odhad modelu. Všetky ostatné dáta su potom testované vzhľadom k tomuto modelu. Pokiaľ teda minimálny vzor obsahuje odľahlé merania, ostatné data nebudú patriť do tohto modelu. V ďalšom kroku je zase vybraná náhodna minimálna množina a proces sa opakuje. RANSAC teda interaktívne testuje, či sú observácie vstupného vzoru náležia hypotetickému modelu. Test je založený na určení prahu  $T$ , ktorý je definovaný ako maximálna vzdialenosť medzi skúmaným bodom a hypotetickým modelom [Kulkarni–40]. Výsledkom odhadu je

model, ktorý obsahuje dostatočný počet bodov patriacich do množiny určenej hypotézou. Výsledná množina sa nazýva CS (*consensus set* - množina zhody). Model je daný ako

$$y = ax + b \quad (4.25)$$

$$f(l, x) = \min(\text{dist}(l, \hat{\ell})) \quad (4.26)$$

CS je definovaný ako

$$S(x) = x \in X : f(l, x) \leq \delta \quad (4.27)$$

kde  $\delta$  je práh definovaný na základe hypotézy.

Podľa [Fischler-49] je počet iterácií na dosiahnutie optimálneho modelu daného pravdepodobnosťou, či minimálnej množiny obsahujúcej odľahlé merania.

$$L \geq \frac{\log P_f}{\log(1 - (P_g)^h)} \quad (4.28)$$

kde  $h$  je nutný počet meraní.  $P_g$  je pravdepodobnosť výberu neodľahlého merania a teda  $(P_g)^h$  je pravdepodobnosť, že minimálna množina nebude zaťažená odľahlými meraniami.  $P_f$  je pravdepodobnosť, že algoritmus skončí bez nájdenia optimálneho odhadu resp. pravdepodobnosť, že množina bude obsahovať odľahlé merania vo všetkých krokoch  $L$  výpočtu.

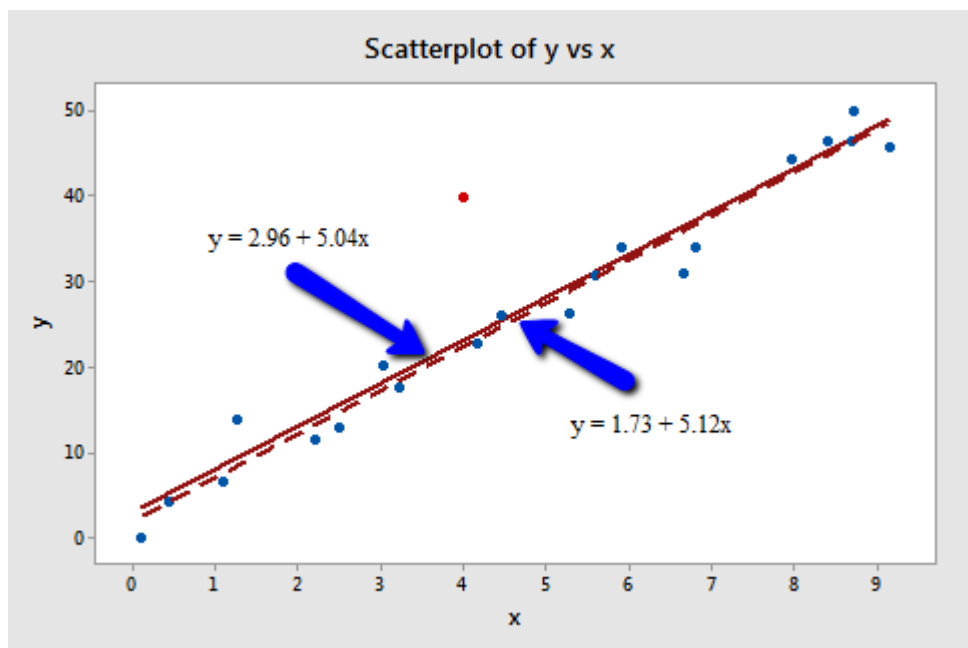
$$P_f = (1 - (P_g)^h)^L \quad (4.29)$$

## 4.6 Odľahlé body a ich vplyv na odhad neznámych parametrov

Predpokladajme, že pracujeme s jednoduchým lineárnym modelom. Majme množinu bodov, ktorá obsahuje chyby zahrnuté do normálneho rozdelenia, a bod resp. nejakú ďalšiu množinu bodov, ktorá je od prvej množiny nejakým spôsobom odľahlá. Body odľahlej množiny môžu odhad neznámych parametrov ovplyvniť rozdielnym spôsobom. Tieto body môžeme rozdeliť do dvoch skupín a to na body odľahlé v ose  $Y$  (*leverage points*) a odľahlé v ose  $X$  (*outliers*). Bod odľahlý v oboch smeroch sa nazýva vplyvový bod. Odľahlé meranie *leverage* je také meranie, ktorého určená hodnota je veľmi odlišná od odhadu založeného na danom modele, teda v prípade jednoduchej lineárnej regresie v dvojrozmernom kartézskom systéme hodnota súradnice  $y$  nespadá do trendu množiny merania. Na druhú stranu odľahlý bod *outlier*

nasleduje trend množiny meraní, ale má extrémnu hodnotu súradnice  $x$ . Bod, ktorý značne ovplyvní odhad neznámych parametrov lineárnej regresie, je potenciálne odľahlý v oboch osách s tzv. vysokým vplyvom.

Na obrázku 4.1 je odľahlý *leverage* bod v ose  $Y$  znázornený červenou farbou, pre-rušovanou čiarou je znázornený odhad lineárnej regresie, ktorá nezahrňuje odľahlý bod do výpočtu. Na druhú stranu plná čiara znázorňuje odhad lineárnej regresie zahrňujúci odľahlý bod. Z obrázku je patrné, že odľahlý nemá takmer žiaden vplyv na výsledný odhad



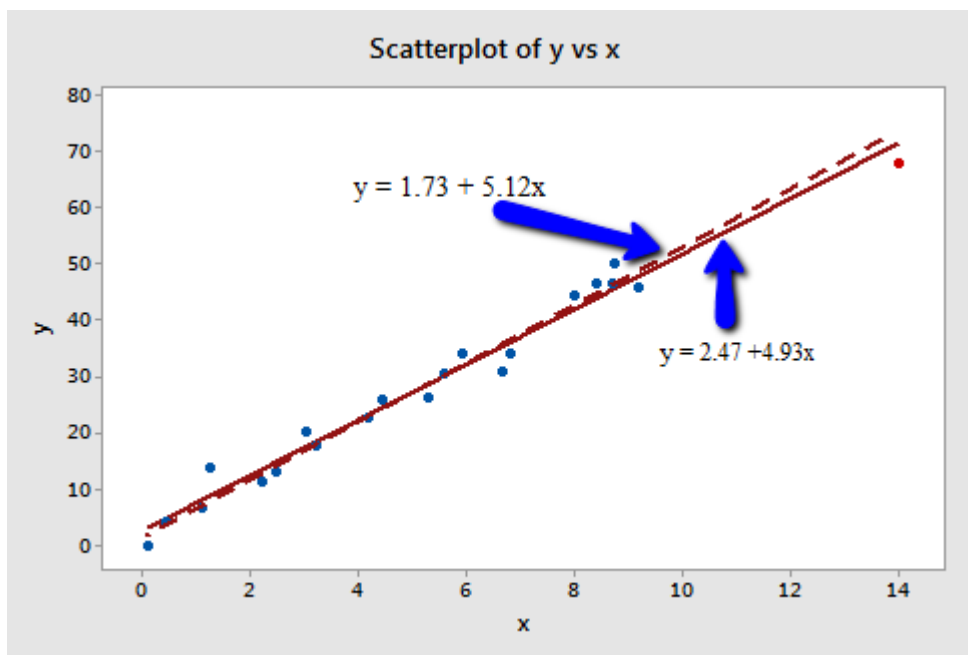
Obrázok 4.1: Znázornenie odľahlého bodu v ose  $Y$  [Outliers-53]

Na obrázku 4.2 je červenou farbou znázornený odľahlý *otlier* bod v ose  $X$ , ktorý nasleduje trend množiny ale má extrémnu hodnotu súradnice  $x$ . Z obrázku je patrné, že bod nemá na výsledok lineárnej regresie takmer žiaden vplyv.

Obdobne ako v predošlých dvoch prípadoch je na obrázku 4.3 znázornený červenou farbou vplyvový bod, ktorý má značný vplyv na celkový odhad. Jeho hodnota je extrémna jak v súradnici  $x$ , tak v súradnici  $y$ . Bod má značný vplyv na výsledok výpočtu lineárnej regresie.

#### 4.6.1 Identifikácia vplyvových bodov

V tejto kapitole je popísaná identifikácia vplyvu odľahlého merania na odhad neznámych parametrov. Výpočet spočíva v identifikácii extrémnych hodnôt matice  $\mathbf{A} \in R^{m \times n}$ , kde  $m$  je počet observácií a  $n$  je počet neznámych parametrov, ktoré



Obrázok 4.2: Znázornenie odľahlého bodu v ose X [Outliers-53]

môžu mať potenciálne značný vplyv na výsledný vektor neznámych parametrov. Definujme teda regresný model ako

$$\ell = \mathbf{A}\mathbf{x} + \epsilon \quad (4.30)$$

Keď nahradíme porušené neznáme parametre ich odhadom  $\hat{\mathbf{x}}$  dostaneme

$$\hat{\ell} = \mathbf{A}\hat{\mathbf{x}} \quad (4.31)$$

Vypočítame odhad neznámych parametrov

$$\hat{\mathbf{x}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\ell \quad (4.32)$$

a dosadíme do rovnice 4.31

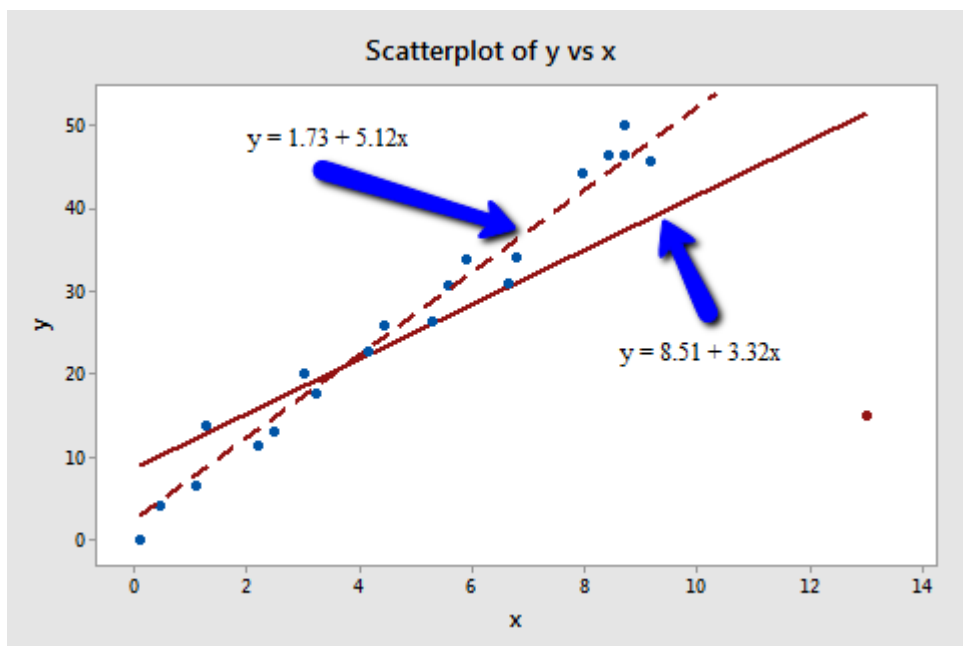
$$\hat{\ell} = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\ell \quad (4.33)$$

Maticu vplyvu merania (*the hat matrix*)  $\mathbf{H} \in R^{m \times m}$  na odhad neznámych parametrov určíme ako

$$\mathbf{H} = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T \quad (4.34)$$

Matica obsahuje informácie o vplyve jednotlivých observácií na výsledný výpočet. V geodézii sa táto matica nazýva kovariančná matica meraní.

$$\hat{\ell} = \mathbf{H}\ell \quad (4.35)$$



Obrázok 4.3: Bod ovplyvňujúci určený odhad [Outliers-53]

diagonála matice vplyvu merania na odhad neznámych parametrov obsahuje hodnoty vplyvu jednotlivých meraní. Hodnota  $h_{ii}$  sa nachádza v intervale  $< 0, 1 >$  a súčet všetkých hodnôt diagonály je rovný počtu neznámych parametrov. Čím je hodnota  $h_{ii}$  bližšia nule, tým je vplyv merania na odhad neznámych parametrov menší. Z geometrického hľadiska je hodnota  $h_{ii}$  vzdialenosť medzi hodnotou  $x$  i-tého bodu a priemeru hodnôt  $x$  všetkých bodov. Za extrémnu hodnotu merania, ktoré môže mať potenciálne vplyv na odhad neznámych parametrov je všeobecne považovaná hodnota presahujúca trojnásobok priemeru všetkých hodnôt diagonály matice  $\mathbf{H}$

$$\bar{h} = \frac{\sum_{i=1}^m h_{ii}}{m} = \frac{n}{m} \quad (4.36)$$

teda

$$h_{ii} > 3 \left( \frac{n}{m} \right) \quad (4.37)$$

## 5 Určenie spoľahlivosti vyrovnania geodetických meraní metódou LMS

Jedným z výsledkov metódy LMS je množina meraní podozrivých z odľahlosti. Aby bolo možné potenciálne odľahlé observácie zo súboru meraní vylúčiť, je nutné s určitou pravdepodobnosťou určiť, či merania skutočne obsahujú hrubé chyby. Určenie meraní, ktoré budú napokon vylúčené zo súboru meraní, sa vykoná pomocou testu štatistických hypotéz. V tejto kapitole budú popísané niektoré postupy analýzy výsledku vyrovnania.

### 5.1 Vzťah medzi reziduami a hrubými chybami

Vzťah medzi reziduami a hrubými chybami je možné vyjadriť vzťahom

$$\mathbf{v} = \Sigma_{vv} \mathbf{P} \epsilon = \mathbf{M} \epsilon \quad (5.1)$$

Predpokladajme, že každá chyba observácie  $\epsilon_i$  sa skladá z náhodnej časti  $\epsilon_{ri}$  a hrubej chyby  $\nabla \ell_i$ . Vektorovo

$$\epsilon = \epsilon_r + \nabla \ell \quad (5.2)$$

takže

$$\mathbf{v} = \Sigma_{vv} \mathbf{P} \epsilon_r + \Sigma_{vv} \mathbf{P} \nabla \ell = \nabla \mathbf{v}_r + \nabla \mathbf{v} \quad (5.3)$$

kde  $\nabla \mathbf{v}_r$  sú reziduá ovplyvnené náhodnými chybami a  $\nabla \mathbf{v}$  sú reziduá ovplyvnené hrubými chybami. Reziduá sú teda ovplyvnené oboma chybami. Po vyrovnaní je pomerne náročne oddeliť náhodné chyby od chýb hrubých iba analýzou vyrovnaných rezidiuí. Z vlastností idempotentnej matice  $\mathbf{M} = \Sigma_{vv} \mathbf{P}$

$$\text{trace}(\mathbf{M}) = \text{rank}(\mathbf{M}) = n - u = r \quad (5.4)$$

kde  $r$  je celková redundancia,  $n$  je počet meraní a  $u$  je počet neznámych parametrov. Celková redundancia je teda rovna súčtu jednotlivých elementov na diagonále.

$$\sum_{i=1}^r = r, \quad 0 \leq r_i \leq 1 \quad (5.5)$$

Keď vezmeme do úvahy rovnicu  $\nabla \mathbf{v} = \Sigma_{vv} \mathbf{P} \nabla \ell$ , tak pre jednotlivé  $v_i$  platí

$$v_i = r_i \nabla \ell_i \quad (5.6)$$

kde  $r_i$  sa nazývajú redundantné čísla a vyjadrujú príspevok jednotilvých meraní  $\ell_i$  do celkovej redundancie. Relatívna redundancia môže byť vyjadrená ako priemer diagonálnych hodnôt matice  $\mathbf{M}$  teda  $\text{trace}(\mathbf{M})/n$ . Z definície

$$\mathbf{M} = \Sigma_{vv}\mathbf{P} = \mathbf{I} - \Sigma_{\hat{\ell}\hat{\ell}}\mathbf{P} = \mathbf{I} - \mathbf{U} \quad (5.7)$$

a

$$u_i = 1 - r_i \quad , 0 \leq r_i \leq 1 \quad (5.8)$$

kde  $u_i$  sú tzv. absorbné čísla

$$\sum u_i = n - \sum r_i \quad , u = n - r \quad (5.9)$$

Keď observácia  $\ell_i$  obsahuje hrubé chyby  $\nabla\ell_i$ , tak

$$u_i\nabla\ell_i = \nabla\ell_i - r_i\nabla\ell_i \quad (5.10)$$

a

$$\nabla\ell_i = u_i\nabla\ell_i + \nabla v_i \quad (5.11)$$

Posledná rovnica ukazuje, že pokiaľ sa hrubá chyba nachádza iba v jednej observácii odrazí sa to práve v konkrétnom reziduu v rovnakej miere ako  $\nabla v_i = r_i\nabla\ell_i$  a zbytok bude rozdelený do odhadu neznámych parametrov. Z toho je zrejmé, že hrubá chyba  $\nabla\ell_i$  ma vysoké redundančné číslo  $r_i$ , ktoré bude ovplyvňovať  $v_i$  a je možné ju odhaliť analýzou reziduí. Čím je väčšia hodnota  $r_i$ , tým je lepšia kontrola nad observáciou. Teda pokiaľ  $r_i = 1$ , to znamená, že 100% hrubej chyby bude obsiahnuté v konkrétnom reziduu a nebude mať vplyv na odhad neznámych parametrov. Na druhú stranu, pokiaľ  $r_i = 0$ , tak to znamená, že nieje žiadna kontrola nad i-tou observáciou a hrubá chyba sa neodrazí v žiadnom reziduu a plne ovplyvňuje odhad neznámych parametrov. Podľa [Kavouras–52] môže byť v rôznych častiach geodetickej siete prakticky rôzna kontrola. Redundantné čísla závisia na robustnosti geometrického návrhu siete a nezávisia na meraných hodnotách.

## 5.2 Globálny test smerodajnej odchýlky

Prvým z testov overenia vyrovnania meraných veličín je globálny test smerodajnej aposteriornej odchýlky  $\sigma_0^2$ . Aplikovanie tohto testu je možné v prípade, že je známa taktiež apriorná presnosť meraní. To znamená, že poznáme apriornu smerodajnú odchýlku  $\sigma_0^2$ . Pre nulovú hypotézu  $H_0$  platí, že  $\hat{\sigma}_0^2/\sigma_0^2$  patrí do Fisherovho rozdelenia  $F_{r,\infty}$ , kde  $r$  je počet nadbytočných meraní. A taktiež predpokladajme, že platí  $F_{r,\infty} =$

$\chi_r^2/r$ . Globálny test sa používa pre detekciu odľahlých meraní, preto sa očakáva, že  $\hat{\sigma}_0^2 > \sigma_0^2$ . Pre nulovú a alternatívnu hypotézu platí

$$H_0 : \sigma_0^2 = \sigma^2 \quad H_a : \sigma_0^2 < \sigma^2 \quad (5.12)$$

Pre test platí

$$\frac{\sigma_0^2}{\sigma_0^2} < F_{r,\infty;1-\alpha} \quad (5.13)$$

a teda

$$\frac{\sigma_0^2}{\sigma_0^2} = \frac{\mathbf{v}^T \mathbf{P} \mathbf{v}}{r \sigma_0^2} \quad (5.14)$$

$$\mathbf{v}^T \Sigma_{\ell\ell}^{-1} \mathbf{v} < r F_{r,\infty;1-\alpha} = \chi_r^2/r \quad (5.15)$$

Pokiaľ sa v observáciach nachádzajú merania s hrubými chybami, potom test vyššie uvedenej kvadratickej formy nemusí platiť. Zaleží na veľkosti chyby odľahlých meraní a akým spôsobom sú odľahlé merania obsiahnuté v reziduách. Pokiaľ test zlyhá, tak je možné zamietnuť nulovú hypotézu  $H_0$ . Zamietnutie nulovej hypotézy môže nastať aj v prípade nesprávneho odhadu váhových koeficientov alebo nekorrektného matematického modelu. Po vylúčení týchto dvoch možnosti je potrebné analyzovať alternatívnu hypotézu  $H_a$ , ktorá zohľadňuje možný posun  $\nabla \varepsilon$  rozdelenia pravdepodobnosti observácii  $\ell$ .

$$E[\ell|H_a] = E[\ell|H_0] + \nabla \varepsilon, E[\mathbf{v}|H_0] = 0, E[\mathbf{v}|H_a] = \nabla \varepsilon \quad (5.16)$$

Vzťah medzi  $\nabla \varepsilon$  a  $\nabla \mathbf{v}$  je

$$\nabla \mathbf{v} = \Sigma_{vv} \mathbf{P} + \nabla \varepsilon \quad (5.17)$$

$$E\left[\frac{\sigma_0^2}{\sigma_0^2} | H_a\right] = E\left[\frac{\sigma_0^2}{\sigma_0^2} | H_0\right] + \nabla \frac{\sigma_0^2}{\sigma_0^2} \quad (5.18)$$

kde

$$\nabla \frac{\sigma_0^2}{\sigma_0^2} = \frac{\nabla \sigma_0^2}{\sigma_0^2} = \frac{1}{\sigma_0^2} \frac{\nabla \mathbf{v}^T \mathbf{P} \nabla \mathbf{v}}{r} = \frac{\lambda}{r} \quad (5.19)$$

a

$$E\left[\frac{\sigma_0^2}{\sigma_0^2} | H_a\right] = 1 + \frac{\lambda}{r} \quad (5.20)$$

$$\lambda = \frac{1}{\sigma_0^2} \nabla \mathbf{v}^T \mathbf{P} \nabla \mathbf{v} = \frac{1}{\sigma_0^2} \nabla \varepsilon \mathbf{P} \Sigma_{vv} \mathbf{P} \nabla \varepsilon \quad (5.21)$$



Pre alternatívnu hypotézu teda platí, že  $\hat{\sigma}_0^2/\sigma_0^2$  má štatisticky necentrálne rozdelenie  $F_{r,\infty,\lambda}$  s necentrálnym parametrom  $\lambda$ . To znázorňuje posun pravdepodobnosti rozdelenia kvôli hrubým chybám  $\nabla\varepsilon$ . Aj napriek tomu, že sú rezidua známe, tak nepoznáme hrubé chyby  $\nabla\varepsilon$  a  $\lambda$ . Z toho nieje možné vyvodiť, ktoré observácie sú kontaminované hrubými chybami. Je ale možné odhadnúť hranicu  $\lambda_0$ , ktorá ukazuje na odchýlku od nulovej hypotézy na hladine pravdepodobnosti  $\alpha_0$  a  $\beta$   $\lambda_0 = \lambda(\alpha, \beta_0, r, \infty)$ . Táto hodnota nám síce dáva informáciu o jednotlivých hodnotách hrubých chýb vektoru  $\nabla\varepsilon$ , ale alternatívna hypotéza  $H_a$  udáva vzťah medzi reziduami a hrubými chybami, takže môže byť určený odhad hranice na hladine pravdepodobnosti  $\beta_0$ .

### 5.3 Data snooping

Majme ďalšiu definíciu alternatívnej hypotézy  $H_a^c : \nabla\varepsilon = \mathbf{c}\nabla\varepsilon$  kde hodnoty vektoru  $\mathbf{c} = (0 \dots 101 \dots 0)$  indikujú observácie podozrivé z odľahlosti. Pri zohľadnení hranice  $\lambda_0$  platí

$$\lambda_0 = \frac{1}{\sigma_0^2} \nabla\varepsilon_0 \mathbf{P} \Sigma_{vv} \mathbf{P} \nabla\varepsilon_0 \quad (5.22)$$

a po dosadení

$$\lambda_0 = \frac{\nabla\varepsilon_0}{\sigma_0^2} \mathbf{c}^T \mathbf{P} \Sigma_{vv} \mathbf{P} \mathbf{c} \quad (5.23)$$

hraničná hodnota  $\nabla\varepsilon_0$  hrubej chyby v obseráciach špecifikovaných vektorom  $\mathbf{c}$  sa získa

$$|\nabla\varepsilon_0| = \sigma_0 \sqrt{\frac{\lambda_0}{\mathbf{c}^T \mathbf{P} \Sigma_{vv} \mathbf{P} \mathbf{c}}} \quad (5.24)$$

Z vyššie uvedených vzťahov je možné odvodiť novú alternatívnu hypotézu  $H_a^c$ , ktorá predpokladá, že práve jedna observácia je zaťažená hrubou chybou. To znamená, že vektor  $\mathbf{c}$  obsahuje samé nulové hodnoty iba okrem prvku  $c_i$  zodpovedajúceho  $i$ -tému meraniu, ktorý ma hodnotu 1.

$$H_a^c : \nabla\varepsilon_{0i} = \mathbf{c}_i \nabla\varepsilon_{0i} \quad (5.25)$$

a

$$|\nabla\varepsilon_{0i}| = \sigma_0 \sqrt{\frac{\lambda_0}{(\mathbf{P} \Sigma_{vv} \mathbf{P})_{ii}}} \quad (5.26)$$

Z jednorozmerného štatistického testu môže byť odvodený test alternatívnej hypotézy

$$w_i = \frac{\mathbf{c}_i^T \mathbf{P} \mathbf{b}}{\sigma_0 \sqrt{\mathbf{c}_i \mathbf{P} \Sigma_{vv} \mathbf{P} \mathbf{c}_i}} \quad (5.27)$$

a pre diagonálnu váhovou maticu platí

$$w_i = \frac{v_i}{\sigma_{vi}} = \frac{v_i}{\sigma_0 \sqrt{\sigma_{vvi}}} \quad (5.28)$$

Z toho vyplýva že štandardizované reziduum  $w_i$  má normálne rozdelenie  $N(0, 1)$ . Tento test naznačuje, že Data-snooping testuje iba jedno štandardizované reziduum v jednom výpočte.

Takže pokiaľ platí, že

$$|w_i| = \left| \frac{v_i}{\sigma_{vi}} \right| > \sqrt{F_{1-\alpha_0; 1, \infty}} \quad (5.29)$$

tak nulová hypotéza  $H_0$  nieje akceptovaná, pričom nulová hypotéza v tomto prípade hovorí, že  $i$ -té meranie neobsahuje hrubú chybu a je označené ako zamietnuté. Pre často volenú a doporučenú hladinu významnosti  $\alpha_0 = 0.001$  je kritická hodnota  $\sqrt{F_{1-\alpha_0; 1, \infty}} = 3.29$  a nulová hypotéza je zamietnutá pokiaľ

$$|w_i| > 3.29 \sigma_{vi} \quad (5.30)$$

Procedúra sa opakuje v prípade podozrenia, že súbor meraní obsahuje viac než jednu hrubú chybu. Podľa [Kavouras–52] je test štandardizovaných reziduí viac robustný, než test reziduí ako takých. Niekedy môže nastať situácia, že nieje dostupná hodnota  $\sigma_{vi}$  a ako štandardizovaná odchýlka je použitá hodnota  $v_i/\sigma_{li}$ . Táto aproximácia však môže znížiť štatistickú hodnotu a zlyhanie testu nastane v nižšom počte prípadov. Všeobecne sa doporučuje použitie globálneho testu pre detekovanie, či sa v súbore meraní nachádza observácia s hrubou chybou, a použitie data-snooping testu pre odhalenie konkrétnych observácií, ktoré hrubú chybu obsahujú. Obe testy by mali byť konzistentné a hraničné hodnoty by sa pre obe testy mali zhodovať. Pre globálny test je nutné vhodne zvoliť hodnotu  $\alpha$  a pre data-snooping hodnotu  $\alpha_0$  a pokiaľ  $\beta = \beta_0$ , tak sú hladiny významnosti oboch testov prepojené pomocou necentralného parametru  $\lambda_0$ .

Štandardná procedúra je:

- voľba  $\alpha_0 = 0.001$
- voľba  $\beta_0 = 0.80$
- výpočet  $\lambda_0$  z  $\lambda_0 = \lambda(\alpha_0, \beta_0, 1, \infty)$

- výpočet  $\alpha$  ako  $\lambda_0 = \lambda(\alpha, \beta_0, r, \infty)$

Pre  $r = 10$  stupňov voľnosti je

| $\alpha_0$ | $\beta_0$ | $\lambda_0$ | $\alpha$ | $\sqrt{F_{1-\alpha_0;1,\infty}}$ | $F_{1-\alpha_0;r,\infty}$ |
|------------|-----------|-------------|----------|----------------------------------|---------------------------|
| 0.05       | 0.8       | 7.85        | 0.3      | 1.96                             | 1.17                      |
| 0.05       | 0.9       | 10.5        | 0.35     | 1.96                             | 1.12                      |
| 0.001      | 0.8       | 17.0        | 0.04     | 3.29                             | 1.9                       |
| 0.001      | 0.9       | 20.9        | 0.05     | 3.29                             | 1.83                      |

## 5.4 Tau-test

V predošlých testoch sa predpokladá, že je hodnota apriórnej smerodajnej odchýlky  $\sigma_0$  známa. Vždy je ale známa aposteriórna smerodajná odchýlka  $\hat{\sigma}_0$ . V tomto prípade nie je možné vykonať vyššie uvedený globálny test a test data-snooping musí byť modifikovaný. Nový štatistický test je teda daný

$$\tau_i = \frac{v_i}{\hat{\sigma}_{vi}} = \frac{v_i}{\hat{\sigma}_0 \sqrt{\Sigma_{vvi}}} = \frac{w_i}{\hat{\sigma}_0} \quad (5.31)$$

a má  $\tau$ -rozdelenie.

Viac rozmerný test je definovaný ako test na maximálnom  $\tau$  na hladine významnosti  $\alpha$  a je funkciou  $n$  nezávislých jednorozmerných testoch na hladine významnosti  $\alpha_0$ .

$$\alpha = 1 - (1 - \alpha_0)^n \quad \alpha_0 \doteq \alpha/n \quad (5.32)$$

Obdobne pokiaľ

$$|\tau_i| = \left| \frac{v_i}{\hat{\sigma}_{vi}} \right| > c_\tau \quad (5.33)$$

tak je  $i$ -té reziduum označené pre zamietnutie.  $\tau$  rozdelenie sa bežne nepoužíva, pretože je pomerne náročné na výpočet.

## 5.5 Test kvadratickej formy reziduí

Táto metóda sa používa v prípade, že je potrebné testovať skupinu reziduí. Majme jednoduchú funkciu reziduí ako kvadratickú formu

$$q^2 = \mathbf{v}^T \mathbf{P} \mathbf{v} \quad (5.34)$$

Štatistický test bol už použitý pre globálny test variancie. Pre tento prípad rozdeľme vektor observácií na

$$\ell = \begin{pmatrix} \ell_1 \\ \ell_2 \end{pmatrix} \quad (5.35)$$

kde

- $\ell_1$  - je skupina meraní bez hrubých chýb
- $\ell_2$  - je skupina meraní podozrivých z odľahlosti

Pre vektor reziduí obdobne platí

$$\mathbf{v} = \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{pmatrix} \quad (5.36)$$

a kovariančnú maticu reziduí  $\Sigma_{\mathbf{v}\mathbf{v}} = \mathbf{P}^{-1} - \mathbf{A}\Sigma_{xx}\mathbf{A}^T$

$$\Sigma_{\mathbf{v}\mathbf{v}} = \begin{pmatrix} \Sigma_{\mathbf{v}\mathbf{v}1} & \Sigma_{\mathbf{v}\mathbf{v}12} \\ \Sigma_{\mathbf{v}\mathbf{v}21} & \Sigma_{\mathbf{v}\mathbf{v}2} \end{pmatrix} \quad (5.37)$$

Skupina príspevkov testovaných meraní k celkovej kvadratickej forme  $q^2$  je daný parciálnou kvadratickou formou

$$d = \mathbf{v}_2^T \Sigma_{\mathbf{v}\mathbf{v}2}^{-1} \mathbf{v}_2 \quad (5.38)$$

Podmienkou je nesingularita matice  $\Sigma_{\mathbf{v}\mathbf{v}2}$ . Pokiaľ je k vyrovnaniu použitá iba množina meraní  $\ell_1$ , tak kvadratická forma reziduí  $\mathbf{v}_1$  nieje ovplyvnená meraniami podozrivými z odľahlosti  $\ell_2$ .

$$\dot{q}^2 = \dot{\mathbf{v}}_1^T \mathbf{P}_1 \dot{\mathbf{v}}_1 \quad (5.39)$$

Z toho je možno odvodiť, že platí

$$\dot{q}^2 = q^2 - d \quad (5.40)$$

Pre nulovú hypotézu predpokladajme, že skupina observácií neobsahuje meranie s hrubou chybou.

$$E[\dot{q}^2] = E[q^2] - E[d] \quad (5.41)$$

Test celkovej kvadratickej formy

$$q^2 < \sigma_0^2 \cdot \chi_{\alpha;r}^2 \quad (5.42)$$

je identický s globálnym testom. Pokiaľ test zlyhá jedným z možných dôvodov je výskyt odľahlých meraní v súbore observácií. Parciálna kvadratická forma je daná ako

$$d < \sigma_0^2 \cdot \chi_{\alpha;p}^2 \quad (5.43)$$

kde  $p$  je veľkosť skupiny meraní podozrivých z odľahlosti  $\ell_2$ . Test ukáže, či je hypotéza o skupine obsahujúcej väčšinu odľahlých meraní validná alebo nie. Test tretej kvadratickej formy  $q_2$

$$q_2 < \sigma_0^2 \cdot \chi_{\alpha;r-p}^2 \quad (5.44)$$

môže odhaliť odľahlé merania v skupine meraní  $\ell_1$ . Správnu detekciu odľahlých observácií je možné vykonať len v prípade dobrej znalosti hodnoty  $\sigma_0^2$  a správnej voľby hladiny významnosti. Pokiaľ je pravdivá hypotéza, že odľahlé merania sú obsiahnuté iba v skupine meraní  $\ell_2$  tak platí

- test 5.42 zlyhá
- test 5.43 taktiež zlyhá
- test 5.44 by mal byť pravdivý

Neplatí ale, že skupina  $\ell_2$  obsahuje iba observácie s hrubou chybou. Obtiažnou úlohou je voľba správnej alternatívnej hypotézy, pre ktorú vyhovujú vyššie uvedené testy.

## 6 Implementačné aspekty numerických operácií riedkych matíc

Riedke matice sú také matice, ktoré obsahujú značný počet nulových hodnôt. Z definície je matica riedka práve vtedy, keď sa na operácie s ňou samotnou vyplatí použiť algoritmy riedkych matíc. Riedke matice majú teda význam hlavne pri algoritmizácii maticových operácií a to z dôvodu možnej redukcie výpočtového času procesoru a redukcie priestoru v pamäti počítača či na pevnom disku. Všeobecne je možné rozdeliť riedke matice na štrukturované a neštrukturované. V prípade štrukturovaných sú nulové hodnoty rozložené pravidelne najčastejšie okolo diagonály. Maticu s nerovnomerným rozložením nulových prvkov nazývame ako neštrukturovanú. Riedke matice je možné s výhodou reprezentovať ako graf matice. Graf matice sa skladá z vrcholov a hrán. V prípade nesymetrických matíc sú hrany orientované. Grafy riedkych matíc hrajú hlavnú úlohu v reprezentácii matíc pri použití riedkych techník.

### 6.1 Graf riedkych matíc

Graf je teda definovaný množinou vrcholov a hrán. Množina vrcholov je definovaná ako

$$V = v_1, \dots, v_n \quad (6.1)$$

Hranu tvorí vždy všeobecná dvojica vrcholov  $v_i, v_j$ . Výnimku tvoria takzvané slučky, kde hrana začína a končí v jednom vrchole

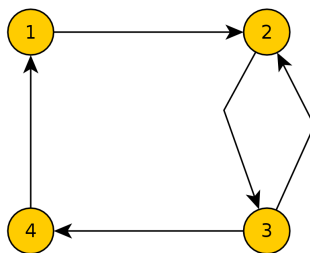
$$E \subseteq V \times V \quad (6.2)$$

Graf je teda možné zapísať ako  $G = (V, E)$  a reprezentuje binárne relácie medzi prvkami matice.

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & 0 & 0 \\ 0 & a_{22} & a_{23} & 0 \\ 0 & a_{32} & a_{33} & a_{34} \\ a_{41} & 0 & 0 & a_{44} \end{pmatrix} \quad (6.3)$$

### 6.2 Štruktúra riedkych matíc

Prvok riedkej matice je jednoznačne určený stĺpcovým indexom a ukazateľom na začiatok riadku, v ktorom sa nachádza. Teda každý uložený prvok riedkej matice zaberá približne dvojnásobok miesta v pamäti oproti matici neriedkej štruktúry. Preto

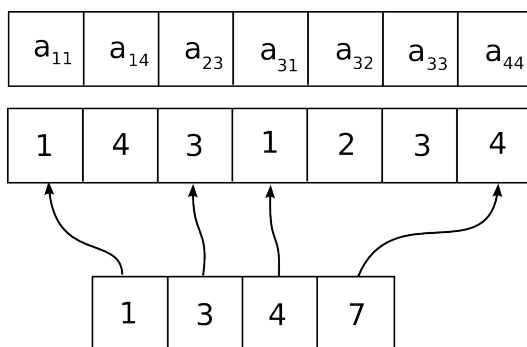


Obrázok 6.1: Graf riedkej matice

je jednoduchou úvahou možné dospieť k tomu, že využitie algoritmov riedkych matíc sa môže vyplatiť vtedy, keď počet nulových prvkov je aspoň trojnásobne vyšší ako počet prvkov nenulových.

Dátová štruktúra riedkej matice je tvorená tromi dynamicky alokovanými poľami. Prvé pole obsahuje nenulové prvky matice. Ku každému nenulovému prvku prináleží jeden stĺpcový index uložený v druhom vektore. Tretie pole obsahuje ukazatele na začiatky  $i$ -tého riadku. Pre takúto dátovú štruktúru je nutné prispôbiť všetky ďalšie operácie a to od prístupu k jednotlivým elementom až po lineárnu kombináciu vektorov či násobenie matíc. Všeobecne je možno s dátovou štruktúrou pracovať ako s dynamickou alebo statickou. V prípade statickej dátovej štruktúry sa rozmer dynamicky alokovaných poľí v priebehu výpočtu nemení, teda polia nieje možné re-alokovať. Z toho vyplýva, že pred započatím výpočtu, v ktorom môžu vzniknúť nové nenulové prvky, je nutné vykonať symbolickú analýzu, ktorej výsledkom je hodnota definujúca rozmer poľí, ktoré budú naplnené hodnotami po vykonaní príslušnej matematickej operácie.

$$\mathbf{A} = \begin{pmatrix} a_{11} & 0 & 0 & a_{14} \\ 0 & 0 & a_{23} & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ 0 & 0 & 0 & a_{44} \end{pmatrix} \tag{6.4}$$



Obrázok 6.2: Štruktúra riedkej matice

### 6.2.1 Dynamicky re-alokovaná štruktúra riedkej matice

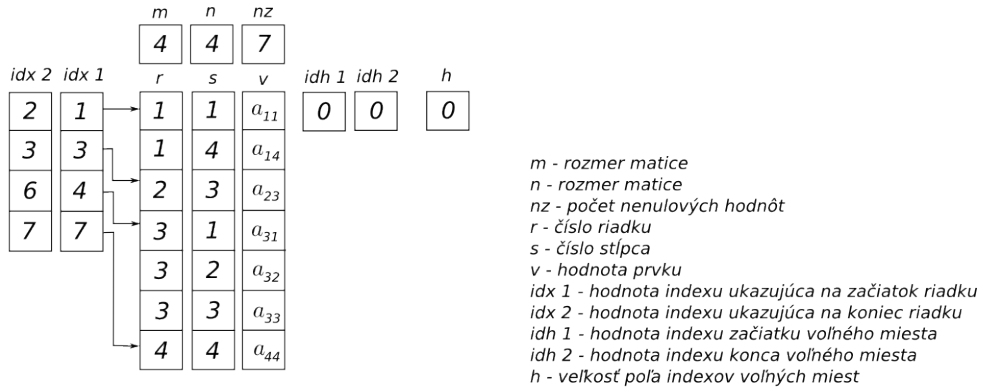
Dynamicky re-alokovanú štruktúru je možné použiť v prípadoch, kde je náročné určiť počet nenulových prvkov riedkej matice, ktoré vzniknú po nejakej matematickej operácii. Túto metódu uvádzam len ako alternatívu k použitej metóde, kde je rozmer výslednej riedkej matice určený výpočtom. Štruktúra riedkej matice je teda v tomto prípade myslená ako agregovaný dátový typ. Pozície prvkov riedkej matice sú uložené v dvojrozmernom poli s dimenziou  $nx2$ , kde  $n$  je počet nenulových prvkov. Hodnoty matice sú uložené v jednorozmernom poli s  $n$  prvkami. Kvôli urýchlení mnohých procesov sa vyplatí uchovávať určité nadbytočné informácie o matici. Prvou informáciou je pole indexov ukazujúcich vždy na začiatok riadku. Pole indexov má rozmer ekvivalentný s počtom riadkov resp. stĺpcov vstupnej matice. Ďalšou súčasťou štruktúry matice sú indexy na konce riadku. Pri presunoch prvkov v matici vznikajú v pamäti voľné nepoužité miesta. Tieto miesta sú následne pri spustení správcu pamäte zaplnené. K lokalizácii týchto voľných miest v matici slúži vektor obsahujúci vždy hornú a dolnú hranicu pozície voľného priestoru. K tomuto vektoru je potrebné si pamätať premennú označujúcu zaplnenie vektora. V prípade polí s pozíciou prvku, jeho hodnotou a riadkovými indexmi sa pamäť alokuje dynamicky. Vektor indexov voľných miest môže byť inicializovaný s pevnou hodnotou. Veľkosť tohto vektora závisí na počte vkladanych prvkov či vektorov do matice a na veľkosti pridaného miesta v pamäti pri realokácii. Preto sa hodnota veľkosti poľa dá predbežne odhadnúť a nehrozí vysoké riziko pretečenia. Štruktúru matice tvoria dodatočné podstatné informácie o dimenzii pôvodnej matice a počte nenulových prvkov.

$$\mathbf{A} = \begin{pmatrix} a_{11} & 0 & 0 & a_{14} \\ 0 & 0 & a_{23} & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ 0 & 0 & 0 & a_{41} \end{pmatrix} \quad (6.5)$$

#### Realokácia pamäti

Realokácia polí vyhradí v pamäti počítača dostatočne voľné miesto pre zápis ďalších prvkov matice. Otázkou je, aký rozsah voľného miesta by sa mal vyhradit'. Veľkosť voľného miesta závisí na veľkosti riedkej matice a približnom odhade počtu operácií, ktoré môžu zmeniť nulové prvky na prvky nenulové. Pre vyhradenie miesta existujú dve základné stratégie. Prvou je, že sa pomocou nejakých výpočtov zistí, koľko nenulových prvkov bude zmenených a podľa toho sa vyhradí potrebné miesto. V tom prípade sú režijné náklady v podobe výpočtového času, na druhu stranu sa vyhradí

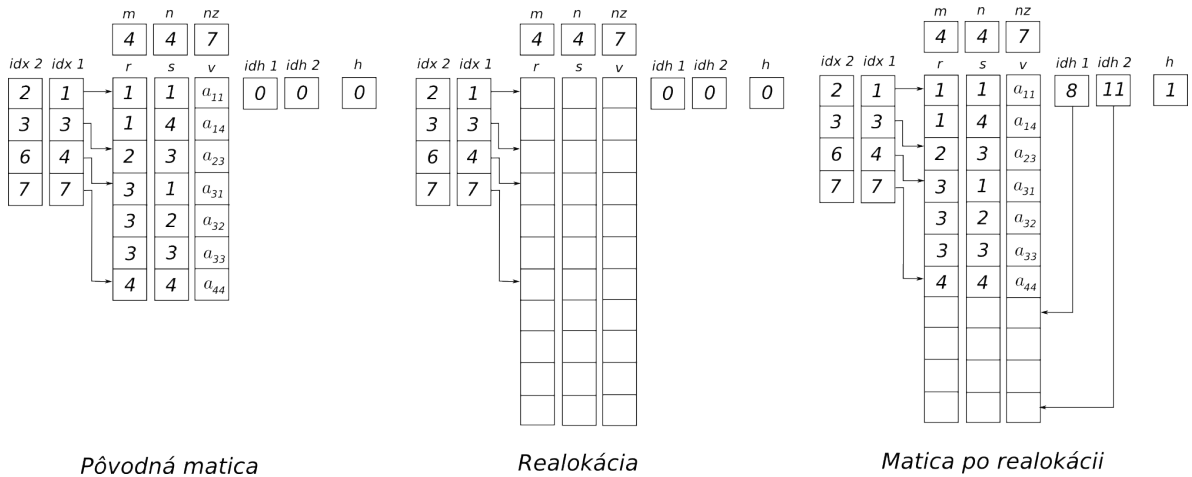




Štruktúra riedkej matice

Obrázok 6.3: Štruktúra dynamicky re-alokovanej riedkej matice A

takmer presný počet miest v pamäti a tým pádom sa môže ušetriť značný počet nadbytočných miest. V prípade, že počet zmenených nenulových prvkov nieje známy, teda nieje predom určený pomocou nejakého výpočtu, veľkosť voľného miesta sa odhadne. To môže mať za následok dve rôzne situácie. V jednej sa vyhradí dostatočný počet miest, avšak časť miesta v pamäti počítača ostane nevyužitá. V druhej situácii voľné miesto v pamäti dôjde. To má za následok spustenie garbage collectoru a ďalšiu realokáciu v pamäti. Realokácia pamäti je výpočtovo náročná operácia, preto by sa mala spúšťať len v niekoľko málo potrebných prípadoch.

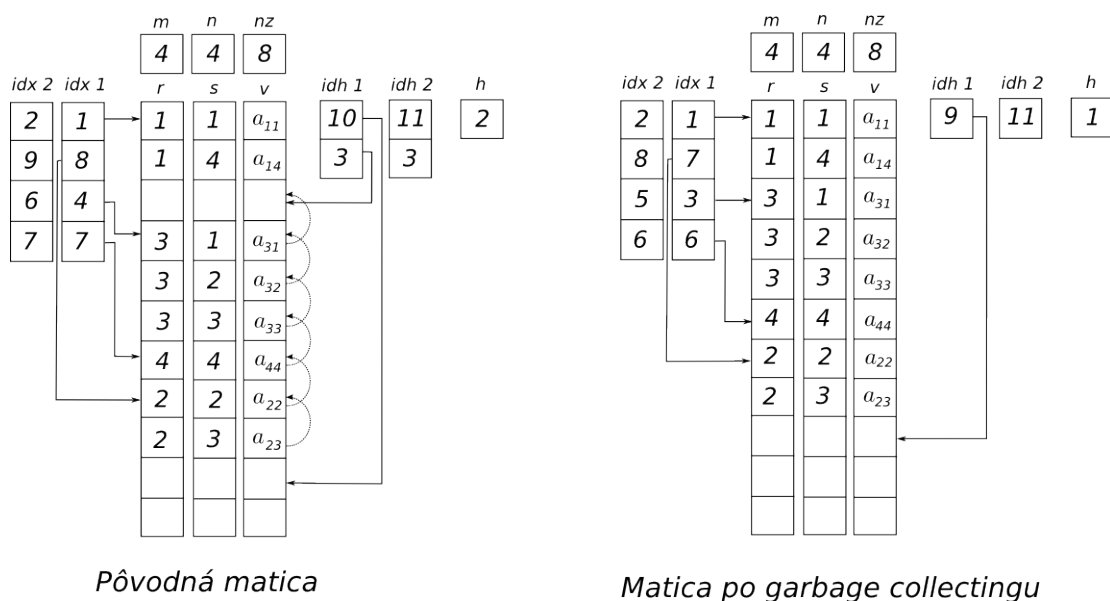


Obrázok 6.4: Realokácia pamäti

### Garbage collector

Garbage collector postupne presúva nepravidelne rozmiestnené prvky matice po pamäti na jedno súvislé miesto spravidla na začiatok alokovaného priestoru a voľné

miesto zostáva na konci priestoru. Garbage collector sa spúšťa pred začiatkom realokácie a v prípade, že pri vložení nového vektora je dostatok voľného miesta v pamäti, avšak miesto nieje súvislé. Garbage collector sa môže spustiť kedykoľvek pri výpočte Choleského rozkladu ortogonalizácie a pod. Takýto spôsob usporiadania pamäti nieje vhodný pre procesy bežiacie v reálnom čase. Pre takéto procesy nieje prípustné, aby sa výpočet v nejakej časti zastavil a spustil sa garbage collector.



Obrázok 6.5: Funkcia garbage collector

### Výber prvku matice

Výber prvku riedkej matice je relatívne nenáročná akcia. Pre výber prvku matice slúži funkcia, ktorá vracia reálne číslo. Vstupnými parametrami je pozícia prvku matice a objekt matice, z ktorej sa prvok vyberá. Pretože sú v objekte matice definované polia s indexmi ukazujúcimi na začiatky riadkov matice, tak algoritmus vyhľadáva prvok iba v požadovanom riadku matice v jednom cykle `for`. Pokiaľ narazí na požadovanú pozíciu stĺpca, cyklus sa zastaví. Pri vyhľadávaní prvku môžu nastať tieto situácie. Zadaná pozícia presahuje rozmer matice, to znamená, že prvok v matici neexistuje, alebo sa prvok v riedkej štruktúre nenachádza, ale hodnota pozície patrí do intervalu dimenzie a to znamená, že hodnota vybraného prvku je nulová.

### Vloženie prvku do matice

Vkladanie prvku do riedkej matice je oproti predošlému výberu prvku o niečo náročnejší proces. Vstupnými parametrami subrutiny je objekt riedkej matice, do ktorej

sa prvok zapíše. Do tohto objektu sa tiež zapíše pozícia prvku, kam sa ma nová hodnota zapísať. Pokiaľ sa prvok v riedkej štruktúre nachádza, vyhledá sa potrebná pozícia a prvok sa jednoducho prepíše. Pokiaľ bol prvok, ktorý sa do vektoru vloží, pôvodne nulový, bude sa musieť štruktúra matice rozšíriť o jeden prvok. Na začiatku sa spočíta počet voľných miest, kde by sa prvok mohol zapísať. Výpočet voľného miesta spočíva v tom, že sa od seba odčíta veľkosť alokovaného vektora riedkej matice a počet nenulových prvkov. Hneď potom nasleduje operácia výpočtu rozmeru vektoru, do ktorého sa bude prvok vkladajúť. Rozmer vektoru rozšírený o vkladajúci prvok musí byť menší ako voľné miesto. Je preto nutné, aby sa pri prvom spustení vkladania prvku, ktorý bol pôvodne nulový spustí realokácia matice. Po realokácii je na konci poľa časť voľného miesta. Do tohto voľného miesta sa postupne zapíšu prvky vektoru, do ktorého sa vkladá požadovaný prvok. Je vhodné popísať zmenu vektorov, do ktorých sa uchováajú rozmery voľných miest. Pred vkladáním prvku je tento vektor prázdny. Po realokácii má vektor na prvom mieste hodnotu pozície posledného prvku matice o jedno väčšiu ako je veľkosť pôvodnej matice a druhý vektor na prvom mieste veľkosť realokovaného vektora. Takto je reprezentovaný voľný priestor pre zápis vektora. Po zápise prvku sa hodnota prvého čísla vo vektore voľného miesta zväčší o hodnotu vkladajúceho vektora a na druhé miesto vektoru voľných miest sa zapíše pozícia, kde sa pred vkladáním nachádzal prvý prvok vektoru, s ktorým sa manipuluje. Vkladajúci vektor sa zapíše vždy na prvú dostatočne veľkú, voľnú pozíciu. Pokiaľ nastane situácia, že vkladajúci vektor je väčší ako akýkoľvek voľný priestor v riedkej matici a zároveň súčet veľkostí voľných priestorov je väčší ako vkladajúci vektor, spustí sa garbage collector. Garbage collector, ako už bolo popísané v predošlej kapitole, z nesúvislých voľných miest vytvorí postupným presunom vektorov jedno súvislé miesto. Na toto miesto je možné zapísať požadovaný vektor. Na záver sa prepíšu indexy ukazujúce na začiatky a konce riadkov.

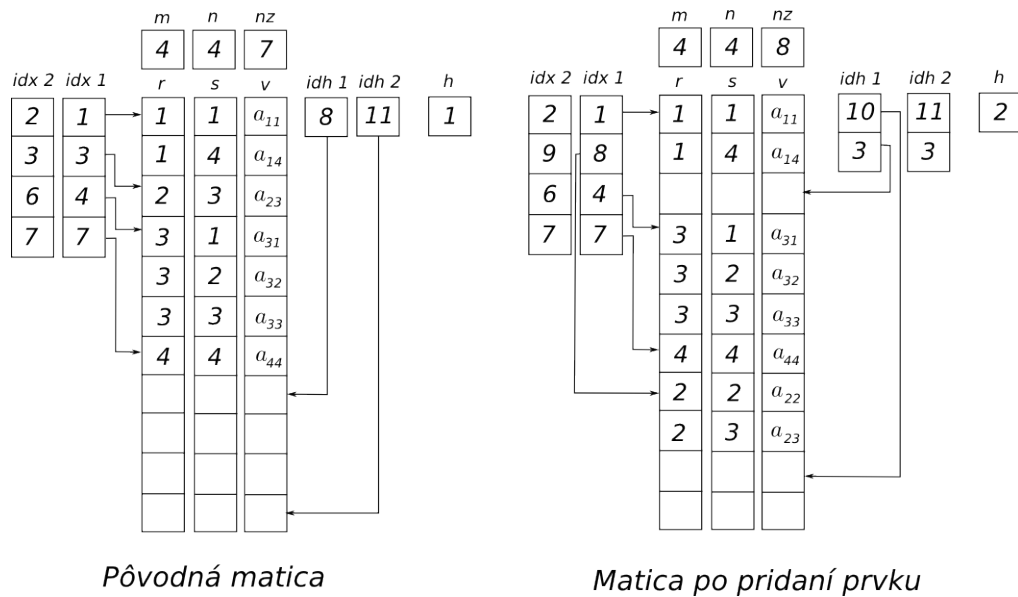
### 6.3 Násobenie riedkej matice a vektora

Násobenie matice  $\mathbf{A} \in \mathbb{R}^{m \times n}$  a vektora  $\mathbf{x} \in \mathbb{R}^n$  pre nejaké  $\mathbf{y} \in \mathbb{R}^m$  platí

$$\mathbf{y} = \mathbf{Ax} + \mathbf{y} \quad (6.6)$$

teda

$$y_i = \sum_{j=1}^n a_{i,j} x_j + y_i, \quad i = 1 \dots n \quad (6.7)$$



Obrázok 6.6: Vloženie prvku do štruktúry riedkej matice

Z toho vyplýva, že pokiaľ je v matici  $\mathbf{A}$  nulový prvok tak po vynásobení príslušným prvkom vektoru vznikne znova nulová hodnota a nakoniec súčet nulových hodnôt. To znamená, že nieje nutné operáciu násobenia nulového prvku vykonávať.

## 6.4 Implementácia násobenia riedkej matice a vektora

Predpokladáme, že matica je reprezentovaná riedkou dátovou štruktúrou. Metóda `begin()` vráti ukazateľ na nenulový prvok riedkej matice, v tomto prípade matice  $\mathbf{A}$ . Metóda `ibegin()` vráti pointer na stĺpcový index nenulového prvku matice. Príslušný prvok vektoru sa vyberie pomocou derferencovanej hodnoty indexu. Vektor je narozdiel od matice indexovaný od nuly matica je indexovaná od 1, preto sa od indexu musí odčítať hodnota rovná jednej.

```

1 Float* y = new Float[size];
2
3 for (Index i=1; i<=size; i++, y++)
4 {
5     Float* n = A->begin(i);
6     Float* e = A->end (i);
7
8     // inicializacie prvku vysledneho vektoru
9     *y = Float();
10
11     for(Index* j=A->ibegin(i) ; n!=e; n++,j++)
12     {

```

```

13         *y += *n * x[*j - 1];
14     }
15 }

```

Kód 6.1: Algoritmus násobenia riedkej matice a vektoru

Pri riešení združených gradientov bez explicitného zostavovania normálnych rovníc je nutné vykonať operáciu  $\mathbf{y} = \mathbf{A}^T \mathbf{x}$ . V prípade riedkych matíc nie je transpozícia matice triviálna a priamočiara úloha v porovnaní s plnými maticami. Transpozícia by znamenala pre-usporiadanie prvkov matice indexov a pointrov na začiatky riadkov matice. V tomto prípade však nieje potrebné explicitne transpozíciu matice zostavovať. Z výhodou teda použijeme algoritmus uvedený nižšie

```

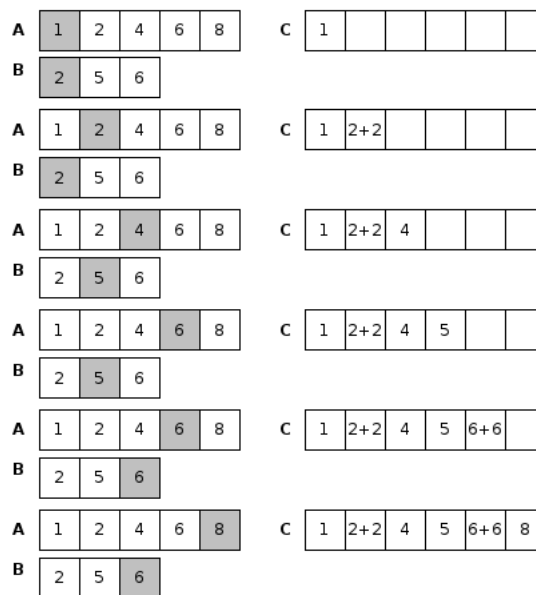
1  Float* y = new Float[cols];
2
3  //inicializacia vektora y
4  for(Index i=Index(); i<cols; i++) y[i]=0.0;
5
6  for(Index i=1; i<=rows; i++ )
7  {
8      Float* n = A->begin(i);
9      Float* e = A->end (i);
10
11     for(Index* j = A->ibegin(i) ; n!=e; n++, j++)
12     {
13         y[*j-1] += *n * x[i-1];
14     }
15 }

```

Kód 6.2: Algoritmus násobenia riedkej transponovanej matice a vektoru

## 6.5 Implementácia sčítania matíc

Implementáciu sčítania riedkych matíc popisuje obrázok 6.7, na ktorom sú znázornené dve polia A a B reprezentujúce indexy na prvky matíc jedného riadku. Pri sčítaní riedkych matíc sa striedavo posúvajú pointre ukazujúce na indexy na prvky matice. V počiatočnom stave ukazujú obe pointre na začiatok polí. V prvom kroku sa porovnávajú hodnoty odkazov na prvky matice. V prípade, že sa hodnoty odkazov zhodujú, sčítajú sa obe prvky matice, pretože to znamená, že sa nachádzajú v jednom stĺpci a obe pointre sa posunú o jeden prvok poľa. Pokiaľ sa hodnoty indexov nezhodujú, posunie sa o jeden prvok práve ten pointer, ktorého hodnota indexu je nižšia. Výpočet sa skončí, až obe pointre dosiahnu posledný prvok poľa. Postup sa opakuje pre všetky riadky riedkej matice.



Obrázok 6.7: Sčítanie riedkych matíc

```

1 for(int i=1; i<=rows; i++)
2 {
3   Float* n1 = A-> begin(i);
4   Float* n2 = B-> begin(i);
5   Float* e1 = A-> end(i);
6   Float* e2 = B-> end(i);
7   Index* i1 = A-> ibegin(i);
8   Index* i2 = B-> ibegin(i);
9
10  C->new_row();
11
12  while(true)
13  {
14    if(*i1<*i2) { C->add_element(*n1,*i1); i1++; n1++; }
15    else if(*i1>*i2) { C->add_element(*n2,*i2); i2++; n2++; }
16    else { C->add_element( *n1 + *n2, *i1); n1++; n2++; i1++; i2++; }
17
18    if(n1==e1 && n2==e2){ break; }
19
20    if(n1==e1) { for(;n2!=e2;n2++,i2++){ C->add_element(*n2,*i2); } break; }
21    if(n2==e2) { for(;n1!=e1;n1++,i1++){ C->add_element(*n1,*i1); } break; }
22
23  }
24 }

```

Kód 6.3: Algoritmus násobenia riedkej matice a vektoru

## 6.6 Násobenie riedkych matíc

Násobenie dvoch matíc je možné vykonať niekoľkými spôsobmi a to ako lineárnu kombináciu stĺpcov matice  $\mathbf{A}$ , skalárny súčin zodpovedajúceho riadku a stĺpca matice pre každý prvok výslednej matice alebo ako súčet vonkajších súčinov matíc. V tejto práci bola zvolená posledná z metód z dôvodu efektivity výpočtu a jednoduchosti implementácie. Predpokladajme, že výsledkom násobenia dvoch matíc  $\mathbf{A} \in \mathbb{R}^{m \times p}$  a  $\mathbf{B} \in \mathbb{R}^{p \times n}$  s riedkou štruktúrou  $\mathbf{C} \in \mathbb{R}^{m \times n}$  je matica s opäť riedkou štruktúrou. Počet prvkov výslednej matice sa môže zmeniť. Podrobný popis zmeny počtu nenulových prvkov je založený na analýze grafu riedkej matice.

## 6.7 Implementácia vonkajších súčinov matíc

Predošlá sekcia je zameraná na súčet dvoch riedkych matíc, preto je postačujúce zamerať sa iba na vonkajší súčin matíc. Ďalej sa obmedzíme na konkrétny prípad maticového súčinu  $\mathbf{A}^T \mathbf{A}$ , ktorý je použitý pre vyhotovenie normálnych rovníc. Uvedený algoritmus rieši vonkajší súčin vektoru  $\mathbf{v}^T \mathbf{v}$ . Implementácia je navrhnutá tak, že nie je nutné explicitne vyhotovovať transpozíciu vstupnej matice. Výsledkom vonkajšieho súčinu vektorov riedkej matice je riedka matica. Výsledná riedka matica obsahuje neznámy počet nenulových prvkov. Znalosť počtu nenulových prvkov výslednej matice je nutná kvôli alokácii riedkej matice v pamäti počítača, ktorá musí byť uskutočnená pred započatím výpočtu. Riedku maticu nie je možné v priebehu výpočtu re-alokovať, pretože operácia relokácie je pomerne časovo náročná, čo by malo za následok zníženie efektivity výpočtu. Jednou z možností ako riedku štruktúru pripraviť je vyhradiť v pamäti toľko miesta, koľko by zaberala plná matica. Toto je jednoduchá cesta, avšak v mnohých prípadoch je dynamicky alokované pole predimenzované. V optimálnom prípade je určený počet nenulových prvkov presne analýzou grafu matice. V prípade násobenia  $\mathbf{A}^T \mathbf{A}$  je počet nenulových prvkov výslednej matice rovný počtu hrán grafu matice  $\mathbf{A}$ . Určiť počet hrán grafu matice je bez použitia štandardnej knižnice jazyka C++ pomerne náročný na implementáciu, pretože pri analýze je potrebné sledovať, či pri vložení hrany grafu do štruktúry už hrana neexistuje. Tento problém je možné vyriešiť použitím štandardných kontajnerov `map<>` resp. `set<>`. Z toho vyplýva, že veľkosť štandardného kontajnera je rovná počtu prvku výslednej riedkej matice.

```

1 Float* p = A->begin(j);
2 Index* ii = A->ibegin(j);
3 Float m = 0;
4
5 for(Index k=1;k<=cols;k++)

```

```
6 {
7     Float* n = A->begin(j);
8     Float* e = A->end(j);
9
10    N->new_row();
11
12    if(*ii==k)
13    {
14        for(Index* i=A->ibegin(k) ; n!=e; n++, i++)
15        {
16            m = *p * *n;
17            N->add_element(m,*i);
18        }
19        p++;ii++;
20    }
21 }
```

Kód 6.4: Algoritmus vonkajšieho súčinu dvoch vektorov



## 7 Numerické metódy riešenia sústavy lineárnych rovníc

V procese výpočtu metódy LMS sa opakovane rieši sústava lineárnych algebraických rovníc. Počet opakovaní riešenia je závislé na veľkosti sústavy. So stúpajúcim počtom rovníc nelineárne rastie aj počet opakovaní výpočtu. Existuje mnoho metód určenia neznámych parametrov sústavy lineárnych rovníc. Určenie neznámych parametrov je možné pomocou jak iteračných, tak finitných metód. Kritériami výberu metódy bola ich presnosť, efektívnosť a možnosť implementácie s riedkymi maticami. Prvou popísanou metódou je Gramm-Schmidtova ortogonalizácia, ktorú pre vyrovnanie sprostredkujúcich meraní zovšeobecnil F. Charamza v [Charamza–35]. Ďalšou finitnou metódou je metóda singulárneho rozkladu. Poslednou popísanou metódou je iteračná metóda združených gradientov, ktorá je z hľadiska efektivity výpočtu optimálna pre určenie neznámych parametrov sústavy lineárnych algebraických rovníc. V priebehu výpočtu združených gradientov sa nevyhotovujú normálne rovnice. Priebeh konvergenzie je možné ovplyvniť prepodmiením.

### 7.1 Gramm-Schmidtova ortogonalizácia

Prvou z možností riešenia sústavy  $\mathbf{Ax} = \mathbf{b}$  je ortogonálnym rozkladom na matice  $\mathbf{QR}$ .

*Veta:* Nech  $\mathbf{A}$  je matica typu  $m \times n$ ,  $m \geq n$ , s lineárne nezávislými stĺpcami, potom existuje jediná dvojica matíc  $\mathbf{Q}$   $m \times n$  a  $\mathbf{R}$   $n \times n$  taká, že  $\mathbf{Q}$  je diagonálna matica s kladnými diagonálnymi prvkami  $\mathbf{D}$  a  $\mathbf{R}$  je horná trojuholníková matica

$$\mathbf{A} = \mathbf{QR} \quad (7.1)$$

Takýto rozklad je možné získať pomocou Gramm-Schmidtovho ortogonalizačného procesu. Majme bázu vektorov  $\mathbf{x}_1 \dots \mathbf{x}_n$ , z ktorej vytvoríme ortogonálnu bázu  $\mathbf{v}_1 \dots \mathbf{v}_n$ . Algoritmus procesu môžeme zapísať

$$\begin{aligned} \mathbf{v}_1 &= \mathbf{x}_1 \\ \mathbf{v}_2 &= \mathbf{x}_2 - \frac{\mathbf{x}_2 \mathbf{v}_1}{\mathbf{v}_1 \mathbf{v}_1} \mathbf{v}_1 \\ \mathbf{v}_3 &= \mathbf{x}_3 - \frac{\mathbf{x}_3 \mathbf{v}_1}{\mathbf{v}_1 \mathbf{v}_1} \mathbf{v}_1 - \frac{\mathbf{x}_3 \mathbf{v}_2}{\mathbf{v}_2 \mathbf{v}_2} \mathbf{v}_2 \\ &\vdots \\ \mathbf{v}_n &= \mathbf{x}_n - \frac{\mathbf{x}_n \mathbf{v}_1}{\mathbf{v}_1 \mathbf{v}_1} \mathbf{v}_1 - \frac{\mathbf{x}_n \mathbf{v}_2}{\mathbf{v}_2 \mathbf{v}_2} \mathbf{v}_2 - \dots - \frac{\mathbf{x}_n \mathbf{v}_{n-1}}{\mathbf{v}_{n-1} \mathbf{v}_{n-1}} \mathbf{v}_{n-1} \end{aligned} \quad (7.2)$$

Ortonormálnu bázu  $\mathbf{v}_1 \dots \mathbf{v}_n$  získame vynásobením ortogonálneho vektoru reciprokovou hodnotou normy daného vektoru

$$\begin{aligned} \mathbf{u}_1 &= \frac{1}{\|\mathbf{v}_1\|} \mathbf{v}_1 \\ &\vdots \\ \mathbf{u}_n &= \frac{1}{\|\mathbf{v}_n\|} \mathbf{v}_n \end{aligned} \quad (7.3)$$

Rozklad na matice  $\mathbf{QR}$  s lineárne nezávislými stĺpcami následne získame tak, že z pôvodnej matice  $A = [\mathbf{x}_1 \dots \mathbf{x}_n]$ , kde  $\mathbf{x}_1 \dots \mathbf{x}_n$  sú lineárne nezávislé stĺpce matice  $\mathbf{A}$ , získame ortogonalizačným procesom maticu  $\mathbf{Q} = [\mathbf{u}_1 \dots \mathbf{u}_n]$ , kde stĺpce matice tvoria ortonormálnu bázu vektorov. Z toho vyplýva, že  $\mathbf{Q}^T \mathbf{Q} = I$  teda pokiaľ je matica  $\mathbf{A} = \mathbf{QR}$  platí

$$\mathbf{R} = \mathbf{Q}^T \mathbf{A} \quad (7.4)$$

### 7.1.1 Zovšeobecnená ortogonalizácia

[Charamza–35] navrhol zovšeobecnenú ortogonalizáciu, ktorá sa s výhodou používa pre vyrovnanie sprostredkujúcich veličín pomocou GSO.

Uvažujme blokovú maticu

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{A}_3 & \mathbf{A}_4 \end{pmatrix} \quad (7.5)$$

Kde

- $\mathbf{A}_1$  ma rozmer  $m \times n$
- $\mathbf{A}_2$  ma rozmer  $m \times q$
- $\mathbf{A}_3$  ma rozmer  $q \times n$
- $\mathbf{A}_4$  ma rozmer  $p \times q$

pričom  $p \geq 0$ ,  $q \geq 0$

Zovšeobecnenie spočíva v pretvorení matice  $\mathbf{A}$  Gramm-Schmidtovou ortogonalizáciou na maticu  $\mathbf{Q}$  s rovnakou štruktúrou

$$\mathbf{Q} = \begin{pmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \\ \mathbf{Q}_3 & \mathbf{Q}_4 \end{pmatrix} \quad (7.6)$$

Výpočet je vykonaný podľa pravidla, že skalárny súčin je počítaný iba z prvkov submatic  $\mathbf{A}_1\mathbf{A}_2$  a ortogonalizačný proces je použitý na maticu  $\mathbf{A}_1$ . Použitie ortogonalizácie na maticu  $\mathbf{A}_1$  sa nelíši od popísaného postupu ortogonalizácie

$$\mathbf{A}_1 = \mathbf{Q}_1\mathbf{R} \quad (7.7)$$

Z predchádzajúceho odstavca uvažujeme, že matica  $\mathbf{R}$  je horná trojuholníková a v prípade, že matica  $\mathbf{A}_1$  má lineárne nezávislé stĺpce, tak aj matica  $\mathbf{R}$  je regulárna a je možné vypočítať jej inverziu  $\mathbf{R}^{-1}$ . Transformácia  $\mathbf{A}_1 \rightarrow \mathbf{Q}_1$  spočíva v prenásobení submatice  $\mathbf{A}_1$  maticou  $\mathbf{R}^{-1}$  z prava. Z toho platí

$$\begin{aligned} \mathbf{Q}_1 &= \mathbf{A}_1\mathbf{R}^{-1} \\ \mathbf{Q}_2 &= \mathbf{A}_2 - \mathbf{Q}_1\mathbf{Q}_1^T\mathbf{A}_2 \\ \mathbf{Q}_3 &= \mathbf{A}_3\mathbf{R}^{-1} \\ \mathbf{Q}_4 &= \mathbf{A}_4 - \mathbf{Q}_3\mathbf{Q}_1^T\mathbf{A}_2 \end{aligned} \quad (7.8)$$

## 7.2 Singulárny rozklad SVD

Algoritmus singularného rozkladu sa považuje za veľmi stabilný a rýchly. Existujú však prípady, v ktorých môže výpočet trvať až dvakrát dlhšie oproti QR rozkladu, avšak v konečnom dôsledku by to nemalo mať vplyv na stabilitu výpočtu. Pre výpočet singularných čísel a vektorov existuje okrem iných Jacobiho metóda pre SVD. Metóda opakovane násobí maticu  $\mathbf{A}$  elementárnou ortogonálnou maticou (tzv. Jacobiho rotácie), pokiaľ matica  $\mathbf{A}$  nekonverguje k  $\mathbf{U}\Sigma$ . Výsledkom je matica  $\mathbf{V}$ . Metóda je síce pomalšia ako ostatné algoritmy SVD (QR algoritmus, DQDS algoritmus, metóda bisekcie a inverznej iterácie a pod.), avšak pre malé hodnoty singularných čísel a vektorov dáva omnoho presnejšie výsledky ako spomínané algoritmy [Demel-46].

*Veta:* Každú maticu  $\mathbf{A}^{m \times n}$  je možné rozložiť na súčin

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T \quad (7.9)$$

kde  $\mathbf{U} \in R^{m \times m}$ ,  $\mathbf{V} \in R^{n \times n}$  sú ortonormálne matice a matica  $\Sigma \in R^{m \times n}$  je diagonálna

$$\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_k), k = \min(m, n) \quad (7.10)$$

a diagonálne prvky splňujú

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \geq \sigma_{r+1} = \dots = \sigma_k \quad (7.11)$$

kde  $r = h(\mathbf{A})$

Diagonálne prvky matice  $\sigma_1, \sigma_2, \dots, \sigma_k$  matice  $\Sigma$  sa nazývajú singulárne čísla matice  $\mathbf{A}$ ,  $i$ -ty stĺpec matice  $\mathbf{U}$  sa nazýva  $i$ -tý ľavý singulárny vektor a  $i$ -tý stĺpec matice  $\mathbf{V}$  sa nazýva  $i$ -tý pravý singulárny vektor. V tom prípade platí

$$\mathbf{u}_i^T \mathbf{A} = \sigma_i \mathbf{v}_i^T, \quad \mathbf{A} \mathbf{v}_i = \sigma_i \mathbf{u}_i, \quad i = 1, 2, \dots, k \quad (7.12)$$

Pre ortonormálne matice  $\mathbf{U}$  a  $\mathbf{V}$  platí

$$\mathbf{U}^T \mathbf{U} = \mathbf{I}, \quad \mathbf{V}^T \mathbf{V} = \mathbf{I} \quad (7.13)$$

Inverziu matice  $\mathbf{A}$  teda dostaneme ako

$$\mathbf{A} = \mathbf{V} \mathit{diag} \left( \frac{1}{\sigma_i} \right) \mathbf{U} \quad (7.14)$$

teda

$$\mathbf{x} = \mathbf{A}^{-1} \mathbf{b} = \mathbf{V} \mathit{diag} \left( \frac{1}{\sigma_i} \right) \mathbf{U} \mathbf{b} \quad (7.15)$$

### 7.3 Metóda združených gradientov

Riešenie sústavy lineárnych rovníc  $\mathbf{A} \mathbf{x} = \mathbf{b}$  je ekvivalentné s hľadaním minima funkcionálu

$$F(x) = \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} - \mathbf{x}^T \mathbf{b} \quad (7.16)$$

Extrém funkcionálu je v smere gradientu teda vektoru, ktorého súradnice sú rovné parciálnej derivácii funkcie  $F$  v bode  $\mathbf{x}_k$

$$-\nabla F(\mathbf{x}_k) = \mathbf{b} - \mathbf{A} \mathbf{x}_k \quad (7.17)$$

Hodnotu  $-\nabla F(\mathbf{x}_k)$  označme ako  $\mathbf{r}_k$ , čo je hodnota rezídua. Majme ďalej aproxiáciu  $\mathbf{x}_k$  a postupujme po smere danom smerovým vektorom  $\mathbf{p}_k$  s vhodne zvolenou konštantou  $\alpha_k$ . Matematicky je túto situáciu možné zapísať

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k \quad (7.18)$$

Koeficient  $\alpha_k$  určíme tak, že vykonáme minimalizáciu  $F(\mathbf{x}_k + \alpha_k \mathbf{p}_k)$

$$F(\mathbf{x}_k + \alpha_k \mathbf{p}_k) = \frac{1}{2} (\mathbf{x}_k + \alpha_k \mathbf{p}_k)^T \mathbf{A} (\mathbf{x}_k + \alpha_k \mathbf{p}_k) - (\mathbf{x}_k + \alpha_k \mathbf{p}_k)^T \mathbf{b} \quad (7.19)$$

$$F'(\mathbf{x}_k + \alpha_k \mathbf{p}_k) = \mathbf{p}_k^T \mathbf{A} (\mathbf{x}_k + \alpha_k \mathbf{p}_k) - \mathbf{p}_k^T \mathbf{b} \quad (7.20)$$

Derivácia bude nulová práve vtedy, keď

$$\alpha_k = \frac{\mathbf{p}_k^T (\mathbf{b} - \mathbf{A}\mathbf{x}_k)}{\mathbf{p}_k^T \mathbf{A}\mathbf{p}_k} \quad (7.21)$$

a po dosadení rezidua

$$\alpha_k = \frac{\mathbf{p}_k^T \mathbf{r}_k}{\mathbf{p}_k^T \mathbf{A}\mathbf{p}_k} \quad (7.22)$$

Hodnota rezidua v ďalšom kroku je teda

$$\mathbf{r}_{k+1} = \mathbf{b} - \mathbf{A}\mathbf{x}_{k+1} \quad (7.23)$$

a taktiež

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{A}\mathbf{p}_k \quad (7.24)$$

V poslednom kroku je nutné nejakým spôsobom definovať smerový vektor  $\mathbf{p}_{k+1}$ . Momentálne sme schopní zostrojiť aproximáciu  $\mathbf{x}_{k+1}$  a rezíduum  $\mathbf{r}_{k+1}$ . Smerové vektory sú určené ortogonalizáciou rezíduí vzhľadom k skalárnemu súčinu, ktorý je daný pozitívne definitnou maticou  $\mathbf{A}$  ako  $\mathbf{p}_i^T \mathbf{A}\mathbf{p}_j = 0$  pre  $i \neq j$ . Ďalší smerový vektor  $\mathbf{p}_{k+1}$  ortogonálny k vektoru  $\mathbf{p}_k$  je potom daný ako lineárna kombinácia rezidua  $\mathbf{r}_{k+1}$  a vektora  $\mathbf{p}_k$

$$\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta_k \mathbf{p}_k \quad (7.25)$$

kde koeficient  $\beta_k$  je volený tak, aby bol nový smerový vektor kolmý k ostatným smerovým vektorom. Z podmienky ortogonalítity teda odvodíme, že

$$\beta_k = -\frac{\mathbf{p}_k^T \mathbf{A}\mathbf{r}_{k+1}}{\mathbf{p}_k^T \mathbf{A}\mathbf{p}_k} \quad (7.26)$$

Ostáva do-odvodiť klasický tvar koeficientov  $\alpha$  a  $\beta$ . Z rovnice 7.25 potom vyplýva  $\mathbf{p}_k = \mathbf{r}_k + \beta_{k-1} \mathbf{p}_{k-1}$ , teda  $\mathbf{p}_k^T \mathbf{A}\mathbf{p}_k = (\mathbf{r}_k + \beta_{k-1} \mathbf{p}_{k-1})^T \mathbf{A}\mathbf{p}_k = \mathbf{r}_k^T \mathbf{A}\mathbf{p}_k$ . Následne pre  $\alpha$  a  $\beta$  platí

$$\alpha_k = \frac{\mathbf{r}_k^T \mathbf{r}_k}{\mathbf{p}_k^T \mathbf{A}\mathbf{p}_k} \quad (7.27)$$

a z rovnice 7.24

$$\mathbf{A}\mathbf{p}_k = -\frac{1}{\alpha_k} (\mathbf{r}_{k+1} - \mathbf{r}_k) \quad (7.28)$$

a potom

$$\beta_k = \frac{1}{\alpha_k} \frac{(\mathbf{r}_{k+1} - \mathbf{r}_k)^T \mathbf{r}_{k+1}}{\mathbf{p}_k^T \mathbf{A} \mathbf{p}_k} = \frac{\mathbf{r}_{k+1}^T \mathbf{r}_{k+1}}{\mathbf{r}_k^T \mathbf{r}_k} \quad (7.29)$$

### 7.3.1 Zastavenie výpočtu

Pretože metóda združených gradientov, je iteračná metóda je nutné určiť hranicu, kedy je výsledok výpočtu dostatočne presný pre ukončenie výpočtu. Prvým kritériom je, že počet iterácií v metóde združených gradientov by nemal presiahnuť veľkosť matice. Druhým kritériom je, že zmena rezidui medzi dvoma krokmi je ovplyvnená nanaajvýš zaokrúhľovaciami chybami.

$$\boldsymbol{\varepsilon} = \mathbf{x} - \mathbf{x}_0 \quad (7.30)$$

$$\mathbf{r} = \mathbf{A} - (\mathbf{x} - \mathbf{x}_0) = \mathbf{A}\boldsymbol{\varepsilon} \quad (7.31)$$

$$\|\mathbf{r}\| = \|\mathbf{A}\boldsymbol{\varepsilon}\| \leq \|\mathbf{A}\| \|\boldsymbol{\varepsilon}\| \quad (7.32)$$

$$\frac{\|r\|}{\|r_0\|} \approx \frac{1}{\kappa} \frac{\|\boldsymbol{\varepsilon}\|}{\|\boldsymbol{\varepsilon}_0\|} < \tau \quad (7.33)$$

$$\frac{\|r\|}{\|r_0\|} < \frac{1}{\kappa} \tau \quad (7.34)$$

$$\|r\| \leq \frac{1}{\kappa} \tau \|r_0\| \quad (7.35)$$

$$\kappa \|r\| \leq \tau \|r_0\|, \quad \kappa = \frac{\lambda_{max}}{\lambda_{min}} \quad (7.36)$$

Najväčšie a najmenšie vlastné číslo je možné odhadnúť pomocou Lanczosovej tri-diagonálnej matice, ktorú je možné určovať v každom kroku metódy združených gradientov [Saad–17].

$$\mathbf{A} = \begin{pmatrix} \frac{1}{\alpha_0} & \frac{\sqrt{\beta_0}}{\alpha_0} & & & & & \\ \frac{\sqrt{\beta_0}}{\alpha_0} & \frac{1}{\alpha_1} + \frac{\beta_0}{\alpha_0} & \frac{\sqrt{\beta_1}}{\alpha_1} & & & & \\ & & \cdot & \cdot & & & \\ & & & \cdot & & & \\ & & & & \cdot & & \\ & & & & & \frac{\sqrt{\beta_{m-2}}}{\alpha_{m-2}} & \\ & & & & & \frac{\sqrt{\beta_{m-2}}}{\alpha_{m-2}} & \frac{1}{\alpha_{m-1}} + \frac{\beta_{m-2}}{\alpha_{m-2}} \end{pmatrix} \quad (7.37)$$

Označme diagonálne prvky tri-diagonálnej matice  $b_1 \dots b_n$  a subdiagonálne prvky  $c_1 \dots c_{n-1}$ . Matica má tvar

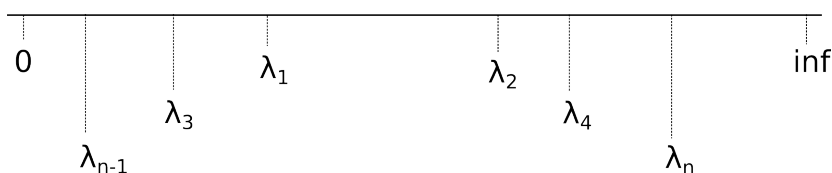
$$\mathbf{A} = \begin{pmatrix} \lambda - b_1 & -c_1 & & & \\ -c_1 & \lambda - b_2 & -c_2 & & \\ & & \cdot & \cdot & \cdot \\ & & & c_{n-1} & \lambda - b_n & c_{n-1} \end{pmatrix} \quad (7.38)$$

Pokiaľ označíme znakom  $f_{n-i}(\lambda)$  determinant, ktorý vznikne z tejto matice vyškrtnutím posledných  $n - i$  riadkov a stĺpcov platí

$$f_{n-(i+1)}(\lambda) = (\lambda - b_{i+1}) f_{n-i}(\lambda) - c_i^2 f_{n-(i-1)}(\lambda), \quad i = 1 \dots n - 1 \quad (7.39)$$

Hodnota funkcie v bode je vypočítaná pomocou niekoľkých krokov metódy sečen.

$$\lambda_n = \lambda_{n-1} - f(\lambda_{n-1}) \frac{\lambda_{n-1} - \lambda_{n-2}}{f(\lambda_{n-1}) - f(\lambda_{n-2})} \quad (7.40)$$



Obrázok 7.1: Poloha vlastných čísel

Hodnota  $inf$  na obrázku 7.1 je najmenšia časť komplexnej roviny, v ktorej sa nachádzajú vlastné čísla matice.

### 7.3.2 Metóda predpodmienovaných združených gradientov

Výberom vhodného prepodmieňovača sa môže znížiť počet iterácií pri výpočte nepriaznivo podmienených sústav. Majme symetrickú pozitívne definitnú maticu  $\mathbf{A}$  a prepodmieňovač  $\mathbf{M}$ . Požadujeme, aby matica  $\mathbf{M}$  bola taktiež symetrická, pozitívne definitná. Požiadavok na maticu  $\mathbf{M}$  je taktiež, aby jej výpočet  $\mathbf{M}\mathbf{x} = \mathbf{b}$  nebol časovo náročný, pretože algoritmus v každom kroku rieši túto rovnicu. Majme teda náhradnú rovnicu

$$\tilde{\mathbf{A}}\mathbf{x} = \tilde{\mathbf{b}} \quad (7.41)$$

teda

$$\mathbf{M}^{-1}\mathbf{A}\mathbf{x} = \mathbf{M}^{-1}\mathbf{b} \quad (7.42)$$

$$\mathbf{A}\mathbf{M}^{-1}\mathbf{u} = \mathbf{b}, \quad \mathbf{x} = \mathbf{M}^{-1}\mathbf{u} \quad (7.43)$$

Matica  $\mathbf{M}$  je teda matica vypočítaná z neúplného Choleského rozkladu

$$\mathbf{M} = \mathbf{L}\mathbf{L}^T \quad (7.44)$$

Zachovanie symetrie docielime rozdelením prepodmieňovača na pravú a ľavú stranu

$$\mathbf{L}^{-1}\mathbf{A}\mathbf{L}^{-T}\mathbf{u} = \mathbf{L}^{-1}\mathbf{b}, \quad \mathbf{x} = \mathbf{L}^{-T}\mathbf{u} \quad (7.45)$$

### 7.3.3 Neúplný Choleského rozklad s riedkou maticou

Jedným z prepodmieňovačov vstupnej matice je neúplný Choleského rozklad (ILU). Všeobecne existujú dve možnosti pre zhotovenie ILU. Prvou je, že sa do rozkládanej matice vkladajú len prvky spadajúce do určitého predom definovaného intervalu. V prípade tejto situácie nieje možné bez analýzy matice určiť počet nenulových prvkov, ktoré vznikli rozkladom. Analýza matice by predstavovala ďalšie režijné náklady, čo by malo za následok zvýšenie času výpočtu prepodmieňovača. Druhou možnosťou je stanoviť podmienku, že do výsledného rozkladu nebudú vkladané nenulové prvky, ktoré vznikli rozkladom. Tento prípad je nazývaný ILU0. Choleského rozklad je preto možné vykonávať na vstupnej veľkosti matice bez dynamickej alokácie pamäti. Je dôležité si uvedomiť že zvyšovaním intervalu hodnôt, ktoré môžu v rozklade vzniknúť sa zvyšuje kvalita prepodmieňovača, to znamená že výpočet gradientov môže konvergovať k výsledku rýchlejšie. Na druhú stranu však výpočet prepodmieňovača je časovo náročnejší. Hranica, kedy je veľkosť intervalu vyvážená pre čo najvyššiu rýchlosť konvergenzie gradientov a čo najvyššiu rýchlosť výpočtu ILU, je odlišná pre každú maticu. Hranicu je možné stanoviť pre určitú skupinu dát či úloh s podobným príznakom.

Pre úlohy popisované v tejto práci bol zvolený ILU0 pre jeho jednoduchšiu implementáciu a vysokú rýchlosť vlastného výpočtu bez potreby ďalších režijných nákladov na výpočet veľkosti pamäti k alokácii. Testy na experimentálnych dátach ukázali, že vo väčšine prípadoch použitie ILU0 zrýchli celkový výpočet gradientov a rapidne zvýši presnosť výpočtu.

Všeobecný výpočet Choleského rozkladu je daný

$$\mathbf{M} = \mathbf{L}\mathbf{L}^T \quad (7.46)$$

Prvky matice  $\mathbf{L}$  môžeme vypočítať pomocou algoritmu



$$l_{jj} = \sqrt{m_{jj} - \sum_{s=1}^{j-1} l_{js}^2} \quad j = 1, \dots, n \quad (7.47)$$

$$l_{ij} = \frac{m_{ij} - \sum_{s=1}^{j-1} l_{js} l_{is}}{l_{jj}} \quad i = j + 1, \dots, n \quad (7.48)$$

### 7.3.4 Riešenie lineárnych rovníc trojuholníkových systémov

V metóde predpodmienенých združených gradientov sa určujú predpodmienенé rezídua ako  $\mathbf{z} = \mathbf{M}^{-1}\mathbf{r}$ , pričom  $\mathbf{M} = \mathbf{L}\mathbf{L}^T$  resp.  $\mathbf{M} = \mathbf{U}^T\mathbf{U}$  je zrejmé, že v každom kroku sa bude riešiť sústava lineárnych rovníc s hornou resp. dolnou trojuholníkovou maticou.

$$\mathbf{z} = (\mathbf{U}^T\mathbf{U})^{-1}\mathbf{r} = \mathbf{U}^{-1}\mathbf{U}^{-T}\mathbf{r}, \quad \mathbf{U}^{-T}\mathbf{r} = \mathbf{q} \quad (7.49)$$

V prvom kroku sa teda bude riešiť priama substitúcia  $\mathbf{U}^{-T}\mathbf{r} = \mathbf{q}$ , pričom  $\mathbf{U}^T = \mathbf{L}$  a v druhom kroku substitúcia spätná  $\mathbf{U}^{-1}\mathbf{q} = \mathbf{z}$

Priama substitúcia je daná vzorcom

$$q_i = \frac{r_i - \sum_{j=1}^{i-1} l_{ij}q_j}{l_{ii}} \quad i = 1, \dots, n \quad (7.50)$$

Spätná substitúcie je daná vzorcom

$$z_i = \frac{r_i - \sum_{j=1+1}^n u_{ij}z_j}{u_{ii}} \quad i = n, \dots, 1 \quad (7.51)$$

Keďže vyhotovenie transpozície je v prípade riedkych matíc časovo náročná úloha je vhodné sa tejto operácii vyhnúť. Výstupom z predpodmieňovača je horná trojuholníková matica. Pre priamu substitúciu je teoreticky potrebné vyhotovenie matice transponovanej z hornej trojuholníkovej matice. Obecne je možné túto úlohu nahraďiť riešením bez explicitného vytvárania transpozície úlohou stĺpcovej verzie spätnej substitúcie.

```

1 for(Index k=1; k<=size; k++)
2 {
3   Float* n=L->begin(k);
4   Float* e=L->end(k);
5   Index* i=L->ibegin(k);
6

```

```
7     for( ; n!=e; n++, i++)
8     {
9         if(*i==k){ break;}
10    }
11    z[k-1] /= *n;
12    n++; i++;
13
14    for( ; n!=e; n++, i++)
15    {
16        z[*i-1] -= z[k-1] * *n;
17    }
18 }
```

Kód 7.1: Algoritmus stĺpcovej verzie priamej substitúcie

```
1 for (Index k=size; k>0; k--)
2 {
3     Float* e=L->end(k)-1;
4
5     Float t = z[k-1];
6
7     for(Index* i=L->iend(k)-1 ; *i!=k ; e--, i--)
8     {
9         t -= *e * z[*i-1];
10    }
11
12    p[k-1] = z[k-1] = t / *e;
13 }
```

Kód 7.2: Algoritmus spätnej substitúcie

## 8 Implementačné detaily LMS a vybraných robustných metód.

V počiatočnej fáze bolo vykonané overenie funkčnosti metódy LMS. Spoľahlivé určenie minima mediánu štvorcov rezidui je možné dosiahnuť vyriešením všetkých možných kombinácií meraní. Merania postupne z výpočtu odstraňujeme. Dôležitou podmienkou je, aby minimálny počet pozorovaní obsahoval aspoň jedno nadbytočné meranie. V opačnom prípade sú reziduá nulové a tým pádom aj ich medián. Overenie metódy LMS spočíva vo generovaní vstupných dát sústavy lineárnych rovníc a ich vyriešení pomocou metódy MNČ, metódy LMS a vybraných robustných metód. Generovanie dát je popísané v nasledujúcej kapitole. Veľkosť sústavy je nutné obmedziť tak, aby bolo možné spočítať všetky kombinácie v reálnom čase. Pravá strana lineárnych rovníc, ktorá reprezentuje merané hodnoty, môže byť určená tak, že:

- neobsahuje žiadne poruchy
- obsahuje malé poruchy z normálnym rozdelením
- neobsahuje žiadne poruchy iba odľahlé merania
- obsahuje malé poruchy z normálnym rozdelením a odľahlé merania

V nasledujúcich kapitolach bude detailne popísaný postup overenia správnosti metódy LMS.

### 8.1 Generovanie vstupných dát

V tejto kapitole sú definované požiadavky na vstupné data, ktoré sú použité pre posúdenie presnosti a porovnanie metódy LMS s vybranými robustnými metódami. Kapitola obsahuje taktiež popis postupu určenia vstupných dát prostriedkami jazyka C++11.

#### 8.1.1 Vygenerovanie vstupných parametrov sústavy lineárnych rovníc

Majme maticu plánu  $\mathbf{A} \in R^{m \times n}$ ,  $m > n$ , ktorá je vygenerovaná funkciou pre rovnomerné rozdelenie reálnych hodnôt v intervale  $\langle a, b \rangle$  podľa funkcie pravdepodobnosti

$P(i|a, b) = 1/(b - a)$ . Matica plánu je všeobecná, obdĺžniková, regulárna, pozitívne definitná. Ďalej majme vektor riešenia  $\mathbf{x} \in R^n$  predom daný ako postupnosť celých čísel  $1 \dots n$ . Exaktne daný vektor riešenia je zvolený z dôvodu kontroly výpočtu vyrovnania. Keďže má matica plánu a vektor riešenia sústavy známe hodnoty, je jednoduché spätne určiť vektor meraných hodnôt  $\mathbf{b} \in R^M$ ,  $\mathbf{b} = \mathbf{A} * \mathbf{x}$ . Rozmery sústavy sú taktiež generované náhodne s podmienkou  $m = rand(i), i \in \langle 10, 20 \rangle$ ,  $n = m/2 - rand(k), k \in \langle 1, 3 \rangle$

### 8.1.2 Vygenerovanie a rozdelenie porúch meraní

Vektor meraných hodnôt  $\mathbf{b}$  môže obsahovať dve druhy porúch. Štandardné poruchy merania, ktoré sú vygenerované podľa funkcie normálneho rozdelenia definovanej ako

$$f(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2} \left(\frac{x - \mu}{\sigma}\right)^2\right) \quad (8.1)$$

Vektor vygenerovaných porúch  $\mathbf{e}$  je pripočítaný k vektoru meraných hodnôt  $\mathbf{b}_e = \mathbf{b} + \mathbf{e}$ . Druhým typom porúch sú také poruchy  $\mathbf{E}$ , ktorých pripočítaním k meraným hodnotám vzniknú odľahlé merania  $\mathbf{b}_E = \mathbf{b}_e + \mathbf{E}$ . Počet odľahlých meraní je náhodne generovaný alebo je ho možné explicitne zvoliť. Bez ujmy na všeobecnosti sú odľahlé merania rozdelené medzi prvých  $N$  meraní.

### 8.1.3 Implementácia generátoru v C++11

Na vygenerovanie vstupných dát sú použité funkcie zo štandardnej knižnice jazyka C++11. Inštancia dátového typu `random_device` reprezentuje generátor rovnomerne rozdelených nedeterministických pseudo-náhodných čísel, ktorá je vstupným parametrom konštruktoru enginu náhodných čísel `mt19937` založeného na *Mersenne Twister* algoritme. Tento algoritmus produkuje vysoko kvalitne náhodné nezáporné 32-bitové celé čísla. Inštancia tejto triedy slúži ako vstupná hodnota pri generovaní konkrétneho rozdelenia. V tejto práci bolo použité normálne (Gaussovo) rozdelenie reprezentované triedou `normal_distribution`, rovnomerné rozdelenie reálnych čísel `uniform_real_distribution` a rovnomerné rozdelenie celých čísel `uniform_int_distribution`.

```

1 #include <random>
2 ...
3 std::random_device rd;

```

```

4 std::mt19937 generator(rd());
5 std::normal_distribution<> norm(0,0.1);
6 std::uniform_real_distribution<> rdist(-500,500);
7 std::uniform_int_distribution<Index> idist(8,20);
8 ...
9 // generate first design matrix
10 for (Index i=1; i<=M; i++)
11     for (Index j=1; j<=N; j++)
12         A(i,j) = rdist(generator);

```

Kód 8.1: Generátor náhodných čísel

## 8.2 Exaktný výpočet metódy LMS

Exaktný výpočet LMS je možné vykonať tak, že vyriešime sústavu lineárnych rovníc pre všetky možné kombinácie meraní. Označme počet meraní ako  $m$  a počet neznámych parametrov ako  $n$ . Kombinácie vytvárame tak, že zo sústavy rovníc postupne vylučujeme  $1 \dots m + 1$  meraní a im odpovedajúci riadok matice plánu. Pre každú kombináciu vyriešime sústavu  $\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{b}$  a určíme medián štvorcov reziduí  $med(\mathbf{A}\mathbf{x} - \mathbf{b})$ . Pokiaľ je medián menší než medián určený v predošlom riešení, nahradíme posledný výsledok riešenia výsledkom novým, zodpovedajúcim minimálnemu mediánu. Postupne tak vyriešime  $\binom{m}{m/2+1}$  sústav. Riešenie všetkých možných kombinácií v reálnom čase obmedzuje veľkosť sústavy. Aby bolo možné medián určiť exaktne v reálnom čase, tak by počet meraných hodnôt nemal prekročiť 25 pri variabilnom počte neznámych parametrov. V prípade určovania rozsiahlejších sústav, je nutné využiť pokročilejšie metódy k určeniu približného riešenia. Exaktné riešenie neznámych parametrov je potom porovnané s riešením určeným metódou MNČ a vybranými robustnými metódami.

## 8.3 Implementácia vybraných M-odhadov

Riešenie určené metódou LMS je vhodné porovnať s nejakou rovnocennou metódou, čo sa týka robustnosti v prípade výskytu odľahlých meraní. Pre porovnanie boli zvolené niektoré M-odhady. Matematický aparát M-odhadov je detailne popísaný v kap. 4.4. M-odhady sú v zásade založené na opakovanom preváhovaní riešenej sústavy, kde váhy  $\mathbf{w}$  sú volené v závislosti na odhadovej funkcii, vplyvovej funkcii a veľkosti opráv  $\mathbf{v}$ . Počet opakovaní preváhovaní  $\mathbf{N} = \mathbf{A}^T \mathbf{w} \mathbf{A}$ ,  $\mathbf{n} = \mathbf{A}^T \mathbf{w} \mathbf{b}$  je rovný počtu meraných hodnôt. V tabuľke sú uvedené vybrané odhady použité pre porovnanie výsledkov a voľba konštánt pre určenie váh.

| Metóda         | a     | b    | c    |
|----------------|-------|------|------|
| Huber          | •     | •    | 1.5  |
| Modified Huber | •     | 30.0 | 20.0 |
| Hampel         | 2.0   | 4.0  | 8.0  |
| Talwar         | 2.795 | •    | •    |
| Tukey          | 4.685 | •    | •    |
| Geman Mcclure  | •     | •    | •    |
| Fair           | 2.985 | •    | •    |
| L1             | •     | •    | •    |
| L1L2           | •     | •    | •    |

Tabuľka 8.1: Voľba konštánt vybraných robustných odhadov

Implementácia:

Generická funkcia `solve` triedy `MEstimators` očakáva ako vstupný parameter ukazateľ na funkciu definujúcu konkrétny odhad. Funkcia vráti adresu na vektor triedy `GNU_gama::Vec` obsahujúci určené váhy, ktoré sú znovu použité vo funkcii `solve`.

```

1 int solve( int (*estimator)(Mat& w, const Vec& v, MEstimators* pme ) )
2 {
3   Mat w(m_rows, m_rows);
4   Vec x(m_cols),v(m_rows);
5
6   Mat N;
7   Vec n;
8
9   set_eye(w);
10
11  for( Index i = 1; i <= m_rows; i++ )
12  {
13    N = trans(*m_A) * w * *m_A;
14    n = trans(*m_A) * w * *m_b;
15
16    x = inv(N) * n;
17    v = *m_A * x - *m_b;
18
19    if ( (*estimator)(w,v,this) ) return -1;
20  }
21
22  m_x = x;
23  m_v = v;
24
25  return 0;
26 }
27
28 static int huber_estimator(Mat& w, const Vec& v, MEstimators* pme)

```

```

29 {
30     double c = 1.5;
31
32     pme->set_eye(w);
33
34     for ( Index i = 1; i <= pme->m_rows; i++ )
35     {
36         if( pme->abs(v(i)) > c )
37         {
38             w(i,i) = c/pme->abs(v(i));
39         }
40     }
41
42     return 0;
43 }
44
45 ...

```

Kód 8.2: Definícia riešiča robustných odhadov

Volanie vybraných robustných funkcií v súbore *main.cpp* je potom možné vykonať tak, že naplníme vektor pointrov na funkcie. Iteráciou cez vektor sa potom vypočítajú všetky robustné odhady.

```

1
2 MEstimators mestimator(A, b);
3 vector< int (*)(Mat&, const Vec&, MEstimators*) > func_vec;
4
5 func_vec.push_back(MEstimators::huber_estimator);
6 func_vec.push_back(MEstimators::modified_huber_estimator);
7
8 for( unsigned int i = 0; i < func_vec.size(); i++)
9 {
10     if( mestimator.solve(func_vec[i]) >= 0 )
11     {
12         mestimator.print_solution();
13     }
14 }

```

Kód 8.3: Volanie funkcií robustných odhadov

## 8.4 Implementácia metódy LTS

Algoritmom určeným pre výpočet LTS je získaný odhad neznámych parametrov vyčerpaním všetkých možných kombinácií. Algoritmus je takmer totožný s algoritmom LMS s tým rozdielom, že namiesto minimalizácie mediánu sa hľadá riešenie s minimálnou sumou štvorcov rezidui množiny orezanej o určitý počet meraní. Maximálny počet odstránených meraní je  $n/2$ . Celý výpočet je implementovaný v triede *LtsExact*. Dátové členy triedy sú totožné s dátovými členmi LMS s tým rozdielom, že namiesto minimálneho mediánu je definovaná suma štvorcov reziduí.

```
1 ...
2
3 Mat N = trans(Asample) * Asample;
4 Vec n = trans(Asample) * bsample;
5
6 x = inv(N) * n;
7 v = Asample * x - bsample;
8
9 double sum_residuals = 0.0;
10
11 for ( Index i = 1; i <= m_sample_size; i++ )
12     sum_residuals += v(i)*v(i);
13
14
15 if( sum_residuals < m_sum_residuals )
16 {
17     m_sum_residuals = sum_residuals;
18     m_x = x;
19     m_v = v;
20 }
21
22 ...
```

Kód 8.4: Hlavný výpočet LTS

V prvom kroku sa vyberie vzor meraní a určí sa odhad neznámych parametrov riešením sústavy lineárnych rovníc. Následne sa určí suma štvorcov reziduí a porovná sa s predošlou sumou. Pokiaľ je suma menšia než suma z predošlého kroku, tak sa vzor sústavy uloží ako potenciálne riešenie. Riešením je množina meraní s minimálnou sumou štvorcov reziduí po vyčerpaní všetkých kombinácií.

## 8.5 Implementácia metódy RANSAC

Obdobne ako pre metódu LMS je výpočet RANSAC vykonaný pre všetky kombinácie meraní s tým rozdielom, že pri výpočte je použitý vždy vzor, ktorý obsahuje nutný počet meraní. Množina meraní, ktorá tvorí množinu zhody, nieje obmedzená. Celý výpočet je implementovaný v triede `Ransac`. Trieda má niekoľko privátnych členov a to ukazatele na maticu plánu a vektor meraní. Ďalšími členmi je maximálna vzdialenosť meraní od určeného odhadu a počet observácií tvoriaci množinu zhody. Výsledkom výpočtu je teda odhad neznámych parametrov, vyrovnané opravy neznámych parametrov a indexy meraní tvoriaci množinu zhody.

```
1 ...
2
3 Mat N = trans(Asample) * Asample;
4 Vec n = trans(Asample) * bsample;
5
6 x = inv(N) * n;
```



```
7
8 v_est = *m_A * x - *m_b;
9
10 consensus_set.clear();
11
12 for( Index i = 1; i<= m_rows; i++)
13 {
14     if( abs(v_est(i)) < m_dist )
15     {
16         consensus_set.push_back(i);
17     }
18 }
19
20 if( consensus_set.size() > m_consensus_set.size() )
21 {
22     m_consensus_set.clear();
23     m_consensus_set = consensus_set;
24
25     m_x = x;
26     m_v = v_est;
27 }
28
29 ...
```

Kód 8.5: Hlavný výpočet RANSAC

V prvom kroku sa vyberie minimálny vzor meraní a určí sa odhad neznámych parametrov riešením sústavy lineárnych rovníc. Odhad neznámych parametrov sa dosadí do pôvodnej sústavy, pričom sa určia hodnoty vyrovnaných reziduí. Následne sa absolútne hodnoty vyrovnaných reziduí porovnajú s maximálnou vzdialenosťou. Merania náležiacie reziduám vyhovujúcim podmienke, že ich absolútna veľkosť je menšia než maximálna prípustná vzdialenosť tvoria  $n$ -tú množinu zhody. Pokiaľ je počet meraní tvoriaci množinu zhody väčší než počet meraní v predchádzajúcom kroku, tak sa riešenie uloží do pripravených dátových premenných. Množina zhody, ktorý obsahuje maximálny počet meraní, je riešením metódy RANSAC. V algoritme nie je implementovaná podmienka ukončujúca výpočet predčasne a to z dôvodu, že dáta sú generované s obmedzenou veľkosťou, tak aby bolo možné vykonať všetky kombinácie s minimálnymi vzormi. V opačnom prípade je nutné implementovať podmienku ukončenia výpočtu. Výpočet je možné zastaviť v momente, keď množina zhody dosiahne požadovanú veľkosť alebo v momente splnenia podmienky popísanej v kap. 4.5.

## 8.6 Výpočet LMS využitím viacjadrových procesorov

V dizertačnej práci je kladený dôraz okrem vysokej presnosti výpočtu taktiež na vysokú efektivitu pri určovaní výsledku mnohonásobného riešenia sústav lineárnych rovníc. Vysokú efektivitu je možné dosiahnuť implementáciou vhodných algoritmov na výpočet danej úlohy, či využitím hardvérových zdrojov výpočtových prostriedkov. V dnešnej dobe je integrácia procesorov s viacerými jadrami do výpočtových zariadení prakticky štandardom. Súčasťou práce bola preto taktiež analýza zdrojového kódu metódy združených gradientov, vytypovanie miest pre prípadnú paralelizáciu, zistenie potenciálnych miest, kde by dochádzalo k súpereniu vlákien o dáta. Jedným z miest vhodných pre paralelizáciu je výpočet skalárnych súčinov masívne sa vyskytujúcich pri opakovanom výpočte združených gradientov. Dôraz bol kladený na to, aby implementované vlákna boli súčasťou knižnice použitého jazyka. Všetky metódy sú teda súčasťou štandardnej knižnice C++11.

### 8.6.1 Výpočet skalárnych súčinov s využitím multi-thread technológie

Jednou z hlavných matematických operácií, ktoré metóda združených gradientov využíva je opakovaný výpočet skalárnych súčinov dvoch vektorov. Výsledkom skalárneho súčinu je agregovaný súčet súčinov jednotlivých prvkov vektorov. Výpočet skalárnych súčinov dvoch vektorov je možné vykonať s využitím viacerých vlákien procesorov. Vektory sú rovnomerné rozdelené na zhodne veľké časti a každá časť je spracovávaná jedným konkrétnym vláknom.

```

1 Float dot_product(const Float* a, const Float* b, Index size)
2 {
3     Float dp = Float();
4     for (Index i=Index(); i<size; i++) dp += a[i]*b[i];
5     return dp;
6 }

```

Kód 8.6: Implementácia skalárneho súčinu bez využitia vlákien

Funkcia pre výpočet skalárneho súčinu dvoch vektorov s použitím *multi-thread* technológie musí byť rozšírená o niekoľko ďalších parametrov. Prvým z parametrov je odkaz na výsledok skalárneho súčinu. Táto premenná je zdieľaná všetkými vláknami, ktoré sú zapojené do výpočtu. Prístup k tejto premennej musí byť chránený zámkom `std::mutex`, aby k nej vlákna nepristupovali náhodne a ľubovoľne ju nemenili. Ďalšie dve parametre sú indexy počiatku a konca časti rovnomerne rozdeleného poľa, ktoré je možné nazvať tiež hranicami časti výpočtu. Funkcia

`block_threads_until_finish_this_job(barrier)` je špeciálny druh zámku, ktorý sa chová ako bariéra a určuje, že k premennej výsledku skalárneho súčinu majú data pristupovať synchronne.

```

1  static void dot_product_multithread(const Float *a, const Float *b,
2                                     Float& dp, Index left_bound,
3                                     Index right_bound )
4  {
5      Float partial_dp = Float();
6
7      for( Index i=left_bound; i < right_bound; i++)
8      {
9          partial_dp += a[i]*b[i];
10     }
11
12     std::lock_guard<std::mutex>
13     block_threads_until_finish_this_job(barrier);
14
15     dp += partial_dp;
16 }

```

Kód 8.7: Funkcia pre výpočet skalárneho súčinu dvoch vektorov pomocou viacerých jadier

Určenie hraníc výpočtu je implementované vo funkcii uvedenej nižšie. Jednotlivé hodnoty hraníc sú ukladané do štandardného kontajnera. Výpočet hraníc vektorov je prirodzene závislý na počte vlákien, ktoré sú použité vo výpočte. V prípade neceločíselného rozdelenia vektoru je posledná časť väčšia o celočíselný zvyšok po delení.

```

1
2  std::vector<Index> bound(Index threads, Index size) {
3
4      std::vector<Index> bounds;
5
6      Index delta = size / threads;
7      Index reminder = size % threads;
8      Index n1 = 0, n2 = 0;
9
10     bounds.push_back(n1);
11
12     for (Index i = Index(); i < threads; ++i)
13     {
14         n2 = n1 + delta;
15
16         if (i == threads - 1) { n2 += reminder; }
17
18         bounds.push_back(n2);
19         n1 = n2;
20     }
21
22     return bounds;
23 }

```

Kód 8.8: Funkcia pre určenie hraníc rozdeleného vektoru

Konkrétne volanie funkcie prebieha tak, že naplníme štandardný kontajner vždy funkciou vlákna, ktorá bude vykonávať výpočet skalarného súčinu definovaného intervalu. Spustenie výpočtu potom prebieha pripojením vlákien výpočtu do hlavného vlákna. Až výpočet skončí, je vhodné štandardný kontajner uvoľniť, aby v ďalšom kroku výpočtu bolo počítané iba z novodefinovanými vláknami.

```

1
2 Float dp = 0.0;
3
4 for (Index i = 0; i < threads_count; i++)
5 {
6     threads.push_back(std::thread(dot_product_multithread, s,
7     s, std::ref(dp), bounds_cols[i], bounds_cols[i+1]));
8 }
9
10 for(auto &t : threads) { t.join(); }
11
12 threads.clear();

```

Kód 8.9: Vykonanie úlohy skalarného súčinu viacerými jadrami

## 8.6.2 Výpočet bez použitia exkluzívnych zámkov

Synchronizácia vlákien, teda obsluha zámkov, vyžaduje ďalšie režijné náklady, ktoré sa negatívne prejavujú v dobe výpočtu. V tomto konkrétnom príklade je možné použiť trik, ktorý umožní synchronizáciu bez nutnosti použitia zámkov. Trik spočíva v tom, že má zdieľaná premenná datový typ `atomic`, u ktorej je zaistený bezpečný zápis a čítanie. Atomická premenná je tzv. triviálne kopírovateľná a povoľuje len niektoré atomické operácie, ktoré sú definované v hlavičke `atomic`. Podľa [Cplusplus-36] je možné atomickú hodnotu pre číselné typy definovať iba ako celočíselný typ. Aby bolo možné definovať atomickú hodnotu ako typ `float`, je nutné explicitne definovať operáciu pričítania čísla s pohyblivou desatinnou čiarkou. Funkcia `load()` atomicky načíta hodnotu do pomocnej premennej, do ktorej je možné uložiť výsledok nejakej operácie. Funkcia `compare_exchange_weak()` porovná reprezentácie atomického objektu, nad ktorým je volaná s prvým argumentom funkcie teda objektom, ktorý vznikne kópiou z druhého argumentu funkcie. Druhý argument funkcie je objekt, v ktorom je uložená nová hodnota. V tomto konkrétnom prípade sa teda pričíta medzivýsledok skalarného súčinu do atomickej premennej. Medzivýsledok je potom pomocou operácie `std::memcpy` skopírovaný do prvého argumentu funkcie.

```

1
2 static void dot_product_multithread_lockfree
3 (
4     const Float *a, const Float *b,
5     std::atomic<Float>& dp,
6     Index left_bound, Index right_bound

```

```

7 )
8 {
9     Float partial_dp = Float();
10
11     for( Index i=left_bound; i < right_bound; i++)
12     {
13         partial_dp += a[i]*b[i];
14     }
15
16     auto dot_product = dp.load();
17
18     while ( !dp.compare_exchange_weak(dot_product,
19         dot_product + partial_dp) );
20 }

```

Kód 8.10: Výpočet bez použitia exkluzívnych zámkov

### 8.6.3 Paralelizácia určenia minimálneho mediánu štvorcov reziduí

Výpočet minimálneho mediánu štvorcov reziduí je pomerne časovo náročná úloha. Dobu času výpočtu je možné výrazne zredukovať efektívnym využitím výpočtových zdrojov. Úloha je riešená použitím niekoľkých vlákien procesoru. V prípade, že určíme kombinácie  $n$ -vzorov matice  $\mathbf{A} \in R^{m \times n}$ ,  $m > n$ , tak vlákno  $i$  určuje minimálny medián teda všetky kombinácie submatice  $\mathbf{A} \in R^{m-i \times n}$ ,  $m > n$ , kde  $i \in 1 \dots m/2 + 1$ . V C++ triede `LmsExact` na výpočet minimálneho mediánu štvorcov reziduí určením všetkých kombinácií je použitá metóda `get_all_combinations()`, ktorá pre každú submaticu určuje hodnotu minimálneho mediánu pomocou funkcie `get_combinations()`, kde `m_sample_size` je veľkosť submatice a `m_rows` a `m_cols` je veľkosť pôvodnej matice. Funkciu môžeme zavolať nad inštanciou objektu `get_all_combinations()`, ktorou exaktne určíme riešenie LMS.

```

1 int get_all_combinations()
2 {
3     for( ; m_sample_size > Index(lms_rows/2); --m_sample_size )
4     {
5         get_combinations( m_rows, m_sample_size);
6     }
7     return 0;
8 }

```

Kód 8.11: Výpočet mediánu

V prípade, že chceme výpočet zefektívniť využitím viacerých procesorov môžeme túto funkciu postupne volať v  $n$ -vláknach a to tak, že si pre každú submaticu vytvoríme inštanciu triedy LMS. Na danej inštancii potom voláme funkciu `get_combinations()`.

```

1
2 vector< LmsExact > vec_lms;
3 vector< std::thread > vec_threads;
4
5 Index lms_rows = A->rows();
6 Index lms_cols = A->cols();
7 Index lms_sample_size = lms_rows - 1;
8
9 for( ; lms_sample_size > Index(lms_rows/2); --lms_sample_size )
10 {
11     vec_lms.push_back( LmsExact(A, b) );
12 }
13
14 lms_sample_size = lms_rows - 1;
15
16 for( unsigned int i = 0 ; i < vec_lms.size(); i++ )
17 {
18     vec_lms[i].set_sample_size( lms_sample_size );
19
20     vec_threads.push_back( std::thread( &LmsExact::get_combinations,
21                                       &vec_lms[i], lms_rows,
22                                       lms_sample_size-- )
23                           );
24 }
25
26 for( unsigned int i = 0; i < vec_threads.size(); i++ )
27 {
28     vec_threads[i].join();
29 }
30
31 vec_threads.clear();

```

Kód 8.12: Výpočet LMS n-vláknami

V prípade, že sa vytvorí viac vlákien ako je výpočtových zdrojov, tak vlákna, ktoré nezískali výpočtový zdroj, čakajú, kým nejaké vlákno zdroj neuvoľní a následne začnú vykonávať výpočet. Na záver máme v inšanciách triedy `LmsExact` uložené kombinácie s minimálnym mediánom pre konkrétne submatice.

```

1
2 for( unsigned int i = 0; i < vec_lms.size(); i++ )
3 {
4     vec_lms[i].print_solution();
5     cout << "Median multi thread: " << vec_lms[i].get_median() << endl;
6 }

```

Kód 8.13: Výpis minimálnych mediánov pre konkrétne submatice

Minimálny medián a výsledné riešenie určíme na záver z hodnôt minimálnych mediánov konkrétnych submatíc.

## 9 Porovnanie výsledkov vyrovnania simulovaných dát určených metódou LMS a vybranými robustnými metódami.

Táto kapitola úzko súvisí s predošlou kapitolou kap. 8, v ktorej boli popísané implementačné detaily robustných metód. Tieto algoritmy boli použité na riešenie príkladov uvedených v tejto kapitole. Porovnanie výsledkov vyrovnania medzi metódou najmenších štvorcov, metódou LMS a vybranými robustnými metódami je vykonané na príkladoch zo súboru generovaných sústav popísaných v kap. 8.1. Pre zhodnotenie presnosti a výkonu robustných metód bolo vygenerovaných približne 500 sústav rôznych veľkosti s rôznym počtom odľahlých meraní. Z tohto súboru boli vybrané príklady s variabilným počtom odľahlých meraní. Výsledky vyrovnania všetkých robustných metód boli určené exaktne. Určenie exaktného výpočtu obmedzuje veľkosť generovaných sústav na počet meraní  $n=25$  a počet neznámych parametrov  $m=11$ .

### 9.1 Porovnanie výsledkov vyrovnania simulovaných dát metódou LMS s M-odhadmi

V tejto sekcii je popísané overenie výsledkov vyrovnania simulovaných dát metódou LMS a tá je následne porovnaná s výsledkami vyrovnania určenými vybranými metódami M-odhadov. Výpočet LMS je vykonaný na všetkých podmnožinách merania, pričom minimálna veľkosť podmnožiny je  $n \dots n/2 + 1$ . Výsledkom je podmnožina s minimálnym mediánom vyrovnaných rezíduí. Pre zhodnotenie výsledkov slúžia štyri vybrané príklady. Konfigurácie sústav a príslušný počet odľahlých meraní je popísaný v kapitole 9.1. V tabuľkách 9.2 a 9.3 sú uvedené hodnoty observácii  $\ell$ , generované skutočné chyby z normálneho rozdelenia (*Chyby N*) a generované hrubé chyby (*Chyby E*). Bez ujmy na všeobecnosti hrubé chyby prislúšia vždy prvým  $n$  meraniam.

Indexy observácii použité pre výsledný výpočet generovanej siete, pre ktoré platilo, že vykazujú minimálnu hodnotu mediánu sú uvedené v tabuľke 9.4. Pre danú kombináciu meraní má medián minimálnu hodnotu. Minimálne hodnoty mediánu pre každý príklad sú uvedené v tabuľke 9.5. Pretože merania sú ovplyvnené odchýlkami z množiny normálneho rozdelenia a hrubými chybami, tak aj výsledok vyrovnania sa bude nejakým spôsobom tými chybami ovplyvnený. V tabuľkách 9.6, 9.7, 9.8, 9.9 a 9.10 sú uvedené hodnoty rozdielov vyrovnaných neznámych parametrov konkrétnej použitej metódy od očakávaných hodnôt. Prvý riadok obsahuje

očekávané hodnoty vyrovnaných neznámych. Nasleduje vždy riadok s hodnotami vyrovnaných neznámych parametrov pre konkrétnu metódu *LS - metóda najmenších štvorcov*, *Huber - Huberov robustný odhad* a pod. Riadok označený ako *diff* označuje rozdiel vyrovnaných hodnôt danej metódy od očakávaných hodnôt.

Obrázky 9.1, 9.3 a 9.5 znázorňujú odchýlky výpočtu neznámych parametrov od očakávaných hodnôt overovaných metód. Na obrázkoch obr. 9.2, 9.4 a 9.6 sú znázornené hodnoty reziduí výsledného vyrovnania. Hodnoty reziduí výpočtu LMS sú určené z polovičnej množiny meraných dát.

Obrázky 9.7 a 9.8 ukazujú závislosť hodnoty mediánu na veľkosti vzoru sústavy. Výpočty naznačujú, že so znižovaním počtu observácií sa taktiež znižuje hodnota mediánu štvorcov reziduí. Na obrázkoch nie sú uvedené všetky hodnoty mediánov, ale iba prvá výpočítaná hodnota a následne vždy hodnota, ktorá bola menšia než posledná minimálna hodnota nejakého z predošlých výpočtov.

V príklade č.1 je meranie kontaminované hrubými chybami z 33%. Z tohto dôvodu je minimálna hodnota mediánu vzoru matice s jedným vylúčeným meraním stále vysoká. Medián postupne klesá so zvyšovaním počtu vylúčených meraní. Pri dosiahnutí vzoru matice, ktorý už neobsahuje merania kontaminované hrubými chybami je medián ovplyvnený len náhodne chybami z normálneho rozdelenia. Každopádne sa hodnota minimálneho mediánu vždy nachádzala vo vzore s najväčším počtom vylúčených meraní. To naznačuje, že pri riešení úlohy minimalizácie mediánu je optimálne pracovať s polovičným vzorom sústavy tzv. metóda *half-sample size*. Na druhú stranu pokiaľ, je ale kontaminácia meraných dát menšia než 50%, tak sa stratí časť informácie dát obsahujúcich iba náhodne rozdelené chyby. Pri voľbe vzoru sústavy 50% je možné metódu LMS použiť ako nástroj pre identifikáciu meraní podozrivých z odľahlosti a následne toto podozrenie potvrdiť, či vyvrátiť analýzou kovariančnej matice meraní, kovariančnej matice vyrovnaných neznámych, či testovaním štatistických hypotéz.

Na obrázkoch 9.9 a 9.10 sú znázornené odchýlky výpočtu neznámych parametrov od očakávaných hodnôt a hodnoty vyrovnaných reziduí pre merania, ktoré neobsahujú hrubé chyby. Všetky chyby patria do množiny s normálnym rozdelením a sú náhodne priradené observáciám. Z výsledkov je patrné, že metódy založené na M-odhadoch, prakticky kopírujú výpočet štandardnou metódou MNČ a to z toho dôvodu, že už po prvej iterácii bol výpočet neznámych parametrov dostatočne presný. V prípade metódy LMS bola pre dosiahnutie minimálneho mediánu odstránená značná časť meraní a to z už vyššie zmieneného dôvodu, že sa minimum nachádza vo vzore minimálnej veľkosti. Táto strata informácie určitým spôsobom ovplyvnila výsledný odhad. Odhad je aj napriek tomu pomerne presný. Z výsledkov vyplýva, že M-odhady narozdiel od LMS neovplyvnia odhad určenia neznámych parametrov v momente, keď observácie nie sú kontaminované odľahlými meraniami.





|           | Počet meraných<br>hodnôt | Počet neznámych<br>parametrov | Počet odľahlých<br>meraní |
|-----------|--------------------------|-------------------------------|---------------------------|
| Príklad 1 | 24                       | 9                             | 8                         |
| Príklad 2 | 21                       | 8                             | 10                        |
| Príklad 3 | 9                        | 3                             | 3                         |
| Príklad 4 | 20                       | 8                             | 0                         |

Tabuľka 9.1: Tabuľka parametrov overovaných sústav

| Index | Príklad 1  |          |            | Príklad 2  |          |            |
|-------|------------|----------|------------|------------|----------|------------|
|       | Chyby N    | Chyby E  | Observácie | Chyby N    | Chyby E  | Observácie |
| 1     | -0.0873282 | 21.0609  | 4745.61    | 0.0461322  | -135.038 | -2008.97   |
| 2     | -0.116861  | 57.0012  | -5889.58   | -0.0736332 | 30.697   | -81.6208   |
| 3     | -0.0872058 | -23.2176 | -824.519   | 0.0294172  | -56.2205 | 2010.32    |
| 4     | 0.00624455 | 52.3169  | -4172.12   | -0.160202  | -45.1058 | 4109.85    |
| 5     | 0.175395   | 9.09557  | -2526.53   | -0.0491202 | -38.5519 | -3474.29   |
| 6     | -0.0548515 | -48.2462 | -3654.47   | -0.0274426 | -28.8365 | 23.0719    |
| 7     | -0.1732    | 68.0205  | 1922.94    | 0.14814    | 93.6729  | -7100.61   |
| 8     | -0.0246107 | 18.0689  | -1261.13   | -0.0567556 | 4.87337  | -4556.3    |
| 9     | -0.0644235 | 0        | 2778.15    | -0.0316458 | 107.243  | -7085.57   |
| 10    | 0.0227204  | 0        | -4527.99   | -0.233111  | 150.139  | 5434.3     |
| 11    | -0.107959  | 0        | -526.487   | -0.20937   | 0        | -2124.31   |
| 12    | -0.0363303 | 0        | -7059.45   | 0.0573708  | 0        | -5274.43   |
| 13    | -0.0324868 | 0        | -1785.49   | -0.0605771 | 0        | -4531.55   |
| 14    | 0.0469576  | 0        | -2503.72   | -0.0618165 | 0        | -1528.94   |
| 15    | -0.122007  | 0        | 418.749    | 0.18308    | 0        | -380.697   |
| 16    | 0.0865931  | 0        | -12477.5   | 0.0799205  | 0        | -721.617   |
| 17    | 0.00743031 | 0        | -4334.64   | -0.0988204 | 0        | -932.242   |
| 18    | -0.0354738 | 0        | 1065.43    | -0.169952  | 0        | 7.82812    |
| 19    | 0.159858   | 0        | -3844.26   | 0.0975634  | 0        | -3377.24   |
| 20    | 0.0319147  | 0        | 1513.83    | 0.0203453  | 0        | 3452.42    |
| 21    | 0.219185   | 0        | -1030.51   | -0.122178  | 0        | 12625.3    |
| 22    | 0.0115669  | 0        | 5302.41    | -          | -        | -          |
| 23    | -0.100192  | 0        | 2885.5     | -          | -        | -          |
| 24    | -0.0762524 | 0        | -7086.85   | -          | -        | -          |

Tabuľka 9.2: Tabuľka rozdelenia chyb medzi observácie



| Index | Príklad 3  |          |            | Príklad 4  |         |            |
|-------|------------|----------|------------|------------|---------|------------|
|       | Chyby N    | Chyby E  | Observácie | Chyby N    | Chyby E | Observácie |
| 1     | -0.120739  | 95.8157  | 286.97     | 0.0756349  | 0       | 4077.74    |
| 2     | -0.0603478 | -27.9419 | -1487.42   | -0.11758   | 0       | 1019.77    |
| 3     | -0.0151315 | -61.3602 | -140.066   | -0.0707692 | 0       | 1437.76    |
| 4     | 0.154286   | 0        | 81.3524    | 0.116886   | 0       | -2410.92   |
| 5     | 0.0233461  | 0        | -1429.2    | -0.0505992 | 0       | 9824.69    |
| 6     | 0.140553   | 0        | 865.145    | -0.191494  | 0       | 604.354    |
| 7     | -0.0738034 | 0        | 969.088    | 0.00778378 | 0       | -1103.05   |
| 8     | 0.0340935  | 0        | 486.602    | -0.0437961 | 0       | -3532.9    |
| 9     | 0.0343839  | 0        | -290.754   | 0.127534   | 0       | -1009.44   |
| 10    | -          | -        | -          | 0.0284909  | 0       | 11104.9    |
| 11    | -          | -        | -          | 0.116324   | 0       | 8380.07    |
| 12    | -          | -        | -          | 0.0845787  | 0       | -6587.04   |
| 13    | -          | -        | -          | -0.0465377 | 0       | 1994.07    |
| 14    | -          | -        | -          | -0.0317248 | 0       | 606.921    |
| 15    | -          | -        | -          | -0.015944  | 0       | 4790.42    |
| 16    | -          | -        | -          | 0.268297   | 0       | -5762.13   |
| 17    | -          | -        | -          | -0.0454739 | 0       | 6251.6     |
| 18    | -          | -        | -          | 0.0888374  | 0       | 7724.21    |
| 19    | -          | -        | -          | 0.165209   | 0       | -7688.17   |
| 20    | -          | -        | -          | 0.203909   | 0       | -822.079   |

Tabuľka 9.3: Tabuľka rozdelenia chyb medzi observácie

|           |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-----------|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Príklad 1 | 9  | 10 | 11 | 12 | 13 | 15 | 16 | 17 | 18 | 20 | 22 | 23 | 24 |
| Príklad 2 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | -  | -  |
| Príklad 3 | 4  | 5  | 6  | 8  | 9  | -  | -  | -  | -  | -  | -  | -  | -  |
| Príklad 4 | 1  | 2  | 3  | 7  | 9  | 10 | 11 | 12 | 13 | 18 | 20 | -  | -  |

Tabuľka 9.4: Tabuľka indexov kombinácie observácií použitej pre výpočet neznámych parametrov



|           | Hodnota mediánu |
|-----------|-----------------|
| Príklad 1 | 6.56504e-05     |
| Príklad 2 | 0.000172475     |
| Príklad 3 | 0.000330773     |
| Príklad 4 | 6.45596e-07     |

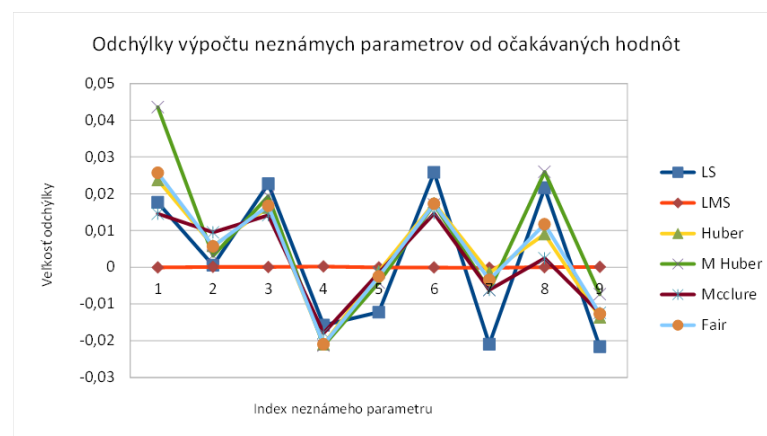
Tabuľka 9.5: Tabuľka minimálnych mediánov

| Expected | 1            | 2           | 3           | 4           |
|----------|--------------|-------------|-------------|-------------|
| LS       | 0.982388     | 1.99947     | 2.97731     | 4.01579     |
| Diff     | 0.0176118    | 0.000526634 | 0.0226873   | -0.0157932  |
| LMS      | 1.00006      | 1.99993     | 2.99996     | 3.99981     |
| Diff     | -5.75143e-05 | 7.20206e-05 | 3.79116e-05 | 0.000189509 |
| Huber    | 0.976211     | 1.99419     | 2.9838      | 4.02096     |
| Diff     | 0.0237888    | 0.00581327  | 0.0162044   | -0.0209608  |
| M Huber  | 0.956422     | 1.9969      | 2.98066     | 4.02134     |
| Diff     | 0.0435779    | 0.0031047   | 0.0193434   | -0.0213434  |
| Mcclure  | 0.985428     | 1.99053     | 2.98584     | 4.01792     |
| Diff     | 0.0145717    | 0.00947325  | 0.0141576   | -0.0179161  |
| Fair     | 0.974298     | 1.99433     | 2.98328     | 4.02099     |
| Diff     | 0.0257021    | 0.00567185  | 0.0167222   | -0.0209899  |

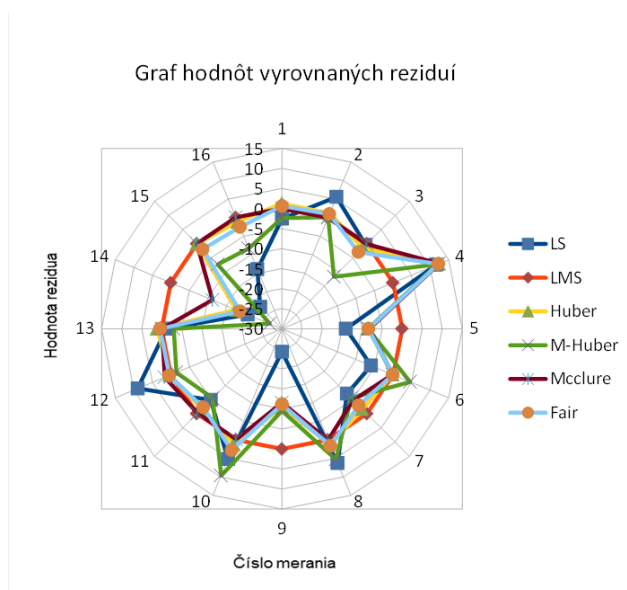
Tabuľka 9.6: Tabuľka odchýlok vyrovnaných neznámych parametrov od očakávaných hodnôt 1.časť. *Príklad 1*

| Expected | 5            | 6           | 7            | 8           | 9           |
|----------|--------------|-------------|--------------|-------------|-------------|
| LS       | 5.01226      | 5.9742      | 7.02099      | 7.9784      | 9.02165     |
| Diff     | -0.012261    | 0.0257962   | -0.0209859   | 0.0216038   | -0.021651   |
| LMS      | 5.00005      | 6.00008     | 7.00017      | 7.99999     | 8.99991     |
| Diff     | -4.89538e-05 | -8.4012e-05 | -0.000169984 | 7.49393e-06 | 8.52386e-05 |
| Huber    | 5.00113      | 5.98257     | 7.00153      | 7.99099     | 9.01364     |
| Diff     | -0.00112894  | 0.0174315   | -0.00153198  | 0.0090131   | -0.0136359  |
| M Huber  | 5.00419      | 5.98291     | 7.00626      | 7.97405     | 9.00737     |
| Diff     | -0.00419067  | 0.017089    | -0.00625778  | 0.0259488   | -0.00736835 |
| Mcclure  | 5.00151      | 5.98536     | 7.00618      | 7.99757     | 9.01246     |
| Diff     | -0.00150943  | 0.0146433   | -0.00617616  | 0.00242551  | -0.0124616  |
| Fair     | 5.00249      | 5.98277     | 7.00342      | 7.9883      | 9.0127      |
| Diff     | -0.0024892   | 0.0172251   | -0.00342316  | 0.011701    | -0.0126994  |

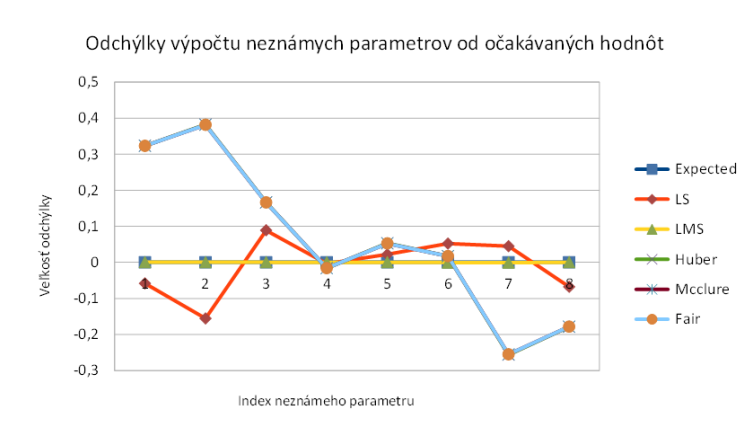
Tabuľka 9.7: Tabuľka odchýlok vyrovnaných neznámych parametrov od očakávaných hodnôt 2.časť. *Príklad 1*



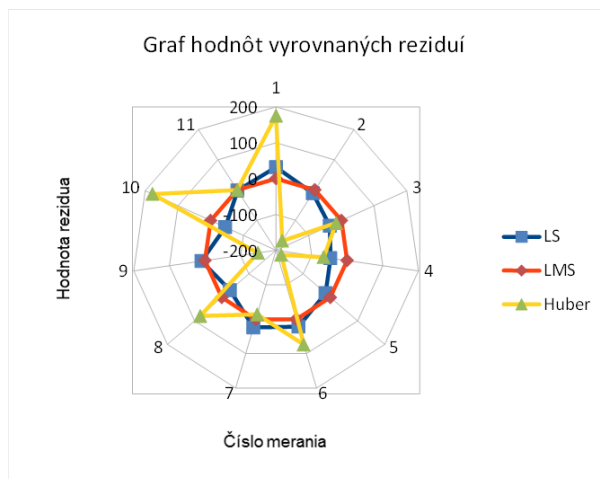
Obrázok 9.1: Odchýlky výpočtu neznámych parametrov od očakávaných hodnôt *Príklad 1*



Obrázok 9.2: Graf hodnôt vyrovnaných reziduí *Príklad 1*



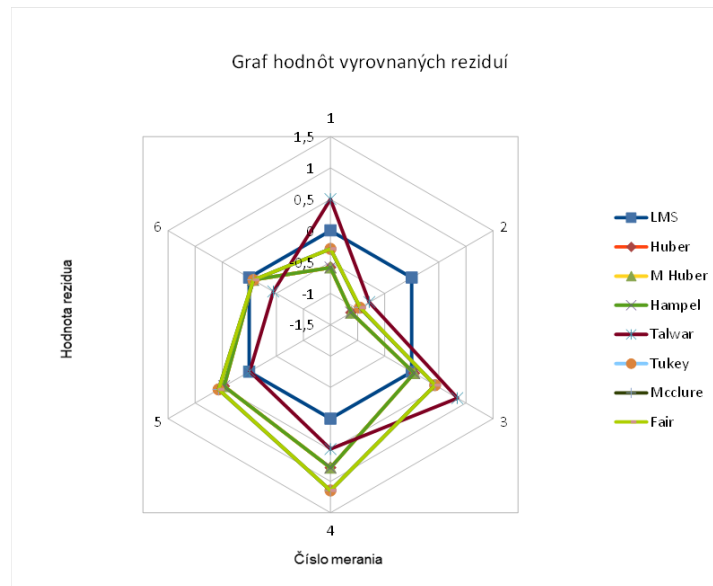
Obrázok 9.3: Odchýlky výpočtu neznámych parametrov od očakávaných hodnôt  
*Príklad 2*



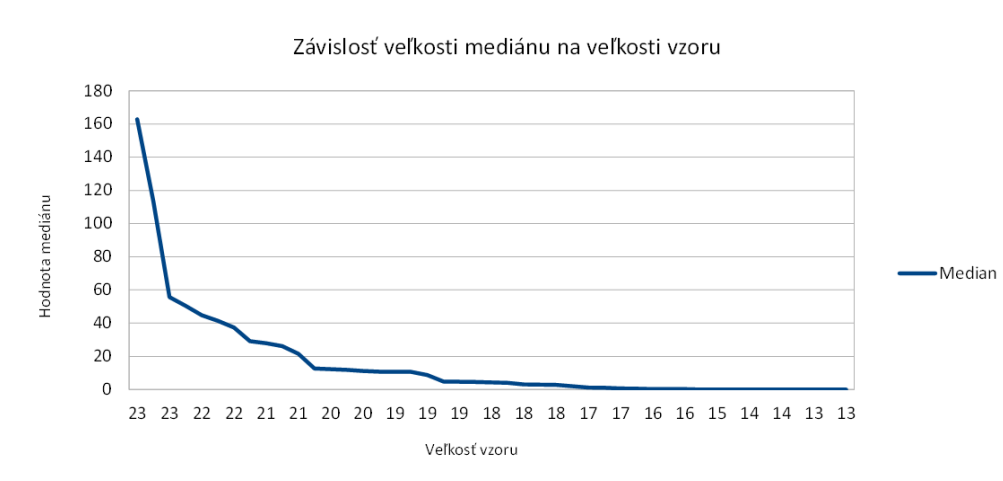
Obrázok 9.4: Graf hodnôt vyrovnaných reziduí *Príklad 2*



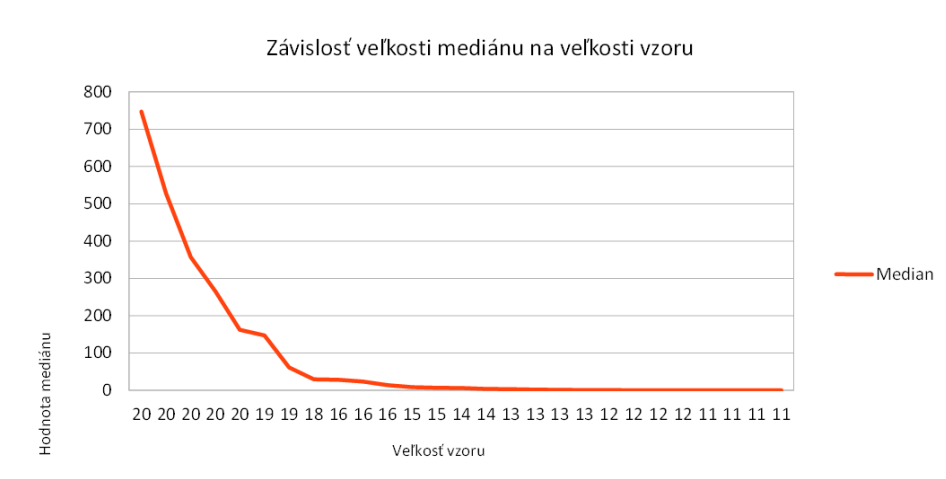
Obrázok 9.5: Odchýlky výpočtu neznámych parametrov od očakávaných hodnôt *Príklad 3*



Obrázok 9.6: Graf hodnôt vyrovnaných reziduí *Príklad 3*

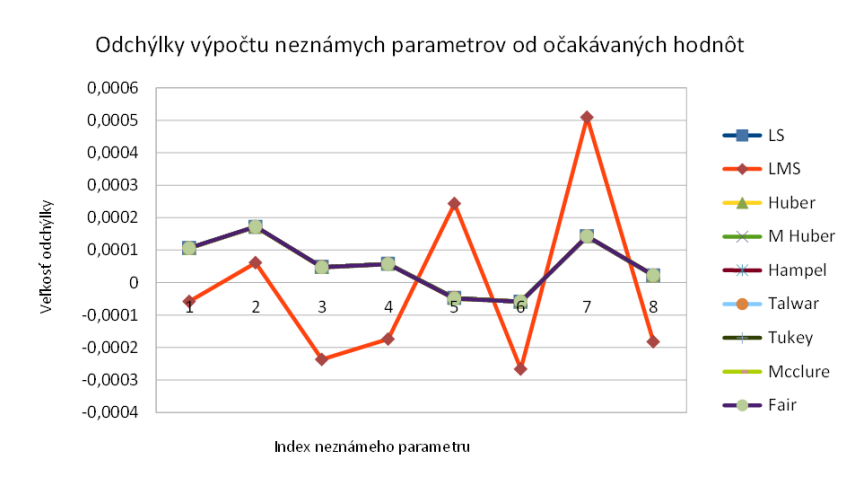


Obrázok 9.7: Závislosť hodnoty mediánu na počtosti meraných hodnôt vzoru sústavy *Príklad 1*

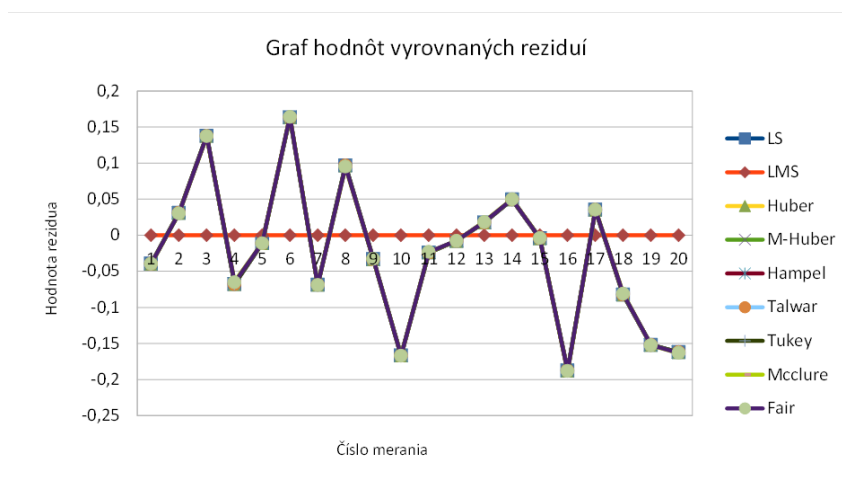


Obrázok 9.8: Závislosť hodnoty mediánu na počtosti meraných hodnôt vzoru sústavy *Príklad 2*





Obrázok 9.9: Odchýlky výpočtu neznámých parametrov od očakávaných hodnôt  
*Príklad 4*



Obrázok 9.10: Graf hodnôt vyrovnaných reziduí *Príklad 2*



| Expected | 1           | 2           | 3           | 4            | 5           | 6           | 7            | 8           |
|----------|-------------|-------------|-------------|--------------|-------------|-------------|--------------|-------------|
| LS       | 1.05848     | 2.15555     | 2.9114      | 4.00381      | 4.97768     | 5.94813     | 6.95518      | 8.06783     |
| Diff     | -0.0584843  | -0.155546   | 0.0886041   | -0.00381359  | 0.022321    | 0.0518736   | 0.0448159    | -0.0678324  |
| LMS      | 0.999518    | 1.99947     | 2.99971     | 4.00027      | 4.99975     | 6.00032     | 7.00063      | 7.99964     |
| Diff     | 0.000482471 | 0.000526452 | 0.000292159 | -0.000267296 | 0.000247407 | -0.00032222 | -0.000633844 | 0.000356092 |
| Huber    | 0.676542    | 1.61778     | 2.83368     | 4.01621      | 4.94685     | 5.98279     | 7.25638      | 8.1788      |
| Diff     | 0.323458    | 0.382219    | 0.166321    | -0.0162144   | 0.0531487   | 0.017209    | -0.256379    | -0.1788     |
| Mcclure  | 0.676813    | 1.61864     | 2.83387     | 4.01601      | 4.94767     | 5.98284     | 7.25499      | 8.17819     |
| Diff     | 0.323187    | 0.381365    | 0.166129    | -0.0160107   | 0.0523303   | 0.0171552   | -0.254992    | -0.178186   |
| Fair     | 0.676808    | 1.61863     | 2.83387     | 4.01601      | 4.94767     | 5.98285     | 7.25499      | 8.17819     |
| Diff     | 0.323192    | 0.38137     | 0.166128    | -0.0160084   | 0.0523297   | 0.0171535   | -0.254994    | -0.178186   |

Tabuľka 9.8: Tabuľka odchýlok vyrovnaných neznámych parametrov od očakávaných hodnôt *Príklad 2*



| Expected | 1           | 2           | 3            |
|----------|-------------|-------------|--------------|
| LS       | 1.08419     | 2.05046     | 3.05176      |
| Diff     | -0.0841883  | -0.0504634  | -0.0517634   |
| LMS      | 0.999896    | 1.99972     | 3.00015      |
| Diff     | 0.000103928 | 0.000277534 | -0.000148525 |
| Huber    | 1.0019      | 2.00254     | 3.00212      |
| Diff     | -0.00189705 | -0.00253796 | -0.00212018  |
| M Huber  | 0.998882    | 1.9997      | 3.0016       |
| Diff     | 0.00111808  | 0.000295996 | -0.00159694  |
| Hampel   | 0.998945    | 1.9998      | 3.00163      |
| Diff     | 0.00105545  | 0.000201807 | -0.0016334   |
| Talwar   | 0.998945    | 1.9998      | 3.00163      |
| Diff     | 0.00105545  | 0.000201807 | -0.0016334   |
| Tukey    | 1.00169     | 2.00179     | 3.00245      |
| Diff     | -0.00168703 | -0.00178526 | -0.00245436  |
| Mcclure  | 1.00169     | 2.00178     | 3.00245      |
| Diff     | -0.00168884 | -0.00178333 | -0.00245264  |
| Fair     | 1.00169     | 2.00178     | 3.00245      |
| Diff     | -0.00168814 | -0.00177606 | -0.00244644  |

Tabuľka 9.9: Tabuľka odchýlok vyrovnaných neznámych parametrov od očakávaných hodnôt *Príklad 3*



| Expected | 1            | 2           | 3            | 4           | 5            | 6            | 7           | 8            |
|----------|--------------|-------------|--------------|-------------|--------------|--------------|-------------|--------------|
| LS       | 0.999894     | 1.99983     | 2.99995      | 3.99994     | 5.00005      | 6.00006      | 6.99986     | 7.99998      |
| Diff     | 0.000106127  | 0.000171879 | 4.80009e-05  | 5.75832e-05 | -4.7437e-05  | -5.88926e-05 | 0.000142654 | 2.19074e-05  |
| LMS      | 1.00006      | 1.99994     | 3.00024      | 4.00017     | 4.99976      | 6.00027      | 6.99949     | 8.00018      |
| Diff     | -5.80561e-05 | 6.10338e-05 | -0.000236797 | -0.00017389 | 0.000243459  | -0.000266531 | 0.000509034 | -0.000182333 |
| Huber    | 0.999894     | 1.99983     | 2.99995      | 3.99994     | 5.00005      | 6.00006      | 6.99986     | 7.99998      |
| Diff     | 0.000106127  | 0.000171878 | 4.80007e-05  | 5.75836e-05 | -4.74372e-05 | -5.8893e-05  | 0.000142655 | 2.19071e-05  |
| M Huber  | 0.999894     | 1.99983     | 2.99995      | 3.99994     | 5.00005      | 6.00006      | 6.99986     | 7.99998      |
| Diff     | 0.000106127  | 0.000171878 | 4.80007e-05  | 5.75836e-05 | -4.74372e-05 | -5.8893e-05  | 0.000142655 | 2.19071e-05  |
| Hampel   | 0.999894     | 1.99983     | 2.99995      | 3.99994     | 5.00005      | 6.00006      | 6.99986     | 7.99998      |
| Diff     | 0.000106127  | 0.000171878 | 4.80007e-05  | 5.75836e-05 | -4.74372e-05 | -5.8893e-05  | 0.000142655 | 2.19071e-05  |
| Talwar   | 0.999894     | 1.99983     | 2.99995      | 3.99994     | 5.00005      | 6.00006      | 6.99986     | 7.99998      |
| Diff     | 0.000106127  | 0.000171878 | 4.80007e-05  | 5.75836e-05 | -4.74372e-05 | -5.8893e-05  | 0.000142655 | 2.19071e-05  |
| Tukey    | 0.999894     | 1.99983     | 2.99995      | 3.99994     | 5.00005      | 6.00006      | 6.99986     | 7.99998      |
| Diff     | 0.000106121  | 0.000171904 | 4.79161e-05  | 5.75424e-05 | -4.75604e-05 | -5.88576e-05 | 0.000142659 | 2.19929e-05  |
| Mcclure  | 0.999894     | 1.99983     | 2.99995      | 3.99994     | 5.00005      | 6.00006      | 6.99986     | 7.99998      |
| Diff     | 0.000105953  | 0.000172364 | 4.61378e-05  | 5.66915e-05 | -5.01482e-05 | -5.80594e-05 | 0.000142738 | 2.37431e-05  |
| Fair     | 0.999894     | 1.99983     | 2.99995      | 3.99994     | 5.00005      | 6.00006      | 6.99986     | 7.99998      |
| Diff     | 0.000106321  | 0.000172394 | 4.70467e-05  | 5.70345e-05 | -4.94721e-05 | -5.84232e-05 | 0.000142956 | 2.27606e-05  |

Tabuľka 9.10: Tabuľka odchýlok vyrovnaných neznámych parametrov od očakávaných hodnôt *Príklad 4*

|           | Počet meraných<br>hodnôt | Počet neznámych<br>parametrov | Počet odľahlých<br>meraní |
|-----------|--------------------------|-------------------------------|---------------------------|
| Príklad 1 | 25                       | 9                             | 10                        |
| Príklad 2 | 20                       | 7                             | 1                         |

Tabuľka 9.11: Tabuľka parametrov sústav porovnania metód LMS a LTS

## 9.2 Porovnanie výsledkov vyrovnania simulovaných dát metódou LMS s metódou LTS

V tejto sekcii je popísané zhodnotenie výsledkov vyrovnania simulovaných dát metódou LMS a tá je následne porovnaná s výsledkami vyrovnania metódou LTS. Výpočet LMS a LTS je vykonaný na všetkých podmnožinách merania, pričom minimálna veľkosť podmnožiny je v oboch prípadoch  $n \dots n/2 + 1$ . Výsledkom je podmnožina s minimálnym mediánom resp. minimálnou sumou štvorcov vyrovnaných reziduí. K zhodnoteniu výsledkov slúžia dve vybrané príklady. Konfigurácie sústav a príslušný počet odľahlých meraní je popísaný v kapitole 9.11. V tabuľke 9.12 sú obdobne ako v predošlej sekcii uvedené hodnoty observácii  $\ell$ , generované skutočné chyby z normálneho rozdelenia (*Chyby*  $n$ ) a generované hrubé chyby (*Chyby*  $E$ ). Bez ujmy na všeobecnosti sú hrubé chyby pričítané vždy k prvým  $n$  meraniam. Indexy observácii použité pre výsledný výpočet generovanej siete, pre ktoré platilo, že vykazujú minimálnu hodnotu mediánu resp. minimálnu hodnotu sumy štvorcov vyrovnaných rezidui sú uvedené v tabuľke 9.13. Pre každú výslednú kombináciu meraní je v tabuľke 9.14 uvedený medián a suma štvorcov rezidui s minimálnou hodnotou. V tabuľkách 9.15, 9.16 a 9.17 sú uvedené hodnoty rozdielov vyrovnaných neznámych parametrov konkrétnej použitej metódy od očakávaných hodnôt. Prvý riadok obsahuje očakávané hodnoty vyrovnaných neznámych. Nasleduje vždy riadok s hodnotami vyrovnaných neznámych parametrov pre konkrétnu metódu. Riadok označený ako *diff* označuje rozdiel vyrovnaných hodnôt danej metódy od očakávaných hodnôt. Výsledky porovnania metód LMS a LTS ukazujú, že v oboch prípadoch sa podarilo správne vylúčiť množinu odľahlých meraní. Na základe definície veľkosti množiny observácii, ktorá má byť z celkového súboru meraní odstránená, bola v závislosti na veľkosti mediánu a veľkosti sumy štvorcov reziduí odstránená množina dát, ktorá obsahovala iba náhodné chyby z normalného rozdelenia. V závislosti na použitej metóde boli odstránené rôzne observácie. Odhad neznámych parametrov bol v oboch prípadoch prakticky rovnaký. Vo vybraných vzorových prípadoch, je odhad určený metódou LTS o niečo presnejší avšak prakticky to všeobecne neplatí.



| Index | Príklad 1  |           |            | Príklad 2  |         |            |
|-------|------------|-----------|------------|------------|---------|------------|
|       | Chyby N    | Chyby E   | Observácie | Chyby N    | Chyby E | Observácie |
| 1     | -0.127687  | -50.3227  | 3500.69    | -0.059771  | -38.603 | 4209.57    |
| 2     | 0.125069   | 45.1725   | 1353.36    | -0.0299922 | 0       | -4295.89   |
| 3     | -0.0876324 | 16.0371   | -300.499   | 0.0735312  | 0       | -3866.63   |
| 4     | -0.0542337 | -18.9577  | 3907.28    | -0.116647  | 0       | -4016.84   |
| 5     | -0.0688171 | 70.3158   | -397.333   | 0.0126235  | 0       | 8457.79    |
| 6     | 0.00449391 | -0.794465 | 3657.56    | -0.109634  | 0       | 814.516    |
| 7     | -0.0157909 | 44.1803   | 5516.75    | -0.104983  | 0       | 3624.01    |
| 8     | -0.077588  | 16.2834   | 7483.81    | 0.0165295  | 0       | 5969.6     |
| 9     | -0.0605913 | 61.3784   | 4518.15    | 0.125953   | 0       | -4597.81   |
| 10    | 0.0280251  | 25.6488   | 5378.11    | 0.104054   | 0       | -2161.2    |
| 11    | -0.0591104 | 0         | 7693.11    | -0.0344407 | 0       | -2064.14   |
| 12    | 0.14939    | 0         | -377.222   | -0.148074  | 0       | 3418.84    |
| 13    | -0.0279798 | 0         | 6384.91    | 0.0170143  | 0       | -3791.15   |
| 14    | 0.00502122 | 0         | -4439.49   | -0.0768646 | 0       | 1171.95    |
| 15    | 0.00918844 | 0         | 5645.75    | -0.127195  | 0       | -1911.62   |
| 16    | 0.0716791  | 0         | 6107.12    | -0.0452412 | 0       | 1725.14    |
| 17    | 0.0474503  | 0         | -2669.37   | -0.232125  | 0       | 3003.92    |
| 18    | 0.0150418  | 0         | 0.461498   | 0.109011   | 0       | 6922.5     |
| 19    | -0.183634  | 0         | -7481.18   | 0.069995   | 0       | 2200.69    |
| 20    | -0.0517068 | 0         | -7552.22   | -0.022412  | 0       | -1535.11   |
| 21    | -0.0666877 | 0         | 1716.94    | -          | -       | -          |
| 22    | -0.146574  | 0         | -8702.71   | -          | -       | -          |
| 23    | -0.0985342 | 0         | -5711.39   | -          | -       | -          |
| 24    | -0.101347  | 0         | -12167.5   | -          | -       | -          |
| 25    | -0.0892014 | 0         | 2338.37    | -          | -       | -          |

Tabuľka 9.12: Tabuľka rozdelenia chyb medzi observácie

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|
| LMS 1 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 20 | 22 | 23 | 24 | 25 |
| LTS 1 | 11 | 12 | 14 | 15 | 16 | 17 | 18 | 20 | 21 | 22 | 23 | 24 | 25 |
| LMS 2 | 2  | 3  | 4  | 6  | 8  | 10 | 11 | 13 | 14 | 15 | 18 |    |    |
| LTS 2 | 4  | 5  | 7  | 8  | 9  | 12 | 14 | 15 | 16 | 17 | 20 |    |    |

Tabuľka 9.13: Tabuľka indexov kombinácie observácií použitej pre výpočet neznámych parametrov



|           | Medián      | Suma štvorcov |
|-----------|-------------|---------------|
| Príklad 1 | 0.00013409  | 0.00940044    |
| Príklad 2 | 1.29191e-05 | 0.000798888   |

Tabuľka 9.14: Tabuľka minimálnych mediánov a minimálnych sum štvorcov

| Expected | 1           | 2           | 3            | 4            |
|----------|-------------|-------------|--------------|--------------|
| LMS      | 0.99994     | 1.99984     | 3.00018      | 4.00018      |
| Diff     | 5.99532e-05 | 0.000160537 | -0.000183928 | -0.000178716 |
| LTS      | 0.999908    | 1.99986     | 3.00023      | 4.00023      |
| Diff     | 9.20041e-05 | 0.000140349 | -0.000234055 | -0.000229098 |

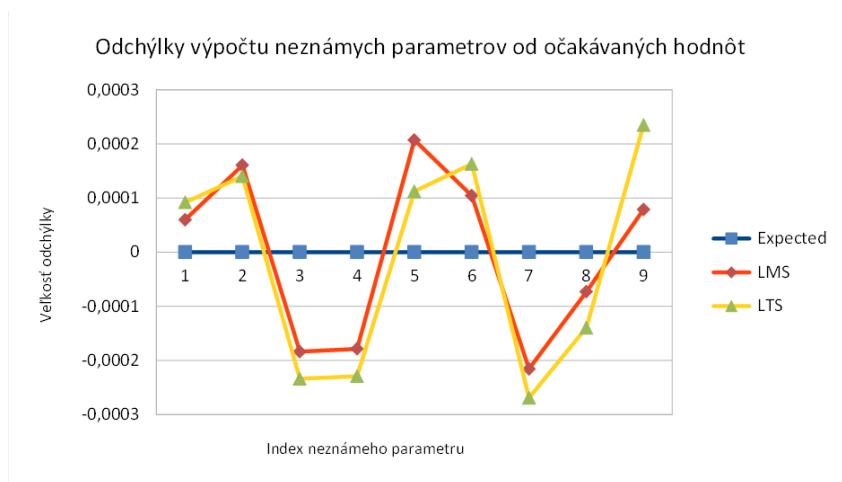
Tabuľka 9.15: Tabuľka odchýlok vyrovnaných neznámych parametrov od očakávaných hodnôt 1.časť. *Príklad 1*

| Expected | 5           | 6           | 7            | 8            | 9           |
|----------|-------------|-------------|--------------|--------------|-------------|
| LMS      | 4.99979     | 5.9999      | 7.00022      | 8.00007      | 8.99992     |
| Diff     | 0.000207135 | 0.000104256 | -0.000215808 | -7.31471e-05 | 7.89315e-05 |
| LTS      | 4.99989     | 5.99984     | 7.00027      | 8.00014      | 8.99977     |
| Diff     | 0.00011243  | 0.000163173 | -0.00026908  | -0.00013952  | 0.00023498  |

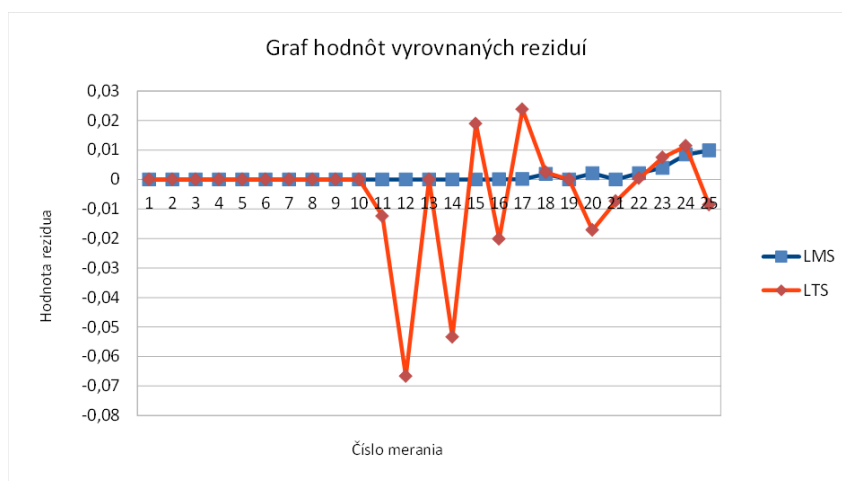
Tabuľka 9.16: Tabuľka odchýlok vyrovnaných neznámych parametrov od očakávaných hodnôt 2.časť. *Príklad 1*

| Expected | 1            | 2           | 3            | 4            | 5           | 6           | 7            |
|----------|--------------|-------------|--------------|--------------|-------------|-------------|--------------|
| LMS      | 1.00037      | 1.99983     | 3.00101      | 4.00079      | 4.99937     | 5.99932     | 6.9999       |
| Diff     | -0.00037413  | 0.000173246 | -0.00100557  | -0.000790349 | 0.000627773 | 0.000679055 | 9.82721e-05  |
| LTS      | 1.00029      | 2.00001     | 3.00002      | 4.00024      | 4.99962     | 5.99992     | 7.00049      |
| Diff     | -0.000294351 | -9.5169e-06 | -1.82132e-05 | -0.0002438   | 0.000382777 | 8.28634e-05 | -0.000485559 |

Tabuľka 9.17: Tabuľka odchýlok vyrovnaných neznámych parametrov od očakávaných hodnôt *Príklad 2*

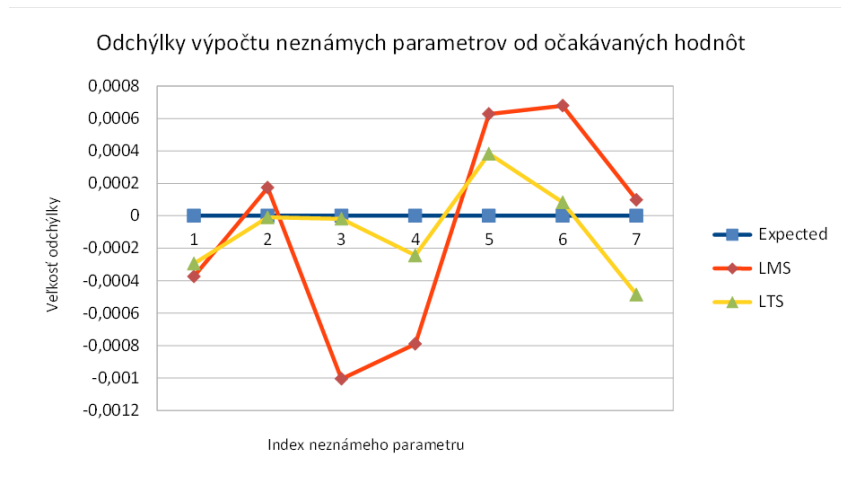


Obrázok 9.11: Odchýlky výpočtu neznámých parametrov od očakávaných hodnôt  
*Príklad 1*

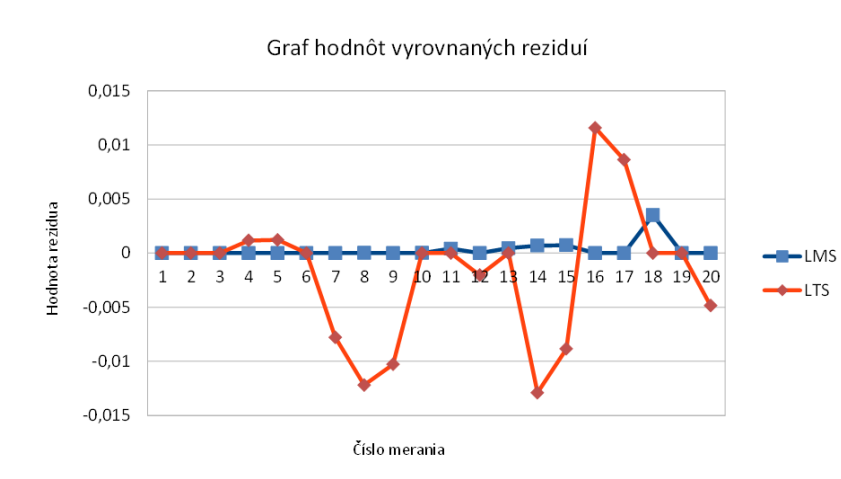


Obrázok 9.12: Graf hodnôt vyrovnaných rezíduí *Príklad 1*





Obrázok 9.13: Odchýlky výpočtu neznámych parametrov od očakávaných hodnôt  
*Príklad 2*



Obrázok 9.14: Graf hodnôt vyrovnaných reziduí *Príklad 2*

|           | Počet meraných<br>hodnôt | Počet neznámych<br>parametrov | Počet odľahlých<br>meraní |
|-----------|--------------------------|-------------------------------|---------------------------|
| Príklad 1 | 25                       | 11                            | 3                         |
| Príklad 2 | 24                       | 9                             | 10                        |

Tabuľka 9.18: Tabuľka parametrov sústav porovnania metód LMS a RANSAC

### 9.3 Porovnanie výsledkov vyrovnania simulovaných dát metódou LMS s metódou RANSAC

V tejto sekcii je popísané zhodnotenie výsledkov vyrovnania simulovaných dát metódou LMS a tá je následne porovnaná s výsledkami vyrovnania metódou RANSAC. Výpočet LMS je vykonaný na všetkých podmnožinách merania, pričom minimálna veľkosť podmnožiny je v oboch prípadoch  $n \dots n/2 + 1$ . Výpočty metódou RANSAC sú vykonané na všetkých možných podmnožinách minimálnej možnej veľkosti t.j.  $n/2 + 1$ . Výsledkom je podmnožina s minimálnym mediánom resp. s maximálnym počtom meraní patriacich do množiny zhody. K zhodnoteniu výsledkov slúžia dve vybrané príklady. Konfigurácie sústav a príslušný počet odľahlých meraní je popísaný v tabuľke 9.18. V tabuľke 9.19 sú obdobne ako v predošlých sekciách uvedené hodnoty observácií  $\ell$ , generované skutočné chyby z normálneho rozdelenia (*Chyby N*) a generované hrubé chyby (*Chyby E*). Bez ujmy na všeobecnosti hrubé chyby sú priradené vždy prvým  $n$  meraniam. Indexy observácií použité pre výsledný výpočet generovanej siete, pre ktoré platilo, že vykazujú minimálnu hodnotu mediánu vyrovnaných reziduí resp. maximálny počet meraní, ktoré patria do množiny zhody sú uvedené v tabuľke 9.20. Pre výsledné kombinácie meraní oboch príkladov je v tabuľke 9.21 určený medián s minimálnou hodnotou a maximálna veľkosť množiny zhody. V tabuľkách 9.22, 9.23, 9.24 a 9.25 sú uvedené hodnoty rozdielov vyrovnaných neznámych parametrov konkrétnej použitej metódy od očakávaných hodnôt. Prvý riadok obsahuje očakávané hodnoty vyrovnaných neznámych. Nasleduje vždy riadok s hodnotami vyrovnaných neznámych parametrov pre konkrétnu metódu. Riadok označený ako *diff* označuje rozdiel vyrovnaných hodnôt danej metódy od očakávaných hodnôt. Výsledky tejto sekcie ukazujú, že metóda RANSAC správne odhalila kontaminované merania bez ohľadu na ich počet. V porovnaní s metódou LMS je odhad neznámych parametrov určený metódou RANSAC presnejší. Dôvodom je to, že metóda nevyhlúčila neodľahlé observácie. Vyrovnané opravy korelujú so skutočnými odchýlkami. Úplný výpočet metódy RANSAC je o niečo efektívnejší než výpočet metódy LMS.



| Index | Príklad 1  |          |            | Príklad 2   |          |            |
|-------|------------|----------|------------|-------------|----------|------------|
|       | Chyby N    | Chyby E  | Observácie | Chyby N     | Chyby E  | Observácie |
| 1     | 0.0322263  | 88.2051  | -3727.19   | 0.00174341  | 49.3357  | -1431.8    |
| 2     | 0.10165    | -43.4079 | 2355.73    | -0.0572566  | -41.1055 | 5058.91    |
| 3     | 0.0244993  | -15.2781 | -608.862   | -0.0712272  | -70.7789 | -1987.53   |
| 4     | 0.0420459  | 0        | 1956.72    | -0.0857888  | -17.6974 | -4040.2    |
| 5     | -0.0773189 | 0        | 8308.39    | 0.108922    | 56.8309  | -3699.87   |
| 6     | 0.015574   | 0        | -12902.3   | -0.120809   | 52.2796  | -13210.8   |
| 7     | -0.0804418 | 0        | -3309.79   | 0.0291435   | 17.3194  | -3455.19   |
| 8     | -0.0911338 | 0        | 10974.7    | 0.10166     | 55.5976  | -578.455   |
| 9     | -0.0915947 | 0        | 10789.3    | -0.164208   | 62.0113  | 5572.86    |
| 10    | -0.161775  | 0        | -4268.08   | 0.0809664   | 27.1208  | 4110.22    |
| 11    | -0.0256788 | 0        | -15793.2   | -0.0610053  | 0        | -2251.51   |
| 12    | 0.0378838  | 0        | 5717.83    | -0.068363   | 0        | -1476.51   |
| 13    | 0.00274992 | 0        | 7413.15    | -0.0748838  | 0        | -8556.7    |
| 14    | -0.113342  | 0        | 7695.25    | 0.00171085  | 0        | -6474.42   |
| 15    | -0.0532245 | 0        | 8674.24    | 0.170779    | 0        | -9820.32   |
| 16    | -0.161018  | 0        | 862.605    | 0.0359357   | 0        | 539.438    |
| 17    | -0.0895896 | 0        | -5302.31   | -0.00733599 | 0        | -5611.88   |
| 18    | 0.172689   | 0        | -9931.23   | -0.00159809 | 0        | -3273.13   |
| 19    | 0.0111849  | 0        | 6041.13    | -0.0445025  | 0        | 2766.46    |
| 20    | 0.0477196  | 0        | 4080.82    | 0.0765002   | 0        | 5532.16    |
| 21    | -0.0452905 | 0        | -12222.6   | -0.0132877  | 0        | 1972.33    |
| 22    | 0.0712828  | 0        | 7326.11    | -0.00107589 | 0        | -3382.37   |
| 23    | -0.18841   | 0        | 230.759    | 0.0236574   | 0        | 524.901    |
| 24    | 0.0573203  | 0        | -2672.68   | 0.13573     | 0        | -1690.23   |
| 25    | 0.0456031  | 0        | -6651.44   | -           | -        |            |

Tabuľka 9.19: Tabuľka rozdelenia chyb medzi observácie



|          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| LMS 1    | 4  | 5  | 7  | 9  | 12 | 13 | 15 | 17 | 18 | 20 | 21 | 22 | 25 | -  | -  | -  | -  | -  | -  |    |
| RANSAC 1 | 4  | 5  | 6  | 7  | 8  | 9  | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 24 | 25 |
| LMS 2    | 11 | 12 | 13 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | -  | -  | -  | -  | -  | -  | -  |
| RANSAC 2 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | -  | -  | -  | -  | -  | -  |

Tabuľka 9.20: Tabuľka indexov kombinácie observácií použitej pre výpočet neznámych parametrov

|           | Medián      | Veľkosť množiny zhody |
|-----------|-------------|-----------------------|
| Príklad 1 | 1.31531e-09 | 20                    |
| Príklad 2 | 0.000304886 | 14                    |

Tabuľka 9.21: Tabuľka minimálnych mediánov a minimálnych sum štvorcov

| Expected | 1           | 2            | 3           | 4           | 5            |
|----------|-------------|--------------|-------------|-------------|--------------|
| LMS      | 0.999675    | 2.00048      | 2.99946     | 3.9995      | 5.00043      |
| Diff     | 0.000324766 | -0.000482699 | 0.000536565 | 0.000503385 | -0.000425159 |
| RANSAC   | 0.999965    | 2.00004      | 2.99992     | 3.99989     | 5.00017      |
| Diff     | 3.457e-05   | -3.58142e-05 | 8.03778e-05 | 0.000114283 | -0.000165131 |

Tabuľka 9.22: Tabuľka odchýlok vyrovnaných neznámych parametrov od očakávaných hodnôt *Príklad 1*

| Expected | 6            | 7            | 8            | 9            | 10          | 11          |
|----------|--------------|--------------|--------------|--------------|-------------|-------------|
| LMS      | 6.00004      | 7.0002       | 8.00005      | 9.00015      | 9.99946     | 10.9997     |
| Diff     | -4.07514e-05 | -0.000204822 | -5.31031e-05 | -0.000149526 | 0.000543662 | 0.000252115 |
| RANSAC   | 6.00004      | 7.00003      | 7.99999      | 9.0001       | 9.99981     | 10.9999     |
| Diff     | -3.60654e-05 | -3.26538e-05 | 7.5117e-06   | -0.000101373 | 0.000193729 | 7.69209e-05 |

Tabuľka 9.23: Tabuľka odchýlok vyrovnaných neznámych parametrov od očakávaných hodnôt *Príklad 1*

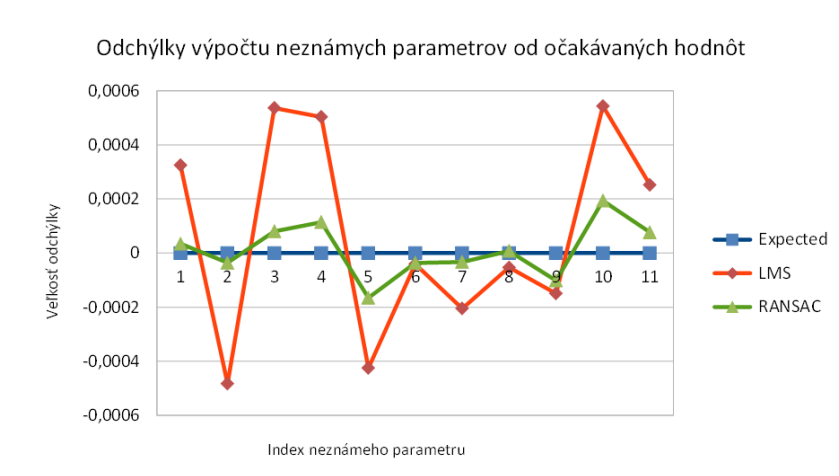


| Expected | 1           | 2           | 3            | 4           |
|----------|-------------|-------------|--------------|-------------|
| LMS      | 0.999955    | 1.99949     | 3.00027      | 3.99981     |
| Diff     | 4.52169e-05 | 0.000509247 | -0.000268189 | 0.000189663 |
| RANSAC   | 0.999894    | 2.00004     | 3.00024      | 4.00002     |
| Diff     | 0.000106196 | -3.9895e-05 | -0.000242826 | -1.9332e-05 |

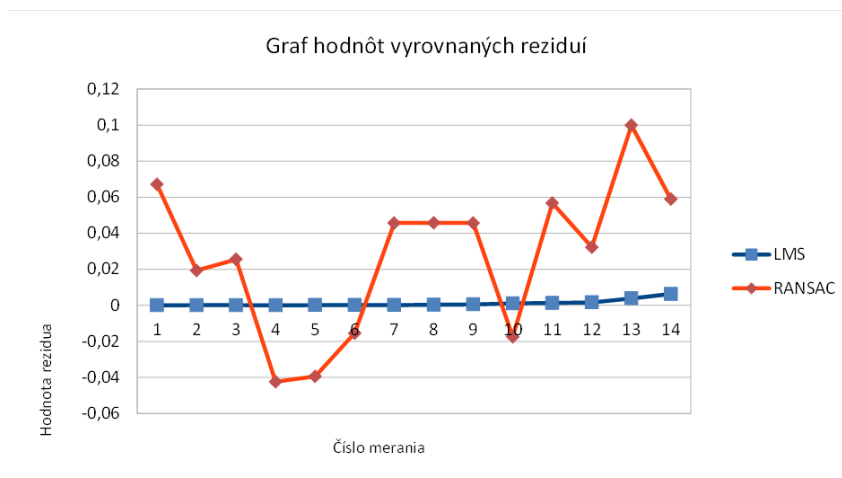
Tabuľka 9.24: Tabuľka odchýlok vyrovnaných neznámych parametrov od očakávaných hodnôt *Príklad 2*

| Expected | 5           | 6           | 7            | 8            | 9            |
|----------|-------------|-------------|--------------|--------------|--------------|
| LMS      | 4.99964     | 6.00023     | 7.00068      | 8.00007      | 9.0004       |
| Diff     | 0.000358409 | -0.00022958 | -0.000681952 | -7.46366e-05 | -0.000404531 |
| RANSAC   | 4.99997     | 5.99981     | 6.99989      | 7.99988      | 8.9998       |
| Diff     | 2.51294e-05 | 0.000186615 | 0.000108234  | 0.000124759  | 0.00020221   |

Tabuľka 9.25: Tabuľka odchýlok vyrovnaných neznámych parametrov od očakávaných hodnôt *Príklad 2*



Obrázok 9.15: Odchýlky výpočtu neznámych parametrov od očakávaných hodnôt *Príklad 1*

Obrázok 9.16: Graf hodnôt vyrovnaných reziduí *Príklad 1*Obrázok 9.17: Odchýlky výpočtu neznámych parametrov od očakávaných hodnôt  
*Príklad 2*Obrázok 9.18: Graf hodnôt vyrovnaných reziduí *Príklad 2*

## 9.4 Zhodnotenie efektivity návrhu paralelného výpočtu LMS

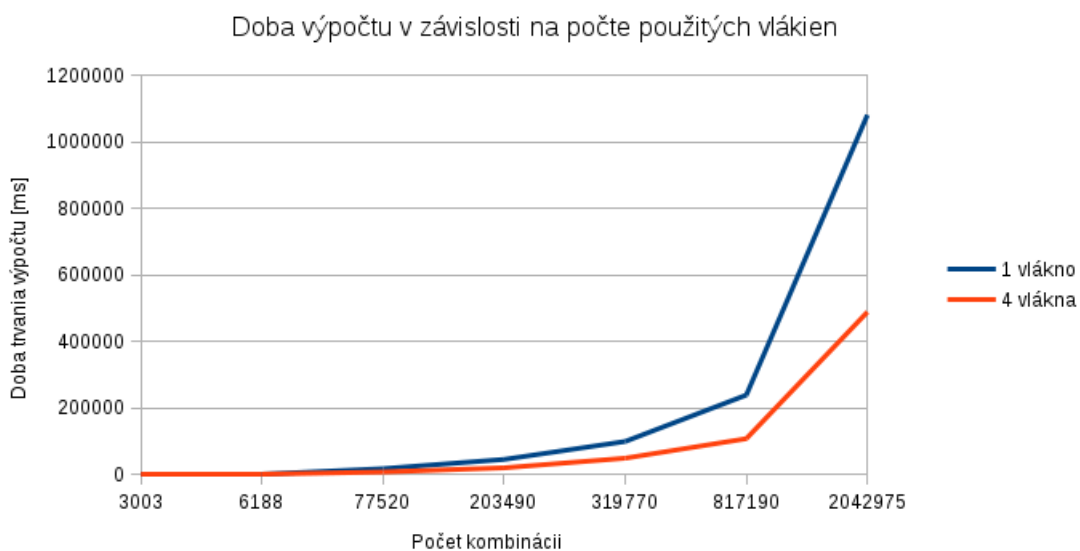
Zhodnotenie efektivity paralelného výpočtu LMS spočíva v porovnaní časov výpočtu za použitia jedného a  $n$  vlákien. Čas výpočtu je určený rozdielom časov konca a začatia výpočtu. Na určenie času bola použitá knižnica C++11 `chrono`. Testy boli vykonané na príkladoch s plnými maticami a niekoľkými operáciami kopírovania prvkov medzi maticami v celom priebehu výpočtu. Výpočtové časy sú preto podstatne vyššie ako v prípade použitia riedkych matíc a metódy združených gradientov pre výpočet sústavy lineárnych rovníc. Čas výpočtu zahrňuje exaktné určenie odhadu neznámych parametrov metódou LMS. Tabuľka 9.26 a graf 9.19 ukazujú, že so zvyšujúcim sa počtom meraní, resp. počtom použitých kombinácií, je výhodne použiť výpočet s použitím viacerých vlákien. Závislosť je násobná úmerne počtu použitých vlákien. Rozdiely časov medzi jednovláknovým a  $n$ -vláknovým výpočtom sa začnú prejavovať pri výpočte prevyšujúcom približne 100tis kombinácií.

```
1
2 #include <chrono>
3
4 using namespace std::chrono;
5
6 auto start1 = steady_clock::now();
7
8 // neparalelny vypocet
9 LmsExact lms(A, b);
10 lms.get_all_combinations();
11
12 auto end1 = steady_clock::now();
13
14 std::cout << "Median single thread took: "
15           << duration_cast<milliseconds>( end1 - start1 ).count()
16           << " milliseconds\n" ;
17
18 auto start2 = steady_clock::now();
19 // paralelny vypocet popis v~kap. vyssie
20 auto end2 = steady_clock::now();
21
22 std::cout << "Median multi thread took: "
23           << duration_cast<milliseconds>( end2 - start2 ).count()
24           << " milliseconds\n" ;
```

Kód 9.1: Určenie času výpočtu danej úlohy

| Veľkosť sústavy | Počet kombinácií | Čas výpočtu [ms] |          |
|-----------------|------------------|------------------|----------|
|                 |                  | 1 vlákno         | 4 vlákna |
| 15x5            | 3003             | 296              | 127      |
| 17x5            | 6188             | 1334             | 755      |
| 20x7            | 77520            | 18460            | 8574     |
| 21x8            | 203490           | 46197            | 21056    |
| 22x8            | 319770           | 100074           | 49975    |
| 23x9            | 817190           | 239339           | 108506   |
| 25x9            | 2042975          | 1080612          | 488976   |

Tabuľka 9.26: Tabuľka doby výpočtu LMS



Obrázok 9.19: Doba výpočtu v závislosti na počte použitých vlákien a počte kombinácií



## 9.5 Dielčí záver

V tejto kapitole boli zhrnuté výsledky vyrovnania generovaných sústav pomocou M-odhadov, metódy MNČ, LMS, LTS a RANSAC. Výsledky v podstate ukazujú, že každú metódu je výhodne použiť v určitých prípadoch a to v závislosti na veľkosti, počtu observácií, počtu neznámych parametrov, počtu meraní kontaminovaných hrubými chybami a pod.. Z výsledkov je patrné, že metódy M-odhadov sú pomerne nenáročné na čas výpočtu a dávajú spoľahlivý odhad pre sústavy s nízkym počtom kontaminovaných meraní. So zvyšujúcim počtom odľahlých meraní sa presnosť odhadu zhoršuje a pri kontaminácii 50% je odhad prakticky nepoužiteľný. Metódy LMS a LTS majú vzájomne prakticky rovnakú efektívnosť výpočtu až na drobné rozdiely, kde určenie mediánu reziduí je efektívnejšie ako určenie sumy štvorcov reziduí. Oproti M-odhadom je ale náročnosť výpočtu mnohonásobne vyššia. Na druhú stranu je ale možné pomocou týchto metód určiť použiteľný odhad neznámych parametrov aj v prípade vysokého počtu kontaminovaných meraní za cenu určitej straty pôvodnej informácie. Tieto metódy je výhodné použiť na analýzu observácií pred samotným vyrovnaním. Poslednou z porovnávaných metód bola metóda RANSAC. Čas výpočtu metódou RANSAC je o niečo pomalší ako v prípade M-odhadov, ale rýchlejší ako v prípade LMS a LTS. Metóda dokáže určiť presne odhad neznámych parametrov aj v prípade vysokej kontaminácie (do 50%) bez straty pôvodnej informácie.

## 10 Použitie metód v geodetických aplikáciach

V tejto kapitole je popísaný prípad vyrovnania geodetických meraní pomocou sprostredkujúci veličín, použitie štatistických a matematických metód na úlohy v geodézii. Hlavným riešeným problémom je aplikácia metódy LMS na vyrovnanie geodetických sietí a rozbor situácií, ktoré môžu potenciálne nastať.

### 10.1 Vyrovnanie geodetickej siete

Jednou z metód vyrovnania geodetického merania je vyrovnanie sprostredkujúcich veličín. Ako už z názvu vyplýva, hľadané hodnoty určujeme nepriamo pomocou sprostredkujúcich veličín určených priamym meraním v teréne. Sprostredkujúce veličiny sú funkciou súradníc a zároveň funkciou meraných veličín a platí pre nich vzťah

$$R^r \rightarrow R^m : s_i = f_i(L_1, \dots, L_r), i = 1, \dots, m \quad (10.1)$$

$$R^{2n} \rightarrow R^m : s_i = F_i(x_1, \dots, x_{2n}), i = 1, \dots, m \quad (10.2)$$

Kde  $L$  je skutočná hodnota merania,  $x_i$  sú pravouhlé súradnice skutočnej polohy bodov jednotlivých konfigurácií a  $s$  je sprostredkujúca veličina. Oba tieto vzťahy je možné zapísať vo vektorovej podobe

$$\mathbf{f}(\mathbf{L}) = \mathbf{F}(\mathbf{X}) \quad (10.3)$$

V prípade vyrovnania pomocou metódy najmenších štvorcov sa určuje odhad neznámych parametrov vektoru  $\mathbf{X}$  na základe definovanej podmienky. Merané hodnoty sú náhodné veličiny so známou kovariančnou maticou  $\Sigma_{LL}$ . Vzťah medzi odhadom parametrov a meranými veličinami je určený pomocou rovnice tzv. Gauss-Markovho modelu.

$$E\{\mathbf{L}\} = L(\mathbf{X}) \quad (10.4)$$

Kde  $E\{\mathbf{L}\}$  sú očakávané hodnoty vektoru  $\mathbf{L}$  a  $L(\mathbf{X})$  sú súradnice vyjadrené funkciou merania. Vyššie uvedenú rovnicu spravidla nelineárnu je možné linearizovať výberom prvých dvoch členov Taylorového polynómu pri určení vektora  $\mathbf{X}$  ako súčet približnej hodnoty  $\mathbf{X}_0$  s jej prírastkom  $\mathbf{x}$ :

$$E\{\mathbf{L}\} = L(\mathbf{X}_0 + \mathbf{x}) = L(\mathbf{X}_0) + \underbrace{DL(\mathbf{X})|_{\mathbf{X}_0}}_{\mathbf{A}} \mathbf{x} \quad (10.5)$$

a teda

$$\mathbf{Ax} = E\{\mathbf{L}\} - L(\mathbf{X}_0) = E\{\mathbf{L} - L(\mathbf{X}_0)\} \quad (10.6)$$

$$\mathbf{Ax} = E\{\ell\} \quad (10.7)$$

kde  $\mathbf{x} \in R^k$  je vektor neznámych,  $\mathbf{A}$  je prvá matica plánu určená deriváciou funkčných vzťahov podľa neznámych parametrov veľkosti  $\mathbf{A} \in R^{n \times k}$  a  $\ell \in R^n$  značí vektor redukovaných meraní  $\ell = \mathbf{L} - L(\mathbf{X}_0)$ , pričom  $n$  je počet meraní a  $k$  počet neznámych parametrov. Vektor oprav je určený rozdielom odhadov medzi redukovaným meraním a skutočným meraním

$$\epsilon = E\{\ell\} - \ell \quad (10.8)$$

$$\epsilon = \mathbf{Ax} - \ell \quad (10.9)$$

kde  $\epsilon \in R^n$  je vektor skutočných chýb meraní.

Je nutné podotknúť, že pomocou Gauss-Markového modelu nieje možné vypočítať parametre  $x$ , ale iba ich odhad  $\hat{x}$  [Mervart–33]. Nestranný odhad týchto parametrov sa v geodetickej praxi najčastejšie určuje pomocou metódy najmenších štvorcov. Z podmienky metódy najmenších štvorcov požadujeme minimalizáciu štvorcov reziduí.

$$\mathbf{v}^T \mathbf{v} \rightarrow \min \quad (10.10)$$

kde  $\mathbf{v} = \hat{\ell} - \ell = \mathbf{Ax} - \ell$  sú takzvané vyrovnané rezíduá

Geodetické merania sú vo väčšine prípadoch nehomogénne, čo znamená, že sú merané smery a dĺžky s rôznou relatívnou chybou merania. V tom prípade je nutné zaviesť takzvanú váhovú maticu  $\mathbf{P} \in R^{n \times n}$ .

$$\mathbf{P} = \sigma_0^2 \Sigma_{\ell\ell}^{-1} \quad (10.11)$$

kde  $\sigma_0^2$  je apriórna smerodajná odchýlka a  $\Sigma_{\ell\ell}^{-1}$  kovariančná matica meraných observácií. Podmienka metódy najmenších štvorcov pre homogenizované merania je preto určená ako

$$\mathbf{v}^T \mathbf{P} \mathbf{v} \rightarrow \min \quad (10.12)$$

dosadením z rovnice 10.64

$$\mathbf{v}^T \mathbf{P} \mathbf{v} = (\hat{\mathbf{x}} \mathbf{A}^T - \ell^T) \mathbf{P} (\mathbf{A} \hat{\mathbf{x}} - \ell) = \mathbf{A}^T \mathbf{P} \mathbf{A} \hat{\mathbf{x}}^2 - \mathbf{A}^T \mathbf{P} \ell \hat{\mathbf{x}} - \ell^T \mathbf{P} \mathbf{A} \hat{\mathbf{x}} + \ell^T \mathbf{P} \ell \quad (10.13)$$

Extrém získame tak, že prvú deriváciu položíme rovno 0

$$\frac{\partial \mathbf{v}^T \mathbf{P} \mathbf{v}}{\partial \hat{\mathbf{x}}} = 2\mathbf{A}^T \mathbf{P} \mathbf{A} \hat{\mathbf{x}} - 2\mathbf{A}^T \mathbf{P} \ell = 0 \quad (10.14)$$

$$\mathbf{A}^T \mathbf{P} \mathbf{A} \hat{\mathbf{x}} = \mathbf{A}^T \mathbf{P} \ell \quad (10.15)$$

kde  $\mathbf{A}^T \mathbf{P} \mathbf{A} = \mathbf{N}$  a  $\mathbf{A}^T \mathbf{P} \ell = \mathbf{n}$ . Výsledné neznáme získame riešením takzvaných *normálnych rovníc*.

$$\mathbf{N} \hat{\mathbf{x}} = \mathbf{n} \quad (10.16)$$

Pričom matica  $\mathbf{N}$  je

- symetrická
- pozitívne semi-definitná
- často s riedkou štruktúrou

Pre odhad presnosti vyrovnaných neznámych veličín je možné vypočítať kovariančnú maticu odhadovaných parametrov, ktorá má na diagonále kvadráty smerodajných odchýlok hľadaných parametrov a mimo diagonálu prvky kovariancií, ktoré vyjadrujú mieru závislosti medzi odhadovanými parametrami. Kovariančná matica je daná nasledujúcim výrazom [Cepek-34]

$$\Sigma_{\hat{\mathbf{x}}\hat{\mathbf{x}}} = \sigma_0^2 \mathbf{N}^{-1} \quad (10.17)$$

a kovariančná matica vyrovnaných observácií

$$\Sigma_{\hat{\ell}\hat{\ell}} = \mathbf{A} \Sigma_{\hat{\mathbf{x}}\hat{\mathbf{x}}} \mathbf{A}^T \quad (10.18)$$

V prípade, že matica  $\mathbf{A}$  je singularárna to znamená, že matica  $\mathbf{N}$  je pozitívne semi-definitná, je potrebné sústavu nejakým spôsobom regularizovať. Singularita sa spravidla nachádza u voľných sietí. Musíme preto dbať na to, aby sa pri vyrovnaní obmedzila deformácia siete. Tento problém sa najčastejšie rieši zavedením pseudo-meraní a je veľmi podobný vyrovnaní siete s podmienkami. Explicitne definujeme systém s podmienkami, ktorý regularizuje pôvodný systém ako

$$\mathbf{C} \mathbf{x} = \mathbf{c} \quad (10.19)$$

a normálne rovnice sú

$$\begin{pmatrix} \mathbf{N} & \mathbf{C}^T \\ \mathbf{C} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}} \\ \boldsymbol{\Lambda} \end{pmatrix} = \begin{pmatrix} \ell \\ \mathbf{c} \end{pmatrix} \quad (10.20)$$

kde  $\Lambda$  je vektor Lagrangeových multiplikátorov. Dodatočnou podmienkou z matematického hľadiska viazaného extrému je minimalizácia euklidovskej normy podmnožiny neznámych parametrov.

$$\sqrt{\sum x_i^2} \rightarrow \min, \quad i \in \Theta \quad (10.21)$$

kde  $\Theta$  je množina vybraných neznámych parametrov, ale v mnohých prípadoch sa do podmienky použije len potrebná podmnožina bodov. Táto podmienka je ekvivalentná s tradičnou helmertovou transformáciou [Čepek-24].

### 10.1.1 Vyrovnanie s podmienkami

Pokiaľ sú neznáme parametre viazané na nejakú podmienku hovoríme o vyrovnaní s podmienkami. Podmienky sa obecné definujú ako

$$\mathbf{g}(\mathbf{x}) = \mathbf{0} \quad (10.22)$$

Analogicky s rov. 10.23 linearizujeme podmienky

$$\mathbf{0} = g\{\mathbf{X}\} = \mathbf{g}(\mathbf{X}_0 + \mathbf{x}) = \underbrace{\mathbf{g}(\mathbf{X}_0)}_{-\mathbf{h}} + \underbrace{D\mathbf{g}(\mathbf{X})|_{\mathbf{x}_0}}_{\mathbf{H}} \mathbf{x} \quad (10.23)$$

$$\mathbf{H}\mathbf{x} = \mathbf{h} \quad (10.24)$$

Pre odhad parametru  $\tilde{\mathbf{x}}$  hľadáme minimum z

$$\tilde{\mathbf{v}}^T \mathbf{P} \tilde{\mathbf{v}} \rightarrow \min \quad \tilde{\mathbf{v}} = \mathbf{A} \tilde{\mathbf{x}} - \ell \quad (10.25)$$

viazaného na podmienku  $\mathbf{H}\tilde{\mathbf{x}} = \mathbf{h}$ . Na vyhľadanie viazaného extrému je možné použiť metódu Lagrangeových multiplikátorov.

$$\tilde{\mathbf{v}}^T \mathbf{P} \tilde{\mathbf{v}} + 2\mathbf{k}^T (\mathbf{H}\tilde{\mathbf{x}} - \mathbf{h}) = \Omega(\tilde{\mathbf{x}}, \mathbf{k}) \rightarrow \min \quad (10.26)$$

Kde  $\mathbf{k}$  je vektor Lagrangeových multiplikátorov. Parciálne derivácie podľa oboch neznámych sú rovné 0.

$$\frac{\partial \Omega}{\partial \tilde{\mathbf{x}}} = 2\mathbf{A}^T \mathbf{P} \mathbf{A} \tilde{\mathbf{x}} - 2\mathbf{A}^T \mathbf{P} \ell + 2\mathbf{H}^T \mathbf{k} = 0 \quad (10.27)$$

$$\frac{\partial \Omega}{\partial \mathbf{k}} = 2(\mathbf{H}\tilde{\mathbf{x}} - \mathbf{h}) = 0 \quad (10.28)$$

derivácie vedu k

$$\begin{pmatrix} \mathbf{A}^T \mathbf{P} \mathbf{A} & \mathbf{H}^T \\ \mathbf{H} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}} \\ \mathbf{k} \end{pmatrix} = \begin{pmatrix} \mathbf{A}^T \mathbf{P} \ell \\ \mathbf{h} \end{pmatrix} \quad (10.29)$$

vyjadríme  $\tilde{\mathbf{x}}$

$$\tilde{\mathbf{x}} = (\mathbf{A}^T \mathbf{P} \mathbf{A})^{-1} (\mathbf{A}^T \mathbf{P} \ell - \mathbf{H}^T \mathbf{k}) = \hat{\mathbf{x}} - (\mathbf{A}^T \mathbf{P} \mathbf{A})^{-1} \mathbf{H}^T \mathbf{k} \quad (10.30)$$

kde

$$\mathbf{k} = (\mathbf{H} (\mathbf{A}^T \mathbf{P} \mathbf{A})^{-1} \mathbf{H}^T)^{-1} (\mathbf{H} \hat{\mathbf{x}} - \mathbf{h}) \quad (10.31)$$

takže

$$\tilde{\mathbf{x}} = \hat{\mathbf{x}} - (\mathbf{A}^T \mathbf{P} \mathbf{A})^{-1} \mathbf{H}^T (\mathbf{H} (\mathbf{A}^T \mathbf{P} \mathbf{A})^{-1} \mathbf{H}^T)^{-1} (\mathbf{H} \hat{\mathbf{x}} - \mathbf{h}) \quad (10.32)$$

### Regularizácia modelu zavedením podmienok do vyrovnania

V tomto prípade predpokladáme, že defekt  $def$  matice  $\mathbf{A}$  je  $def > 1$ . Tento prípad sa vyskytuje celkom často v reálnych aplikáciach. Obvykle sa pre regularizáciu normálnych rovníc používajú podmienky  $\mathbf{H}\tilde{\mathbf{x}} = \mathbf{h}$ . Pokiaľ zvolíme nutný počet podmienok t.j.  $p = 3$  pre rovinné siete, hovoríme o vyrovnaniu voľnej siete. Pokiaľ zvolíme v sieti o  $n$  bodoch väčší počet podmienok  $p$ ,  $3 < p < 2n - 1$ , jedná sa o vyrovnanie siete viazanej. Matica  $\tilde{\mathbf{A}}$  novej sústavy je

$$\tilde{\mathbf{A}} = \begin{pmatrix} \mathbf{A} \\ \mathbf{H} \end{pmatrix} \quad (10.33)$$

Matica  $\tilde{\mathbf{A}}$  novej sústavy má defekt rovný 0 a teda je regulárna. Priamy výpočet parametrov  $\tilde{\mathbf{x}}$  a  $\mathbf{k}$  je ukázaný v rov.10.31 a rov.10.32

$$\begin{pmatrix} \tilde{\mathbf{x}} \\ \mathbf{k} \end{pmatrix} = \begin{pmatrix} \mathbf{A}^T \mathbf{P} \mathbf{A} & \mathbf{H}^T \\ \mathbf{H} & \mathbf{0} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{A}^T \mathbf{P} \ell \\ \mathbf{h} \end{pmatrix} \quad (10.34)$$

Inverziu overíme

$$\begin{pmatrix} \mathbf{A}^T \mathbf{P} \mathbf{A} & \mathbf{H}^T \\ \mathbf{H} & \mathbf{0} \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{U} & \mathbf{V}^T \\ \mathbf{V} & \mathbf{W} \end{pmatrix} \quad (10.35)$$

$$\begin{pmatrix} \mathbf{U} & \mathbf{V}^T \\ \mathbf{V} & \mathbf{W} \end{pmatrix} \begin{pmatrix} \mathbf{A}^T \mathbf{P} \mathbf{A} & \mathbf{H}^T \\ \mathbf{H} & \mathbf{0} \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{U} \mathbf{A}^T \mathbf{P} \mathbf{A} + \mathbf{V}^T \mathbf{H} & \mathbf{U} \mathbf{H}^T \\ \mathbf{V} \mathbf{A}^T \mathbf{P} \mathbf{A} + \mathbf{W} \mathbf{H} & \mathbf{V} \mathbf{H}^T \end{pmatrix} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \quad (10.36)$$

$$\tilde{\mathbf{x}} = \mathbf{U} \mathbf{A}^T \mathbf{P} \ell + \mathbf{V}^T \mathbf{h} \quad (10.37)$$

pretože  $\mathbf{V}$  a  $\mathbf{h}$  sú konštantné môžeme písať

$$\Sigma_{\tilde{x}\tilde{x}} = \mathbf{U}\mathbf{A}^T\mathbf{P}\Sigma_{\ell\ell}\mathbf{P}\mathbf{A}\mathbf{U} = \sigma_0\mathbf{U}\mathbf{A}^T\mathbf{P}\mathbf{A}\mathbf{U} \quad (10.38)$$

Keďže  $\mathbf{U}\mathbf{A}^T\mathbf{P}\mathbf{A} = \mathbf{I} - \mathbf{V}^T\mathbf{A}$ , potom

$$\Sigma_{\tilde{x}\tilde{x}} = \sigma_0(\mathbf{U} - \mathbf{V}^T\mathbf{H}\mathbf{U}) \quad (10.39)$$

a  $\mathbf{H}\mathbf{U} = 0$  teda

$$\Sigma_{\tilde{x}\tilde{x}} = \sigma_0\mathbf{U} \quad (10.40)$$

### Voľba podmienok pre regularizáciu rovinnej siete

Majme teda 3 množiny funkčných vzťahov  $\mathbf{f}(\mathbf{L}), \mathbf{F}(\mathbf{X})$  a  $\mathbf{g}(\mathbf{X})$ , pre ktoré platí.

$$\mathbf{f}(\mathbf{L}) = \mathbf{F}(\mathbf{X}) \quad \text{s podmienkou} \quad \mathbf{g}(\mathbf{X}) = 0 \quad (10.41)$$

Je vhodné podotknúť, že v prípade rovinných geodetických sietí si vystačíme z troch vzťahmi.

- $\mathbf{f}(\mathbf{L})$

$$\omega_{ijk} = \psi_{ik} - \psi_{ij} \quad (10.42)$$

$$\omega_{ijk} = \omega_{ikj} \quad (10.43)$$

$$d_{ij} = d_{ji} \quad (10.44)$$

- $\mathbf{F}(\mathbf{X})$

$$\omega_{ijk} = \arccos \frac{\Delta x_{ij}\Delta x_{ik} + \Delta y_{ij}\Delta y_{ik}}{\sqrt{\Delta x_{ij}^2\Delta y_{ij}^2}\sqrt{\Delta x_{ik}^2\Delta y_{ik}^2}} \quad (10.45)$$

$$\omega_{ij} = \arccos \frac{\Delta x_{ij}\Delta x_{ik} + \Delta y_{ij}\Delta y_{ik}}{\sqrt{\Delta x_{ij}^2\Delta y_{ij}^2}\sqrt{\Delta x_{ik}^2\Delta y_{ik}^2}} \quad (10.46)$$

$$d_{ij} = \sqrt{\Delta x_{ij}^2\Delta y_{ij}^2} \quad (10.47)$$

kde  $\Delta x_{ij}$  značí súradnicové rozdiely v x-ovej súradnici i-tého a j-tého bodu. Ostáva voľba podmienok  $\mathbf{g}(\mathbf{X})$  tak, aby bolo možné sieť umiestniť do priestoru. Podmienky sa nutne volia tak, aby neovplyvňovali hodnoty oprav meraných uhlov a dĺžok  $\tilde{\mathbf{v}}^T \mathbf{P} \tilde{\mathbf{v}} \rightarrow \min$ . Tomuto požiadavku vyhovujú funkčné vzťahy  $S(x)$  odvodené z lineárnej zhodnostnej transformácie.

$$x_i = a + \cos(\varphi)x_i^0 - \sin(\varphi)y_i^0 \quad (10.48)$$

$$y_i = b + \sin(\varphi)x_i^0 - \cos(\varphi)y_i^0 \quad (10.49)$$

kde  $x_i^0$  je približná súradnica i-tého bodu a  $x_i$  je výsledná súradnica po transformácii. Týmto vzťahom po linearizácii zodpovedá matica  $S$ .

$$\mathbf{H} = \mathbf{S}^T = \begin{pmatrix} 1 & 0 & \cdots & 1 & 0 \\ 0 & 1 & \cdots & 0 & 1 \\ -y_1^0 & x_1^0 & \cdots & -y_n^0 & x_n^0 \end{pmatrix} \quad (10.50)$$

Tieto podmienky, ktoré zodpovedajú modifikovanej Helmertovej transformácii, môžeme aplikovať na vybraný počet bodov siete alebo na všetky body. Výsledkom je stotožnenie ťažiska súradníc oboch sietí približných  $x_i^0, y_i^0$  a vyrovnaných  $x_i = x_i^0 + dx_i, y_i = y_i^0 + dy_i$  a stotožnenie bodov dané podmienkou  $\sqrt{\sum x_i^2} \rightarrow \min$ . Pokiaľ v matici  $S^T$  necháme iba súradnice dvoch ľubovoľných prvkov siete jedná sa o variantu jedného bodu a smerníku.

$$\mathbf{H} = \mathbf{S}^T = \begin{pmatrix} 1 & 0 & \cdots & \cdots & 1 & 0 \\ 0 & 1 & \cdots & \cdots & 0 & 1 \\ 0 & 0 & \cdots & -y_j^0 & x_j^0 & 0 \end{pmatrix} \quad (10.51)$$

### Vyrovnanie sprostredkujúcich veličín pomocou GSO

Zovšeobecnenú ortogonalizáciu môžeme s výhodou použiť na riešenie vyrovnania sprostredkujúcich veličín, pričom touto transformáciou získame priamo vektor neznámych  $\mathbf{x}$  a vektor reziduí  $\mathbf{v}$ . Teda z rovnice  $\mathbf{v} = \mathbf{A}\mathbf{x} - \mathbf{l}$  zostrojíme blokovú maticu a použitím vzťahov z rovnice 7.7 získame výslednú transformáciu

$$\begin{pmatrix} \mathbf{A} & \mathbf{l} \\ \mathbf{I} & \mathbf{0} \end{pmatrix} \rightarrow \begin{pmatrix} \mathbf{Q} & \mathbf{v} \\ \mathbf{R}^{-1} & \mathbf{x} \end{pmatrix} \quad (10.52)$$

A  $\mathbf{R}^{-1}$  je inverziou k hornej trojuholníkovej matici normálnych rovníc

$$\mathbf{A}^T \mathbf{P} \mathbf{A} = \mathbf{N} = \mathbf{R}^T \mathbf{R} \quad (10.53)$$



V prípadoch, keď sú sprostredkujúce veličiny merané uhly a dĺžky s rôznou presnosťou, je nutné sústavu  $\mathbf{v} = \mathbf{A}\mathbf{x} - \mathbf{l}$  homogenizovať pomocou váhovej matice. Homogenizovaná sústava je v tvare  $\dot{\mathbf{v}} = \dot{\mathbf{A}}\mathbf{x} - \dot{\mathbf{l}}$ , pričom  $\dot{\mathbf{v}} = \mathbf{P}^{\frac{1}{2}}\mathbf{v}$ ,  $\dot{\mathbf{A}} = \mathbf{P}^{\frac{1}{2}}\mathbf{A}$  a  $\dot{\mathbf{l}} = \mathbf{P}^{\frac{1}{2}}\mathbf{l}$ , kde  $\mathbf{P} = \text{diag}[p_i]$ . Postup riešenia homogenizovanej sústavy zovšeobecneným ortogonalizačným procesom je zhodný s 10.52. Skalárnym súčinom  $\mathbf{R}^{-1}$  a  $\mathbf{Q}$  dostaneme príslušné matice váhových koeficientov.

Kovariančná matica meraní

$$\Sigma_{LL} = \mathbf{Q}\mathbf{Q}^T \quad (10.54)$$

resp.

$$\Sigma_{LL} = \mathbf{P}^{-\frac{1}{2}}\mathbf{Q}\mathbf{P}^{-\frac{1}{2}}\mathbf{Q}^T \quad (10.55)$$

Kovariančná matica vyrovnaných neznámych

$$\Sigma_{xx} = \mathbf{R}^{-1}\mathbf{R}^{-T} \quad (10.56)$$

[Cepek-34]

## 10.2 Zostavenie normálnych rovníc

Zostavenie normálnych rovníc môže byť pri väčšom množstve dát výpočtovo náročná úloha. Zamerajme sa teda na čo najvyššiu efektívnosť zostavenia týchto rovníc. Predpokladáme, že matica plánu je obecná riedka matica. Pri homogenizácii meraní je váhová matica vo väčšine prípadoch diagonálna. Váhovú maticu je preto možné uložiť ako vektor. V závislosti na predošlých predpokladoch môžeme očakávať, že výsledkom operácie  $\mathbf{A}^T\mathbf{P}\mathbf{A}$  je symetrická pozitívne semi-definitná matica s riedkou štruktúrou. Násobenie je možné s výhodou vykonávať po riadkoch matice.

$$\mathbf{A} = \begin{pmatrix} a_{11} & 0 & 0 & a_{14} \\ 0 & 0 & a_{23} & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ 0 & 0 & 0 & a_{44} \end{pmatrix} \quad (10.57)$$

$$\mathbf{A}^T\mathbf{P}\mathbf{A} = \sum_{i=1}^n \mathbf{a}_i^T p_{ii} \mathbf{a}_i \quad (10.58)$$

$$\mathbf{A}^T\mathbf{P}\mathbf{A} = \begin{pmatrix} a_{11} \\ 0 \\ 0 \\ a_{14} \end{pmatrix} p_{11} \begin{pmatrix} a_{11} & 0 & 0 & a_{14} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ a_{23} \\ 0 \end{pmatrix} p_{22} \begin{pmatrix} 0 & 0 & a_{23} & 0 \end{pmatrix} + \dots \quad (10.59)$$

Riedku maticu vkladáme znovu do riedkej štruktúry. Z následujúceho vzťahu je patrné, že nieje potrebné vytvárať transpozíciu matice  $\mathbf{A}$ . V prípade zostavenia vektoru  $\mathbf{n}$  je situácia nepatrne jednoduchšia. V prvom kroku je výhodne vynásobiť váhovou maticu s vektorom redukovaných meraní a následne výsledný vektor vynásobiť maticou plánu.

### 10.3 Riedke matice v geodézii

V geodézii má prvá matica plánu v mnohých prípadoch riedku štruktúru. Riadky matice plánu reprezentujú jednotlivé merania a stĺpce tvoria súradnice observovaných bodov. Každá meraná veličina je v matici plánu vyjadrená pomocou sprostredkujúcej veličiny. Sprostredkujúca veličina je meraná veličina vyjadrená v závislosti na hľadaných parametroch. Prvky matice tvoria diferenciály sprostredkujúcich veličín podľa súradníc. Nenulová hodnota v matici plánu sa nachádza na pozíciách, kde je meraná hodnota závislá na súradniciach. Výpočet sústav je možné v mnohých prípadoch geodetických uloh značne urýchliť použitím algoritmov optimalizovaných pre štruktúru riedkej matice. V prípade, že je matica dostatočne riedka, prevod na štruktúru riedkej matice môže mať omnoho menšie pamäťové a výpočtové nároky, akoby bola matica ukladaná v pôvodnej podobe. Dostatočnú riedkosť matice nieje možné úplne jasne určiť nejakou pevnou hranicou.

Pre iteratívne riešenie metódy najmenších mediánov štvorcov reziduí je veľmi dôležitá nielen presnosť, ale aj rýchlosť výpočtu, pretože sa sústava rieši opakovane s postupným vylučovaním odľahlých hodnôt z matice plánu.

### 10.4 Riedke matice pri určovaní presnej polohy bodu pomocou GPS

Metóda určovania presnej polohy bodu (precise point positioning PPP) je nová, progresívne sa vyvíjajúca metóda na určovanie polohy bodu s relatívne vysokou presnosťou porovnateľnou s metódou dvojitéch diferencií. Tento vývoj je podmienený zvyšujúcou sa presnosťou polohy družice a chodu jej hodín poskytovanou agentúrou IGS (International GPS Service). Metódou dvojitéch diferencií dokážeme väčšinu systematických chýb merania eliminovať súčasným pozorovaním na dvoch relatívne blízkych stanicích pričom stanica, ktorá má známu polohu posiela korekcie stanicím s neznámu polohou. V prípade použitia metódy PPP, kde sa na určenie polohy bodu používa iba jedna stanica, je postup odstránenia systematických chýb odlišný. Vplyv chyby jonosféry sa odstráni ionosfer-free kombináciou signálu prijímaného na dvoch frekvenciách[Hofmann-44]. Troposférické meškanie signálu sa modeluje

funkciou  $zpd$  v závislosti na veľkosti zenitového uhla a mapovaciou funkciou. Observácie sa rozdeľujú na kódové a fázové. Obe typy observácií sa spracujú metódou najmenších štvorcov spoločne, avšak s rozdielnou váhou, pretože kódové meranie omnoho nepresnejšie ako fázové. Výsledkom vyrovnania sú súradnice stanice, chyba hodín stanice, chyba troposféry a vyrovnané ambiguity.

### 10.4.1 Rovnice pozorovaní

Ako už bolo spomenuté v GPS sa rozdeľuje pozorovanie na kódové a fázové merania. Obe z týchto pozorovaní popisujú mierne odlišné rovnice

$$\ell_p = \rho + C(dt + dT) + Tr + \epsilon_p \quad (10.60)$$

$$\ell_\varphi = \rho + C(dt + dT) + Tr + N\lambda + \epsilon_\varphi \quad (10.61)$$

kde

- $\ell_p, \ell_\varphi$  - ionosfer-free kombinácie kódového resp. fázového merania
- $dt$  - chyba hodín stanice
- $dT$  - chyba hodín družice
- $C$  - rýchlosť svetla vo vákuu
- $Tr$  - vplyv troposféry
- $\lambda$  - vlnová dĺžka fázovej observácie
- $N$  - počet ambiguity
- $\epsilon_p, \epsilon_\varphi$  - zbytkový šum
- $\rho$  je geometrická vzdialenosť spočítaná zo súradníc družice  $X_s, Y_s, Z_s$  a stanice  $x, y, z$

$$\rho = \sqrt{(X_s - x)^2 + (Y_s - y)^2 + (Z_s - z)^2} \quad (10.62)$$

Troposféra sa v zjednodušenom prípade vyjadruje funkciou  $zpd$  popisujúcou zmeškanie signálu v závislosti na zenitovom uhle a mapovacou funkciou  $M$ . Metóda PPP je založená na základe znalosti hodín družice, preto je  $dT$  z rovnice možno vylúčiť [Kuba-43]. Z toho získame zjednodušený matematický model

$$f_p = \rho + Cdt + Mzpd + \epsilon_p - \ell_p = 0 \quad (10.63)$$

$$f_\varphi = \rho + Cdt + Mzpd + N\lambda + \epsilon_\varphi - \ell_\varphi = 0 \quad (10.64)$$

### 10.4.2 Sekvenčné vyrovnanie

Sekvenčné vyrovnanie umožňuje vyrovnávať meranie po menších častiach. Tento spôsob vyrovňania je časovo a pamäťovo menej náročný, pretože sa matice vstupujúce do vyrovňania rozdelia na menšie bloky, ktoré sa vyrovnávajú zvlášť. Špeciálnym prípadom sekvenčného vyrovňania je Kalmanov filter, ktorý sa prednostne využíva na odhad presnosti vyrovnaných veličín GPS meraní. Nesprávny počiatkový odhad parametrov sa následne vyhladzuje metódou "backward smoothing". Predpokladajme, že matice  $\mathbf{A}$ ,  $\mathbf{P}$  a vektor redukovaných meraní  $\ell$  rozdelíme na  $n$  blokov rovnakej veľkosti. Potom platí, že

$$\left( \sum_{i=1}^n \mathbf{A}_i^T \mathbf{P}_i \mathbf{A}_i \right) \mathbf{x} = \left( \sum_{i=1}^n \mathbf{A}_i^T \mathbf{P}_i \ell_i \right) \quad (10.65)$$

kde

$$\mathbf{A} = \begin{pmatrix} \frac{\partial f(X, \ell_p)}{\partial X_i} & \cdots & \frac{\partial f(X, \ell_p)}{\partial X_n} \\ \vdots & & \vdots \\ \frac{\partial f(X, \ell_\varphi)}{\partial X_i} & \cdots & \frac{\partial f(X, \ell_\varphi)}{\partial X_n} \\ \vdots & & \vdots \end{pmatrix} \quad \mathbf{x} = \begin{pmatrix} x \\ y \\ z \\ zpd \\ dt \\ N_j \end{pmatrix} \quad (10.66)$$

### 10.4.3 Analýza matice plánu GPS meraní

Zjednodušíme maticu plánu GPS meraní  $\mathbf{A}$  a rozdelíme ju na bloky  $\mathbf{A}_1 \dots \mathbf{A}_6$ . Matica plánu obsahuje kódové meranie v rozmedzí 1 až  $\frac{n}{2}$  riadkov a fázové meranie  $\frac{n}{2} + 1$  až  $n$ . Na obrázku sú bloky meraní farebne odlíšené.

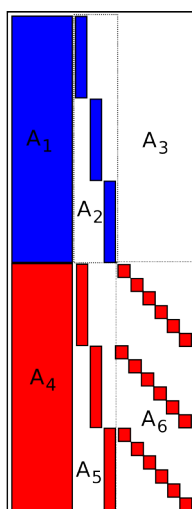
- Blok  $\mathbf{A}_1$  má rozmery  $kn \times 4$  a zahŕňa parciálne derivácie podľa súradníc a troposféry kódového merania. Blok neobsahuje nenulové prvky v prípade, že je hodnota troposféry platná pre celý deň. Obecne sa troposféra akvizuje v menších časových intervaloch.
- Blok  $\mathbf{A}_2$  má rozmery  $kn \times pe$  a zahŕňa parciálne derivácie podľa hodín stanice kódového merania. Pre každú epochu je zaplnený jeden stĺpec matice.
- Blok  $\mathbf{A}_3$  má rozmery  $kn \times pd$  a zahŕňa parciálne derivácie podľa ambiguit. Kódové observácie ambiguit neobsahujú, to znamená, že  $\mathbf{A}_3 = 0$ .

- Blok  $\mathbf{A}_4$  má rozmery  $pn \times 4$  a zahŕňa parciálne derivácie podľa súradníc a tropoféry fázového merania. Platia tam rovnaké podmienky ako v prípade kódového merania.
- Blok  $\mathbf{A}_5 = \mathbf{A}_2$ .
- Blok  $\mathbf{A}_6$  a  $kn \times pd$  zahŕňa parciálne derivácie podľa ambiguit. Každý riadok obsahuje jedinú hodnotu rovnú jednej a ostatné prvky riadku sú nulové.

kde

- $kn$  - počet kódových meraní
- $pn$  - počet fázových meraní, obvykle  $kn = pn$
- $pe$  - počet epoch
- $pd$  - počet všetkých družíc, na ktoré bolo observované v rámci vyrovnania.

Z obrázku je teda poznať, že sa budú vyrovnávať tri epochy kódového merania, tri epochy fázového merania a v každej epoche bolo merané na 6 družíc. Pri každom vyrovnaní sa štruktúra matíc mení len nepatrne. Pridávaním epoch sa bude zväčšovať počet stĺpcov v blokoch  $\mathbf{A}_2$  a  $\mathbf{A}_5$ . Z obrázku je patrné, že pre  $n$  epoch budú tieto submatice obsahovať približne  $\frac{1}{n}$  nenulových prvkov z celkového počtu prvkov patriacich do submatíc  $\mathbf{A}_2$  a  $\mathbf{A}_5$ . Z toho vyplýva, že riedkosť matice stúpa s pridávaním epoch. Pre matice obsahujúce viac ako tisíc epoch sa už môže stať výhodným použitie algoritmov a úložných štruktúr riedkych matíc.



Obrázok 10.1: Matica plánu

#### 10.4.4 Choleského rozklad a minimum degree permutácia

Ostáva ešte ukázať rozloženie  $\mathbf{A}$  pomocou Choleského rozkladu. Pre  $\mathbf{A}$  existuje taká dolná trojuholníková matica  $\mathbf{L}$ , že

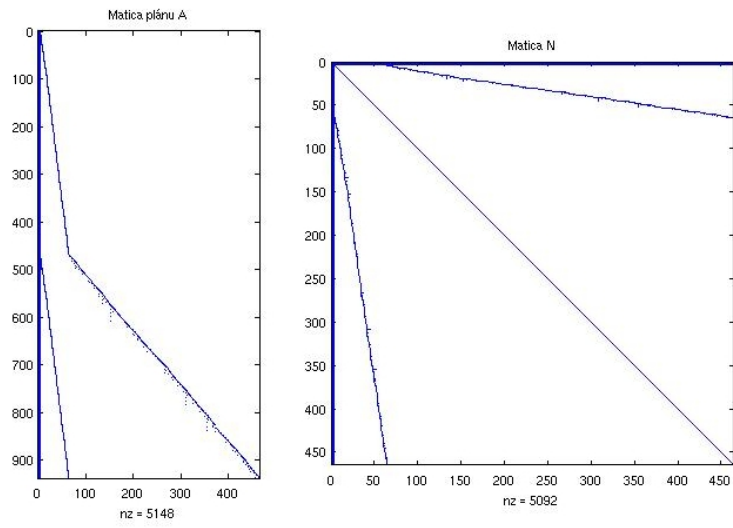
$$\mathbf{M}^T \mathbf{A} \mathbf{M} = \mathbf{L} \mathbf{L}^T \quad (10.67)$$

kde matica  $\mathbf{M}$  je permutačná matica. Permutačná matica má zamedziť vzniku nových nenulových prvkov. Jedným z mnohých algoritmov na vyhotovenie permutačnej matice je algoritmus minimum degree. Algoritmus vyberie z matice riadky, ktoré majú najmenší počet nenulových prvkov a vzostupne ich zoradí. Pre určenie počtu nenulových prvkov riadku sa môže s výhodou použiť rozdiel indexov ukazujúcich na začiatok a koniec riadku obrázkov 13.18. Nové nenulové prvky môžu vzniknúť na riadkoch, na ktorých bola aplikovaná lineárna kombinácia. Rozklad vedie k riešeniu dvoch systémov s trojuholníkovými maticami

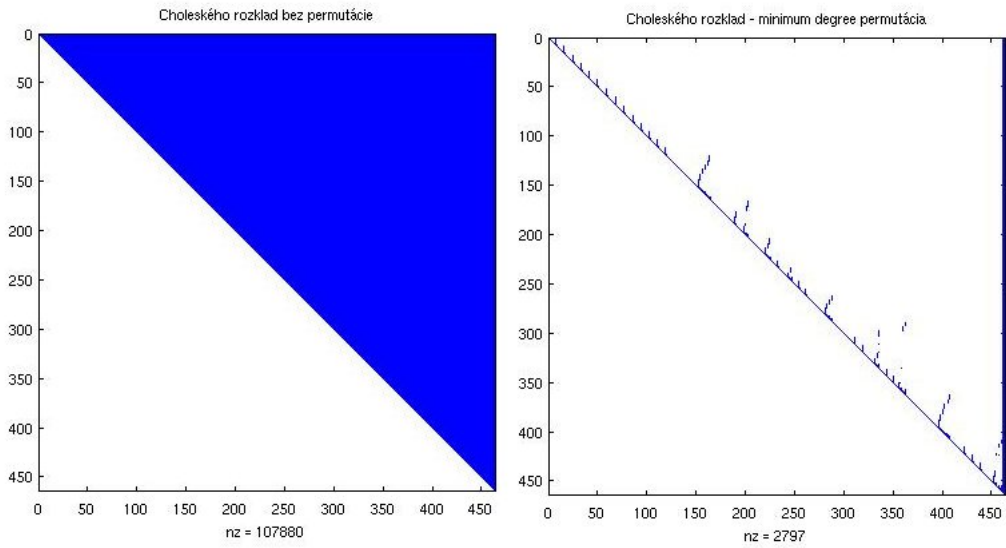
$$\mathbf{L} \mathbf{y} = \mathbf{b}, \quad \mathbf{L}^T \mathbf{x} = \mathbf{y} \quad (10.68)$$

#### 10.4.5 Výsledky observácii stanice NICO

Na ukázanie riedkosti matíc boli použité dáta stanice NICO (Nicosia, Cyprus) siete IGS. Pre názornejšiu ukážku sa k pôvodným dátam softwaerovo vygenerovali fázové skoky. Na obrázkoch sú znázornené grafy riedkych matíc desiatich minút kombinácie fázových a kódových observácií. Rozloženie nenulových prvkov v matici je dokonca tak výhodne, že po vyhotovení matice  $\mathbf{N}$  sa počet nenulových prvkov značne nezvýši, takže sa zachováva pôvodná riedkosť matice. Rozloženie prvkov matice je platné pre observácie všetkých GPS staníc. Pre následný Choleského rozklad je však počiatočná poloha nenulových prvkov matice  $\mathbf{N}$  nevýhodná a v prípade ponechania tejto konfigurácie je výsledok Choleského rozkladu bez zbytku zaplnený. Tento nedostatok je možné odstrániť permutačnou maticou. Permutačná matica je vyhotovená algoritmom minimum degree. Použitie permutácie značne zníži počet nenulových prvkov rozkladu pre daný časový interval. V našom konkrétnom prípade to je až 50-násobne. Inverziou matice je však v každom prípade matica plná. Nie vždy je však nutné poznať presnosti všetkých prvkov vyrovnania. Riešením je uchovanie riedkeho rozkladu, implicitný výpočet súradníc stanice a ich presnosti a v prípade potreby explicitný výpočet požadovaného parametru, či jeho presnosti.



Obrázok 10.2: Matice pre 10 min pozorovania



Obrázok 10.3: Matice Choleského rozkladu

## 11 Určovanie odľahlých geodetických meraní metódou LMS

Metóda LMS môže byť mimo iné použitá pre odhaľovanie meraní podozrivých z odľahlosti. Pre tento účel bola zvolená tzv. stratégia polovičných vzorov. Stratégia polovičných vzorov využíva vzory sústavy s polovičným množstvom meraných hodnôt. V ideálnom prípade by bolo vhodné vytvoriť všetky kombinácie polovičných vzorov a z nich určiť hodnotu minimálneho mediánu. Polovica meraní, z ktorej by bol určený minimálny medián by bola použitá pre výpočet odhadu neznámych parametrov a druhá polovica dát by bola označená ako podozrivá z odľahlosti. Na základe ďalšieho skúmania by sa potvrdilo alebo vyvrátilo, či tieto potenciálne odľahlé merania do tejto skupiny patria alebo naopak. V praxi sú však bežne predmetom vyrovnania omnoho väčšie objemy meraní. V takýchto prípadoch už úplný počet kombinácií nieje možné vytvoriť. Z tohto dôvodu sa polovičné vzory generujú náhodne. Práve náhodne generovanie polovičných vzorov bude použité pre detekciu meraní podozrivých z odľahlosti. Výhodou tohto riešenia je že dokáže odhaliť až polovicu kontaminovaných observácií.

### 11.1 Postup určenia množiny meraní podozrivých z odľahlosti

Postup určenia množiny meraní podozrivých z odľahlosti uvedený nižšie nemusí vrátiť správny odhad neznámych parametrov pri vysokej kontaminácii observácií hrubými chybami. Postup slúži výhradne na určenie množiny kontaminovaných meraní. Postup je založený na metóde LMS použitím náhodných vzorov. Podozrivé merania sa určujú pomocou vzťahu 3.5 a 3.6. V každom kroku sa postupne zostavuje histogram vylúčených hodnôt. Histogram sa vytvára buď iba na základe veľkosti reziduí alebo taktiež na základe poklesu mediánu pred a po vylúčení konkrétneho merania. Podľa kap. 5.1 sa časť hrubej chyby odrazí v samotnej oprave a časť sa rozdelí medzi ostatné opravy. Preto je možné predpokladať, že po odstránení odľahlého merania výrazne klesne hodnota mediánu. Do histogramu sa teda budú počítať iba podozrivé hodnoty opráv, pre ktoré platí, že po ich vylúčení nastal výraznejší pokles mediánu. Voľba prahu poklesu závisí na tom, akú veľkosť hrubej chyby dokáže LMS rozpoznať.

#### **Vyhotovenie histogramu potenciálne odľahlých hodnôt:**

1. Vygenerovanie náhodného polovičného vzoru  $V$



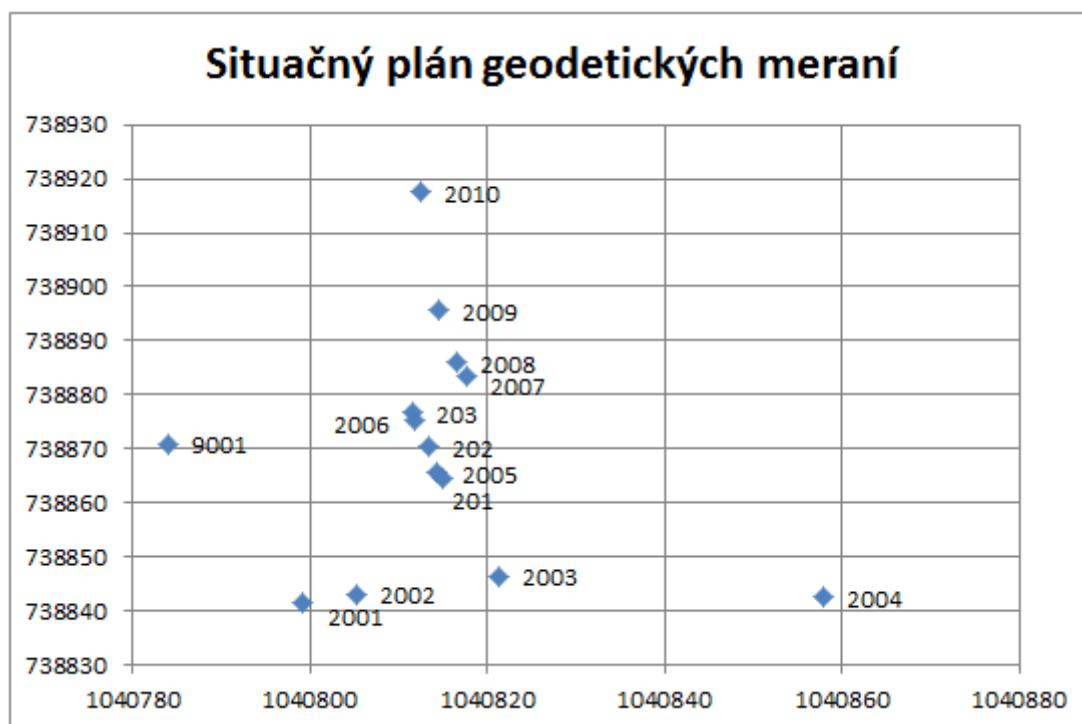
2. Výpočet neznámych parametrov použitím observácií vygenerovaného vzoru štandardnou metódou MNČ
3. Určenie mediánu štvorcov vyrovnaných opráv meraní
4. Výpočet štandardných smerodajných odchýlok robustného vyrovnaní podľa rovnice 3.5
5. Výber (označenie) vyrovnaných opráv presahujúcich 2,5 násobok štandardných smerodajných odchýlok podľa rovnice 3.6
6. Výmena opráv presahujúcich 2,5 násobok štandardných smerodajných odchýlok za merania, ktoré neboli pôvodne zahrnuté do náhodne generovaného vzoru. Táto operácia môže skončiť úspechom či naopak.
  - (a) Množina označených opráv je prázdna. Operácia sa nevykonala a výsledkom je hodnota *false*
  - (b) Množina označených opráv je neprázdna, ale množina meraní nezahrnutých do vzoru ma nedostatočný počet meraní pre uskutočnenie výmeny. Výsledkom operácie je hodnota *false*
  - (c) Výmena prebehla. Výsledkom je hodnota *true*
7. Analýza poklesu alebo nárastu mediánu výmenou označených opráv . V prípade, že pokles mediánu dosiahol predom definovaný prah tak sa k indexu meranej hodnoty inkrementuje hodnota, ktorá udáva počet vylúčení konkrétneho merania.
8. Test minimálneho mediánu. Pri poklese mediánu sa uložia merania vybraného vzoru a vyrovnané hodnoty do pripravených dátových štruktúr. V prípade, že sa medián v priebehu ďalšieho výpočtu nezmení, tak je toto vyrovnanie považované za konečné.
9. Pokiaľ výmena z kroku 6. prebehla v poriadku, teda výsledkom operácie je hodnota *true*, pokračuj na bod 2. V opačnom prípade za splnenia podmienky, že nebol dosiahnutý maximálny počet náhodne generovaných vzorov, pokračuj na bod 1.

Na základe histogramu hodnôt podozrivých z odľahlosti môžu byť z pôvodnej množiny meraní tieto hodnoty vylúčené až do veľkosti 50%. Pokiaľ bude histogram obsahovať viac než 50% meraní, tak môžu teoreticky nastať dve situácie. Buď v meraní neboli detekované žiadne hrubé chyby, alebo na druhu stranu je hrubých chýb v meraní viac než 50%. Za situácie, keď histogram obsahuje menšiu polovicu dát,

tak sa z tej *lepšej* polovice určí odhad neznámych parametrov. Takto určený odhad môžeme považovať za odhad parametrov určených metódou LMS.

## 11.2 Aplikovanie postupu na terestrických geodetických meraniach

Postup stanovenia množiny odľahlých meraní metódou LMS bol overený na niekoľkých príkladoch geodetických meraní vyhotovených v praxi. Zápisník merania voľného stanoviska a súradnice bodov, na základe ktorého sú v tejto kapitole vyhotovené histogramy a ostatné výsledky, sú uvedené v prílohe A. Pre zjednodušenie uvažujeme v tomto prípade homogénne merania. To znamená predpoklad, že sú smery aj dĺžky merané s rovnakou presnosťou. Hrubé chyby boli rozdelené medzi  $n$  definovaných meraní. Spôsob rozdelenia hrubých chýb je uvedený v tabuľke 11.1



Obrázok 11.1: Súradnice bodov geodetických meraní

### Postup vyrovnaní geodetických meraní:

- Prevod šikmých dĺžok na vodorovné.
- Výpočet približných súradníc vyrovnávaných bodov metódou spätného rajónu.
- Určenie matice plánu a vektoru redukovaných meraní

|                    | Príklad 1 | Príklad 2 | Príklad 3 |
|--------------------|-----------|-----------|-----------|
| Počet hrubých chýb | 0         | 2         | 28        |

Tabuľka 11.1: Rozdelenie hrubých chýb

- Stanovenie množiny odľahlých meraní metódou LMS na základe vstupných dát
- Určenie odhadu neznámych parametrov metódou MNČ a LMS

#### Parametre LMS:

- Max počet náhodne generovaných vzorov: 10 000
- Prahová hodnota: 2.5 násobok smerodajnej odchýlky robustných odhadov
- Veľkosť náhodného vzoru: 51%
- Počet meraní: 66
- Počet neznámych: 2

### 11.3 Zhodnotenie výsledkov

Z výsledku je patrné, že pre sústavu rozmerov  $66 \times 2$  postačuje na určenie správneho odhadu metódou LMS 10 000 kombinácií aj pri kontaminácii približne 40% meraní. Rozpoznanie odľahlých meraní sa zhoršuje s ich rastúcim počtom. Dôvodom je, že so zvyšujúcim počtom odľahlých meraní sa znižuje počet náhodných vzorov nimi nezaťažených. Pokles mediánu potom pri výpočte s náhodným vzorom, ktorý stále obsahuje odľahlé meranie nieje tak výrazný a meranie rozpoznané na základe prekročenia dvojnásobku smerodajnej odchýlky robustných vyrovnaní nieje zaradené do výsledného histogramu. Na obrázkoch 11.3, 11.4 a 11.5 je zelenou farbou zobrazené meranie obsahujúce hrubú chybu. Červenou farbou sú zobrazené početnosti nesprávne určených meraní. Na obrázku 11.2 sú zobrazené vyrovnané opravy rôzne kontaminovaných meraní. V tomto konkrétnom prípade sa hrubá chyba v meranej dĺžke prejaví v značnej miere na jej oprave. Zbytok hrubej chyby je rozdelený medzi opravy ostatných dĺžok. Opravy vyrovňovaných uhlov sú touto chybou zaťažené len nepatrne. Všeobecne môžeme rozdeliť skupinu meraní na skupinu *lepšiu*, v ktorej sú iba potenciálne nekontaminované merania a naopak skupinu *horšiu*, ktorá obsahuje

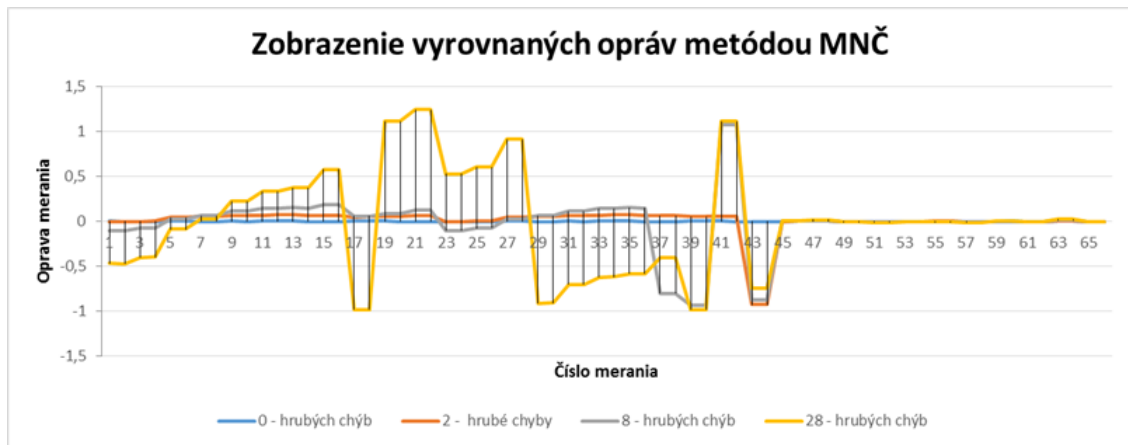
| Metóda | Počet kontaminovaných meraní | X [m]             | Y [m]             |
|--------|------------------------------|-------------------|-------------------|
| MNČ    | 0                            | <b>0.00318222</b> | <b>0.00249310</b> |
| MNČ    | 2                            | -0.05609537       | -0.03703865       |
| MNČ    | 8                            | -0.08787683       | -0.15977342       |
| MNČ    | 28                           | -0.13196252       | -0.59508744       |
| LMS    | 0                            | 0.00397480        | 0.00301887        |
| LMS    | 2                            | 0.00392239        | 0.00306044        |
| LMS    | 8                            | 0.00397250        | 0.00302984        |
| LMS    | 28                           | 0.00321193        | 0.00275135        |

Tabuľka 11.2: Určenie prírastkov vyrovnávaných súradníc v závislosti na použitej metóde a počte kontaminovaných meraní

| Hypotéza | Hodnota               | $\alpha$ | $n$ | $\chi^2_{1-\alpha;n}$ | Zamietnutá hypotéza |
|----------|-----------------------|----------|-----|-----------------------|---------------------|
| 1        | $d = 21.9869$         | 0.05     | 32  | 18.493                | NIE                 |
| 2        | $\hat{q}^2 = 1.29783$ | 0.05     | 34  | 18.493                | ANO                 |

Tabuľka 11.3: Test kvadratickej formy

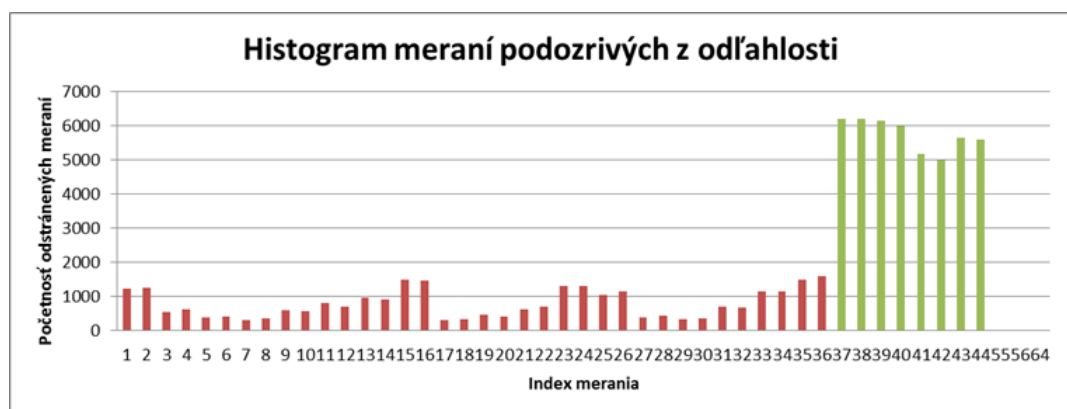
merania podozrivé z odľahlosti. Je nutné podotknúť, že *horšia* skupina veľmi pravdepodobne obsahuje taktiež nekontaminované merania. Z posledného histogramu 11.5 je patrné, že pri vysokom počte odľahlých meraní nieje úplne jasné, či merania podozrivé z odľahlosti sú skutočne merania odľahlé. Taktiež nieje zreteľné, či *lepšia* skupina kontaminované merania neobsahuje. Pomôžme si teda v prípade posledného príkladu, kde je meraniam rozdelených 28 hrubých chýb testom kvadratickej formy popísanom v kapitole 5.5. Tento test bol zvolený preto, že netestujeme jedno potenciálne odľahlé marenie ale celú ich skupinu. Pomocou rovnice 5.38 a 5.39 určíme hodnoty uvedené v tabuľke 11.3. Zo zamietnutia prvej z hypotéz môžeme usúdiť, že odľahlé merania sú obsiahnuté iba v *horšej* skupine meraní. A z potvrdenia hypotézy druhej sa môžeme taktiež domnievať že, *horšia* skupina obsahuje väčšinu odľahlých meraní. Testovaním hypotéz ale nemôžeme určiť, či *horšia* skupina obsahuje iba odľahlé merania a už vôbec nie ich počet.



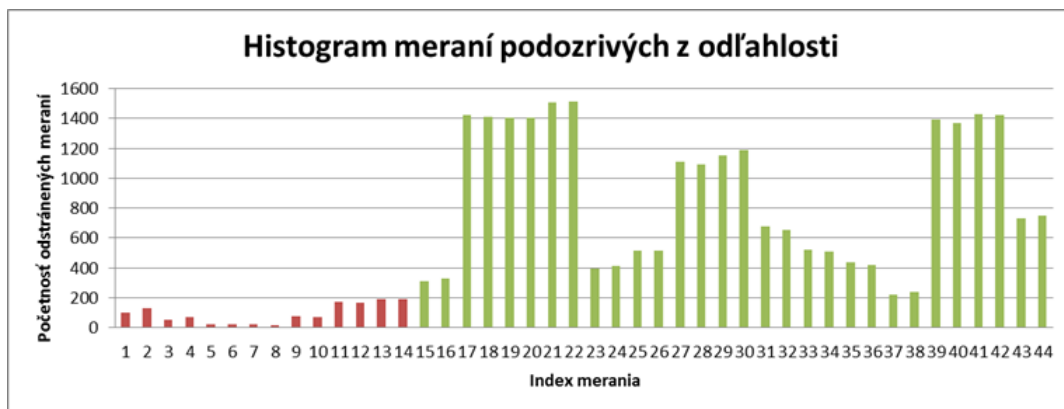
Obrázok 11.2: Hodnoty vyrovnaných opráv v závislosti na počte kontaminovaných meraní



Obrázok 11.3: Histogram početnosti meraní rozpoznaných metódou LMS - 2 hrubé chyby



Obrázok 11.4: Histogram početnosti meraní rozpoznaných metódou LMS - 8 hrubých chýb



Obrázok 11.5: Histogram početnosti meraní rozpoznaných metódou LMS - 28 hrubých chýb

## 12 Diskrétne rozdelenie pravdepodobnosti generovania náhodného vzoru geodetických meraní

Hlavný problémom minimalizácie mediánu štvorcov reziduí je fakt, že neexistuje spoľahlivá metóda, ktorá by dokázala systematicky znižovať medián v každom kroku výpočtu. Jeden krok výpočtu znamená určenie hodnoty mediánu štvorcov reziduí na podmnožine meraní. Nazvime podmnožinu meraní vzorom sústavy. Tento vzor sústavy vznikne odstránením meraní z vektora redukovaných meraní a matice plánu. Keďže neexistuje žiaden vzťah medzi vzorom z predošlého kroku a kroku nasledujúceho, tak nieje možné na základe predošlých výsledkov určiť vstupné predpoklady pre výber ďalších náhodných vzorov. Mnoho riešení sa preto obmedzilo iba na stratégiu výpočtu opakovaným generovaním náhodných vzorov. Pri dostatočnom množstve opakovaní je možné nájsť optimálne riešenie. Problém je, že so zvyšovaním počtu meraní a ich hrubých chýb sa neúmerne zvyšuje potrebný počet náhodne generovaných vzorov. V prípade vyrovnaní geodetických sietí musíme zohľadniť to, že nieje predom známy počet odľahlých meraní a ani ich veľkosť. Taktiež neplatí to, že práve odľahlé merania budú vykazovať najväčšie opravy. Z toho vyplýva, že nemôžeme dopredu určiť veľkosť náhodného vzoru sústavy a generovať náhodne vzory len jednej veľkosti. Aj napriek tomu, že opravy nemajú vypovedajúcu informáciu o veľkosti chýb meraní, tak môžeme informáciu o veľkosti opráv nejakým spôsobom využiť. Až doposiaľ boli vylučované merania generované náhodne s rovnomerným rozdelením. Nástroje jazyka C++11 umožňujú generovať pseudonáhodné čísla pomocou rozdelenia pravdepodobnosti diskkrétnej náhodnej veličiny. Tento nástroj nám umožňuje vyberať náhodné merania s určitou pravdepodobnosťou. Hlavnou myšlienkou je využiť vyrovnané opravy k zmene pravdepodobnosti vylúčenia meraní. Merania, ktoré vykazujú veľké hodnoty vyrovnaných opráv, sa budú vylučovať častejšie než merania s menšími hodnotami opráv. Výhodou tohto riešenia je, že sa meranie z výpočtu nevyľúči úplne, len sa v ďalších krokoch výpočtu bude do vzoru generovať s menšou pravdepodobnosťou. Pokiaľ sa v ďalších krokoch ukáže, že meranie ma znovu nízku opravu, tak sa mu pravdepodobnosť odstránenia zo vzoru postupne naspäť znižuje. V ďalších sekciách bude popísaný spôsob rozdelenia pravdepodobností odstránenia meraní v závislosti na vyrovnaných opravách. Odstránenie veľkého počtu meraní môže viesť k vzniku singularity. Práve z tohto dôvodu je riešenie sústavy algebraických rovníc riešených algoritmom SVD s plnými maticami.

## 12.1 Rozdelenie pravdepodobnosti diskkrétnej náhodnej veličiny

Diskrétna náhodná veličina  $X$  nadobúda konečný počet hodnôt  $x_j$ . Pravdepodobnosť  $f(x_j)$  výskytu náhodnej veličiny  $s_i$  je definovaná ako rozdiel dvoch intervalov  $P(X(s_i) = x_j) = P(X^{-1}(x_j))$ . [Koch-22]

*Definícia:*  $X$  je nazývaná diskrétna náhodná veličina práve vtedy, keď nadobúda konečný počet hodnôt. Pokiaľ  $x_1, x_2, \dots, x_n$  sú tieto hodnoty a  $f(x_1), f(x_2), \dots, f(x_n)$  sú ich pravdepodobnosti, potom je  $f(x_i)$  funkcia hustoty pravdepodobnosti  $X$ .

Funkcia  $f(x_i)$  hodnôt  $x_i$  diskkrétnej náhodnej veličiny  $X$  reprezentuje pravdepodobnosť, ktorá spĺňa nasledujúce tri axiómy.

Reálne číslo  $P(A)$  reprezentuje každú udalosť  $A$  zo  $\sigma$ -algebry  $Z$  podmnožiny z množiny  $S$  elementárnych udalostí. Pre nasledujúce axiómy platí:

1. *Axióm:* Reálne číslo  $P(A) \geq 0$  reprezentuje každú udalosť  $A$  zo  $Z$ .  $P(A)$  je nazývaná pravdepodobnosť  $A$
2. *Axióm:* Pravdepodobnosť istej udalosti je rovná jednej  $P(S) = 1$
3. *Axióm:* Pokiaľ  $A_1, A_2, \dots$  je postupnosť udalosti zo  $Z$ , ktoré sú spočítateľné a ktoré sa vzájomne vylučujú, tak  $A_i \cap A_j = \emptyset$  pre  $i \neq j$  je potom  $P(A_1 \cup A_2 \cup \dots) = P(A_1) + P(A_2) + \dots$

Trojica  $(S, Z, P)$  je nazývaná priestor pravdepodobnosti.

Pre hodnoty  $x_i$  z  $X$  zo vzájomne vylučujúcich sa udalosti získame podmienky pre funkciu hustoty pravdepodobnosti  $f(x_i)$

$$f(x_i) \geq 0 \quad a \quad \sum_{i=1}^n f(x_i) = 1 \quad alebo \quad \sum_{i=1}^{\infty} f(x_i) = 1 \quad (12.1)$$

Pravdepodobnosť udalosti  $X < x_i$  je daná ako

$$P(X < x_i) = \sum_{i < j} f(x_j) = \quad (12.2)$$



## 12.2 Algoritmus diskretného rozdelenia pravdepodobností v závislosti na veľkosti vyrovnaných opráv

Vstupom do výpočtu je homogenizovaná sústava tvorená maticou plánu  $\mathbf{A}$  a vektorom redukovaných meraní  $\ell$ . Výsledkom je odhad neznámych parametrov  $\mathbf{x}$ .

Algoritmus riešenia:

1. Vygenerovanie počtu odstránených riadkov sústavy  $k$ , pre ktoré platí, že  $1 \leq k \leq N/2$  v prípade, že  $M > 2N$  alebo  $1 \leq k \leq N$  v prípade, že  $M > N$ , kde  $M$  je počet meraní a  $N$  je počet neznámych parametrov.
2. Pre  $i = 1 \dots k$  sa vygeneruje číslo odstráneného riadku z diskretného rozdelenia a odstráni sa z matice plánu a vektoru redukovaných meraní. Týmto spôsobom vznikne vzor sústavy.
3. Vyriešenie sústavy lineárnych rovníc metódou SVD. Partikulárne riešenie je označené ako  $\bar{\mathbf{x}}$ .
4. Výpočet vyrovnaných opráv partikulárneho riešenia  $\mathbf{A}\bar{\mathbf{x}} - \mathbf{b} = \mathbf{v}$ .
5. Určenie mediánu štvorcov vyrovnaných opráv.
6. V prípade, že platí  $med_{i-1} > med_i$ , tak sa uloží odhad neznámych parametrov ako najlepšie riešenie a zároveň sa rozdelia pravdepodobnosti pre generovanie ďalšieho vzoru.
7. Pokiaľ bol dosiahnutý maximálny počet iterácií alebo medián dosiahol požadovanej hodnoty, tak sa výpočet ukončí. V opačnom prípade sa výpočet opakuje od kroku 1.

## 12.3 Stratégie rozdelenia pravdepodobnosti

V predošlej sekcii bol predstavený algoritmus riešenia minimalizácie mediánu založeného na diskretnom rozdelení pravdepodobnosti vylúčenia meraní v závislosti na veľkosti vyrovnaných opráv výpočtu. Nebol ale uvedený predpis ako pravdepodobnosti rozdeliť. V tejto práci bolo navrhnutých niekoľko metód rozdelenia pravdepodobnosti a to lineárna, exponenciálna a rozdelenie pravdepodobnosti skupinám meraní teda nespojité rozdelenie. Vo všetkých prípadoch platí, že sa pravdepodobnosť vylúčenia odľahlého merania zvyšuje s veľkosťou opravy, ale len do polovice

meraných hodnôt. Druhej polovici meraní sa naopak pravdepodobnosti znižujú. To znamená, že meraniu, ktorému bola v predošlom kroku zvýšená pravdepodobnosť vylúčenia, sa môže v ďalšom kroku pravdepodobnosť znížiť na pôvodnú úroveň. Predstavme si napríklad, že v prvom kroku zvýšime pravdepodobnosť odstránenia merania s najväčšou opravou o 10%. V kroku druhom sa meranie nevylúči, ale už vykazuje opravu najmenšiu. V tomto prípade sa potom naspäť zníži pravdepodobnosť jeho vylúčenia o 10%.

### Lineárne rozdelenie pravdepodobnosti

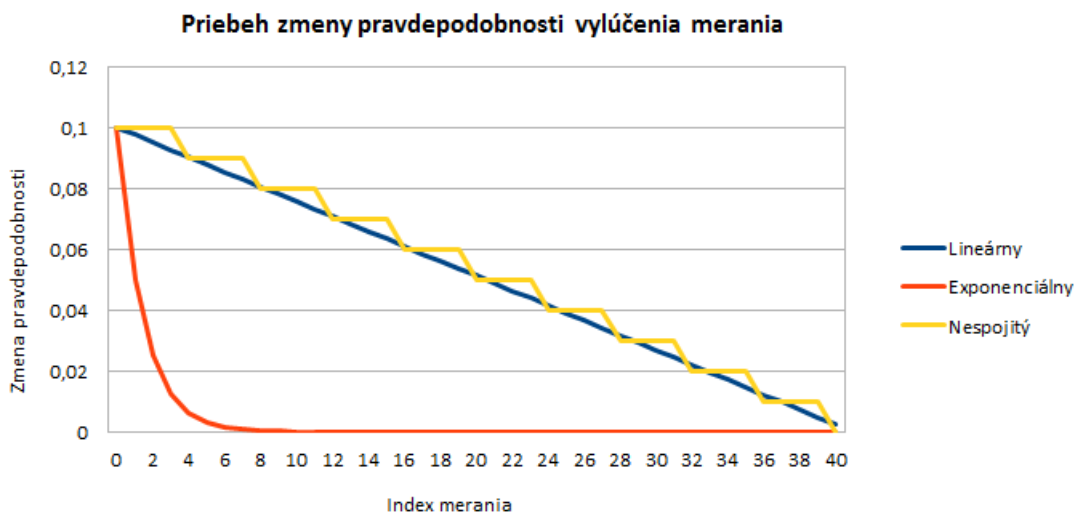
Lineárne rozdelenie pravdepodobnosti má ako vstupný parameter maximálnu hodnotu zvýšenia pravdepodobnosti  $r_{max}$  vylúčenia merania z množiny observácií. Hodnota  $r_{max}$  sa priradí najväčšej oprave. To znamená, že najväčšia oprava bude mať pravdepodobnosť vylúčenia zo skupiny meraní o 10% väčšiu ako ostatné merania. Druhá najväčšia oprava bude mať priradenú pravdepodobnosť  $r = r_{max} - \frac{r_{max}}{N/2}$  a obdobne sa bude pravdepodobnosť postupne znižovať  $r_i = r_{i-1} - \frac{r_{max}}{N/2}$  až po  $N/2$  opráv. Z toho vyplýva, že polovici meraní s horšími opravami sa budú pravdepodobnosti zvyšovať a polovici druhej znižovať. Aby meranie nebolo nikdy vylúčené úplne, tak boli definované prahové hodnoty pravdepodobnosti, ktoré môžu merania nadobúdať.

### Exponenciálne rozdelenie pravdepodobnosti

Exponenciálne rozdelenie pravdepodobnosti je použité obdobne ako v predošlom lineárnom rozdelení, avšak s tým rozdielom, že pokles pravdepodobností klesá s logaritmom o základe 2. V prvom kroku je hodnota  $r_{max}$  priradená meraniu s maximálnou opravou. V kroku druhom je to hodnota  $r = r_{max} - \frac{r_{max}}{2}$ . Pokiaľ označíme hodnotu  $\frac{r_{max}}{2}$  ako  $d$ , tak je pokles definovaný ako  $r_i = r_{i-1} - d$ ,  $d_i = d_{i-1}/2$ . Obdobne ako v predošlom prípade lineárneho rozdelenia sú lepšej polovice meraní pravdepodobnosti vylúčenia znižované.

### Rozdelenie pravdepodobnosti skupinám meraní

Ako už z názvu vyplýva, bude sa skupinám meraní zvyšovať či znižovať pravdepodobnosť vylúčenia znovu na základe vyrovnaných opráv. Prvú polovicu meraní rozdelíme na  $k$  skupín a skupine meraní s najväčšou opravou zvýšime pravdepodobnosť vylúčenia o  $r_{max}$ . Skupine s najmenšími opravami horšej polovice meraní zvýšime pravdepodobnosť o  $r_{min}$ . Zmena hodnoty pravdepodobnosti  $i$ -tej skupiny je teda definovaná ako  $r_i = r_{i-1} - \frac{r_{max}}{k}$ .



Obrázok 12.1: Rozdelenie pravdepodobnosti meraniam

## 12.4 Implementácia diskretného rozdelenia pravdepodobnosti v C++11

Metóda rozdelenia pravdepodobnosti diskretnej náhodnej veličiny je implementovaná v jazyku C++11 pomocou triedy `class discrete_distribution`. Trieda produkuje hodnoty náhodných čísel z diskretného rozdelenia, kde každá možná hodnota ma preddefinovanú pravdepodobnosť danú vzťahom

$$P(i|w_0, w_1, \dots, w_{n-1}) = \frac{w_i}{\sum_{k=0}^{n-1} w_k}, 0 \leq i \leq n \quad (12.3)$$

kde  $w$  je množina  $n$  nezáporných váh. Pravdepodobnosť každej hodnoty z množiny  $n$  je daná váhou delenou sumou všetkých váh.

```
1 typedef std::vector<double>::const_iterator CIT;
2 void DiscreteLms::generateRandomSampleSet(CIT beg, CIT end)
3 {
4     std::random_device rd;
5     std::mt19937 generator(rd());
6     std::discrete_distribution<> d(beg, end);
7
8     while(m_deleted_rows_set.size() <= m_K)
9     {
10         m_deleted_rows_set.insert(d(generator));
11     }
12 }
```

Kód 12.1: Algoritmus generovania diskretných náhodných veličín

Trieda `discrete_distribution` má v konštruktoze ako vstupné parametre konštantné iterátory na štandardný kontajner, ktorý obsahuje hodnoty pravdepodobnosti vylučovaných riadkov. Náhodné čísla sa generujú až kým množina odstránených meraní nieje rovná predom určenému počtu. Vektor pravdepodobnosti sa mení v každom kroku výpočtu. Výpočet vektoru pravdepodobností je vykonaný pomocou funktoru. Každá stratégia rozdelenia pravdepodobnosti má definovaný vlastný funktor. Funktor vracia pointer na vektor pravdepodobnosti. Vstupom do funktoru je konštantná referencia na vektor vyrovnaných opráv.

```
1 class Linear
2 {
3 public:
4
5     Linear(unsigned int size) :
6         m_size(size), m_distribution(m_size, 1.0/m_size){}
7
8     std::vector<double>* operator()(const std::vector<DiscreteLms::Pair> &vsort)
9     {
10         // implementace ...
11         return &m_distribution;
12     }
13
14 private:
15     unsigned int m_size;
16     std::vector<double> m_distribution;
17 };
```

Kód 12.2: Funktor lineárnarného diskretného rozdelenia

Funktory sú uložené v štandardnom kontajnery a potom postupne volané ako argumenty funkcie `solve()` triedy `DiscreteLms`, ktorá postupne vykonáva vykonáva všetky kroky výpočtu.

```
1 vector< std::function< std::vector<double>*
2 (const std::vector<DiscreteLms::Pair> &vsort)> > functorsVec;
3
4 functorsVec.push_back(Linear(A->rows()));
5 functorsVec.push_back(Eexponential(A->rows()));
6 functorsVec.push_back(ProbabilityGroups(A->rows()));
7 functorsVec.push_back(Random(A->rows()));
8
9 for(unsigned int i=0; i<functorsVec.size();i++)
10 {
11     DiscreteLms dLms(A,b);
12     dLms.solve(functorsVec[i]);
13 }
```

Kód 12.3: Volanie funkcie `solve()` vo funkcii `main()`

Kompletnú implementáciu metódy rozdelenia pravdepodobnosti diskretných náhodných veličín je možné naklonovať s *git* repozitára zverejneného na <https://github.com/gyorigab/discreteLms>.

## 12.5 Možnosti paralelizácie výpočtu

So zvyšovaním rozmerov sústav, ktoré obsahujú veľké množstvo hrubých chýb, je nutné zvyšovať počet iterácií algoritmu popísaného v sekcii 12.2. Na základe výsledkov z vykonaných experimentov je možné konštatovať, že pre sústavy s počtom meraní 100, počtom neznámych parametrov 40 a počtom hrubých chýb 20 je nutné vykonať pre určenie správneho odhadu stratégiou lineárneho rozdelenia pravdepodobnosti približne 1000 iterácií. So zvyšujúcim sa počtom meraní bude stúpať náročnosť výpočtu sústav lineárnych rovníc metódou SVD a zároveň bude pre určenie správneho odhadu potrebný väčší počet iterácií. Vyššie popísaný algoritmus môže vykonávať výpočet na niekoľkých procesoroch či počítačoch úplne nezávisle. Každá výpočtová jednotka vyprodukuje odhad vyrovnaných parametrov a minimálny medián. Za výsledok bude považovaný odhad neznámych parametrov s najmenšou hodnotou mediánu.

## 12.6 Zhodnotenie výsledkov

Overenie navrhutej metódy minimalizácie mediánu štvorcov reziduí bolo vykonané na geodetickom meraní, ktorého zápisník je priložený v prílohe A, ale taktiež na veľkom množstve vygenerovaných sústav. Sústavy boli generované obdobne ako v kapitole 8.1, avšak s odlišnými parametrami. Parametre náhodne generovaných sústav sú uvedené v tabuľke 12.1. Cieľom metódy rozdelenia pravdepodobnosti diskretných veličín v závislosti na veľkosti vyrovnaných opráv meraných veličín bolo vylepšenie konvergenzie mediánu bez akejkoľvek znalosti či predpokladu o počte, či veľkosti odľahlých meraní. Táto metóda nijakým spôsobom neupravuje pôvodnú informáciu meraní narozdiel od  $M$ -odhadov či metódy LMS uvedenej v kapitole 13, ktorá používa pre zlepšenie konvergenzie mediánu zmenu váh meraní. Uvedená metóda iba upravuje pravdepodobnosť výberu merania do vzoru sústavy. Zo skupiny vyrovnaných sústav boli vybrané dve sústavy, ktorých rozmery a počet hrubých chýb je popísaný v tabuľke 12.2. Odhad každej z vygenerovaných sústav bol určený metódou LMS s lineárnym a exponenciálnym rozdelením pravdepodobností generovania vzoru sústavy s rozdelením pravdepodobnosti skupinám meraní a náhodným výberom vzoru sústavy. Konvergenzia mediánu je zobrazená v grafoch 12.2 a 12.3. Zo získaných výsledkov vyplýva, že metóda lineárneho rozdelenia pravdepodobností

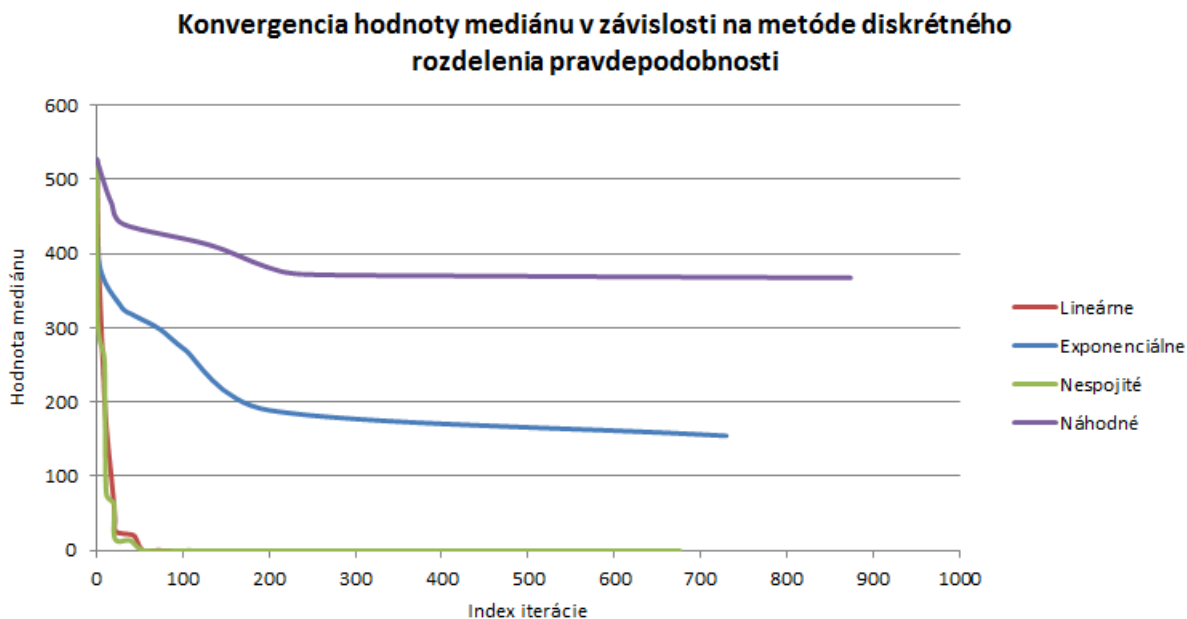
|                        | Rozdelenie           | Min. hodnota | Max. hodnota |
|------------------------|----------------------|--------------|--------------|
| Náhodné chyby          | normálne             | 0            | 0.01         |
| Hrubé chyby            | normálne             | 10           | 100          |
| Koeficienty mat. plánu | rovnorné reálne      | -0.5         | 0.5          |
| Počet riadkov          | rovnorné celočíselné | 8            | 100          |
| Počet stĺpcov          | rovnorné celočíselné | 2            | $(N/2)-1$    |
| Počet hrubých chýb     | rovnorné celočíselné | 1            | $(N/2)-1$    |

Tabuľka 12.1: Tabuľka parametrov generovaných sústav

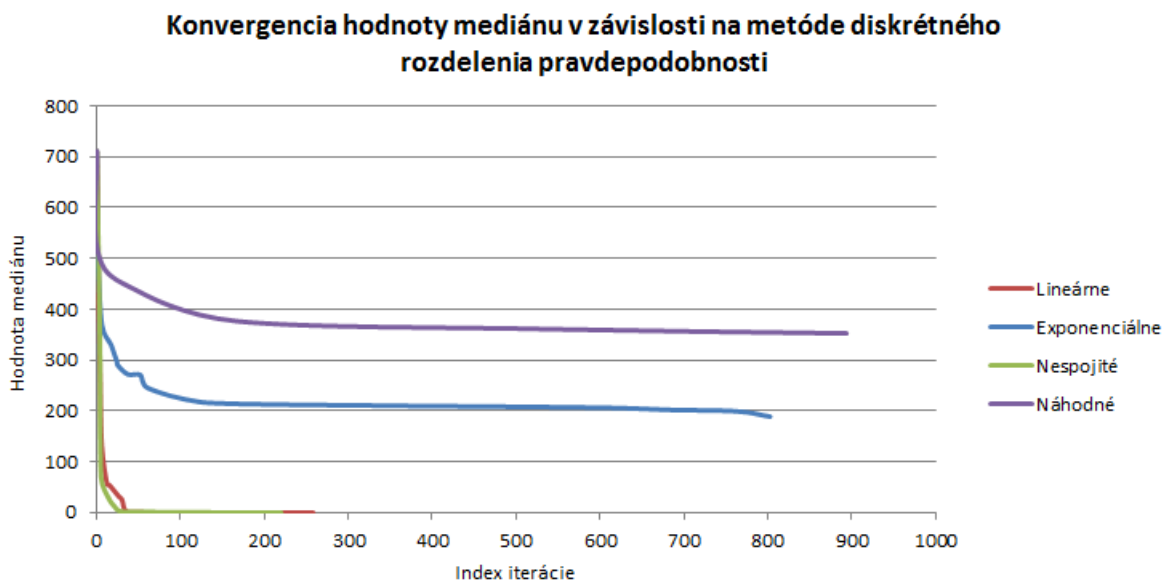
|           | Počet riadkov | Počet stĺpcov | Počet hrubých chýb |
|-----------|---------------|---------------|--------------------|
| Príklad 1 | 88            | 22            | 27                 |
| Príklad 2 | 91            | 15            | 36                 |

Tabuľka 12.2: Tabuľka rozmerov určených sústav

a rozdelenia pravdepodobností skupinám meraní značne vylepší konvergenciu mediánu. Minimalizovaný medián prudko klesá v prvých 100 iteráciách a potom sa jeho hodnota mení už len nepatrne. Z toho vyplýva, že odhad neznámych parametrov je už dostatočne presný po týchto 100 krokoch. Stratégia exponenciálneho rozdelenia pravdepodobnosti zvyšuje pravdepodobnosť odstránenia v podstate len niekoľkým meraniam s najväčšími opravami. Táto stratégia by bola vhodná pre merania s nízkym počtom odľahlých meraní. V uvedených príkladoch je ale počet odľahlých meraní pomerne vysoký a niektoré merania podozrivé z odľahlosti sú do vzoru za použitia tejto stratégie opakovane vyberané. V tomto prípade medián konverguje iba pozvoľne a v oboch uvedených prípadoch je odhad neznámych parametrov značne nepresný. Najhoršiu konvergenciu mediánu vykazuje metóda s náhodným výberom vzorov, kde pravdepodobnosť výberu vylúčeného merania je v každom kroku rovnaká pre všetky observácie. Medián v tomto prípade klesá veľmi pomaly a aj po 1000 iteráciách má stále veľmi vysokú hodnotu. Odhad neznámych parametrov touto metódou pri 1000 násobnom počte opakovaní je značne nepresný. Na základe vykonaných experimentov je možné tvrdiť, že voľba stratégie lineárneho rozdelenia pravdepodobnosti značne vylepší konvergenciu mediánu a pre sústavy o rozmere  $100 \times 50$  s 20% odľahlými meraniami určí správny odhad už po 100 iteráciách.



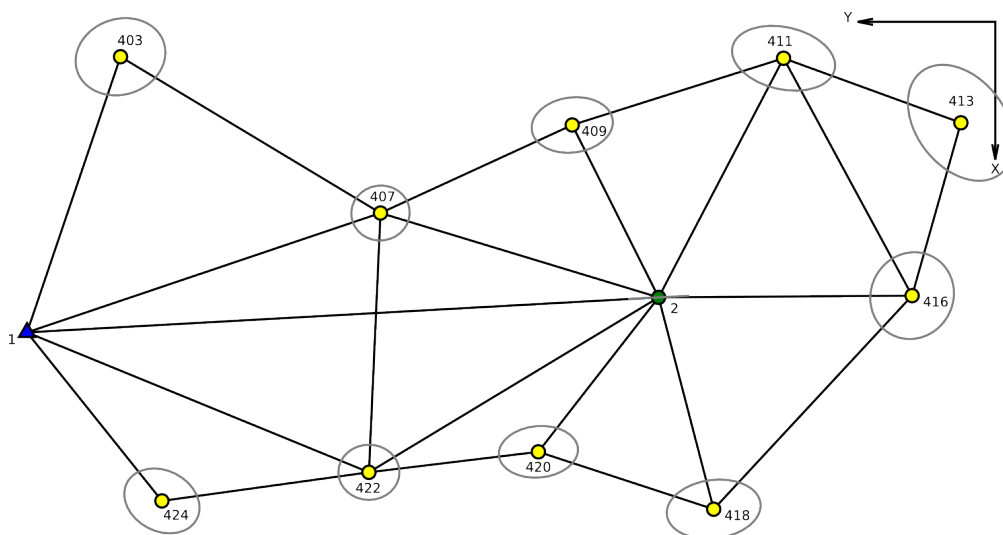
Obrázok 12.2: Priebeh zmeny mediánu v závislosti na vybranej stratégii diskretného rozdelenia pravdepodobnosti - Príklad 1



Obrázok 12.3: Priebeh zmeny mediánu v závislosti na vybranej stratégii diskretného rozdelenia pravdepodobnosti - Príklad 2

## 13 Výpočet geodetickej siete metódou LMS použitím metódy zmeny váh meraní

Riešenie pomocou LMS je určené opakovaným vyrovnaním vybraných podmnožín meraní v závislosti na veľkosti hodnoty mediánu. Do výpočtu metódy LMS vstupuje matica plánu  $\mathbf{A}$ , vektor redukovaných meraní  $\ell$  a váhová matica  $\mathbf{P}$ . Nazvime tieto parametre ako vstupné parametre výpočtu. V prvom kroku je potrebné vykonať homogenizáciu meraní. Homogenizácia sa vykoná na kópii matice plánu a vektora redukovaných meraní tak, aby boli pôvodné vstupné parametre zachované pre výpočet reziduí. Z homogenizovaných parametrov sa vyberie vzor matice a vypočíta sa vektor neznámych parametrov rov. 10.15. Pomocou tohoto vektora je možné vyhotoviť vektor štvorcov reziduí a z neho vypočítať medián. Medián je počítaný z rôznych vzorov tak, aby jeho hodnota konvergovala k minimu. Postup zabezpečenia konvergenencie bude popísaný v tejto kapitole. Zdrojové kódy metód popisovaných v tejto kapitole sú zverejnené na <https://github.com/gyorigab/ReweightLms>.



Obrázok 13.1: Meranie v geodetickej sieti [Gama–29]

### 13.1 Homogenizácia meraní

Homogenizácia matice plánu a vektoru redukovaného meraní sa vykoná v závislosti na tom, aká bude použitá metóda združených gradientov. Ako už bolo popísané v predošlých kapitolách, na riešenie neznámych parametrov môže byť použitá metóda združených gradientov s explicitným vyhotovením normálnych rovníc a bez explicitného vyhotovenia normálnych rovníc. V prvom prípade sa homogenizácia



vyhotovuje pri vytvorení normálnych rovníc podľa rov. 10.15. V druhom prípade však do metódy združených gradientov vstupuje priamo matica plánu a normálne rovnice sa vyhotovujú v priebehu riešenia neznámych parametrov. To znamená, že v metóde združených gradientov sa vykonáva operácia  $\mathbf{A}^T \mathbf{A}$  teda násobenie matice plánu samu so sebou. Rovnicu  $\mathbf{A}^T \mathbf{P} \mathbf{A}$  modifikujeme na rovnicu  $\mathbf{A}^T \mathbf{P}^{\frac{1}{2}} \mathbf{P}^{\frac{1}{2}} \mathbf{A}$ . Pre homogenizovanú maticu plánu platí  $\tilde{\mathbf{A}} = \mathbf{P}^{\frac{1}{2}} \mathbf{A}$  a pre vektor redukovaných meraní  $\tilde{\ell} = \mathbf{P}^{\frac{1}{2}} \ell$ . Algoritmus gradientov teda rieši  $\tilde{\mathbf{A}}^T \tilde{\mathbf{A}} \mathbf{x} = \tilde{\mathbf{A}}^T \tilde{\ell}$

## 13.2 Stratégia výpočtu LMS

Voľba vhodnej stratégie by mala zabezpečovať konvergenciu mediánu k minimu. Snahou je, aby stratégia viedla k získaniu optimálneho riešenia v čo najkratšom časovom intervale. Skôr než sa bude popisovať vlastný postup výpočtu, označme si premenné vstupujúce do výpočtu.

- $\mathbf{A}$  - vstupná matica plánu
- $\ell$  - vstupný vektor redukovaných meraní
- $\tilde{\mathbf{A}}$  - homogenizovaná matica plánu
- $\tilde{\ell}$  - homogenizovaný vektor redukovaných meraní
- $\mathbf{A}_{\text{sr}}$  - vzor matice plánu obmedzený o náhodne vybrané riadky
- $\ell_{\text{sr}}$  - vzor vektoru redukovaných meraní obmedzený o náhodne vybrané riadky
- $\mathbf{A}_{\text{sd}}$  - vzor matice plánu obmedzený o náhodne vybrané riadky a zároveň obmedzený určitý počet riadkov s najväčšími štvorcami reziduami
- $\ell_{\text{sd}}$  - vzor matice plánu obmedzený o náhodne vybrané riadky a zároveň obmedzený určitý počet riadkov s najväčšími štvorcami reziduami
- *med* - vypočítaný medián štvorcov reziduí

### 13.2.1 Vyhotovenie vzoru matice plánu

Vzor matice plánu je vlastne matica plánu, z ktorej je za určitých predom definovaných podmienok odstránená množina riadkov. Obdobne sú odstránené riadky redukovaného merania. Odstránenie riadkov je ekvivalentné s odstránením meraní v geodetickej sieti. Odstránenie riadkov však môže byť príčinou niekoľkých zásadných problémov.

- z regulárnej matice sa po odstránení riadkov stane matica singulárna
- geodetická sieť sa rozpadne na niekoľko podsietí
- stratí sa merítka siete. To sa môže stať v prípade, že sú v sieti merané uhly a dĺžky a odstránia sa všetky merania dĺžok.

Nie je teda možné úplne odstránenie meraní nepatriacich do podmnožiny. Odstránenie je možné vykonať fiktívne, zavedením extrémnych váh. Malé váhy sú priradené meraniam, ktoré nepatria do vybranej podmnožiny pozorovaní, takže budú mať len minimálny vplyv na odhad neznámych parametrov. Obdobne je možné malú váhu priradiť meraniu podozrivého z odľahlosti. Zaveďme si teda maticu extrémnych váh, ktorú označíme  $\tilde{\mathbf{P}}$ . Matica extrémnych váh je diagonálna matica. Vzor matice sa teda získa vynásobením homogenezovanej matice plánu a vektoru redukovaných meraní maticou extrémnych váh.

$$\tilde{\mathbf{P}}\tilde{\mathbf{A}}\mathbf{x} = \tilde{\mathbf{P}}\ell \quad (13.1)$$

to znamená, že  $\mathbf{A}_{\text{sr}} = \tilde{\mathbf{P}}\tilde{\mathbf{A}}$  resp.  $\ell_{\text{sr}} = \tilde{\mathbf{P}}\ell$  v prípade, že odstraňujeme náhodné riadky, teda v matici extrémnych váh sú pozície hodnôt extrémnych váh náhodne vygenerované.

### 13.2.2 Stratégie váhovania

Zvolenými váhami sa násobí vždy určitá užívateľom definovaná množina observácií s pravidla s hodnotami rezidui vyššími než je vypočítaný medián. Vzhľadom k robustnosti zvolenej metódy je možné preváhovať až 50% meraní. Voľba rozdelenia váh môže mať značný vplyv na priebeh hodnoty mediánu a tým pádom aj na celkový výsledok výpočtu. Merania podozrivé z odľahlosti, teda nie sú z výpočtu odstránené, ale sú pozmenené v závislosti na zvolenej stratégii.

#### Stratégia extrémnych váh:

Prvou zo stratégií je preváhovanie zadaného počtu riadkov s najväčšími štvorcami reziduí nejakou zvolenou extrémnou hodnotou. Počet riadkov, ktoré sa majú preváhovať je možné nakonfigurovať ako absolútnu hodnotu, či hodnotu vyjadrenú v percentách vzhľadom k počtu riadkov matice. Veľkosť váhy je možné nastaviť v konfiguračnom súbore programu pre výpočet LMS alebo je možné použiť implicitnú hodnotu, ktorá je definovaná ako reciproká hodnota normy matice plánu. Meranie podozrivé z odľahlosti sa extrémnou váhou násobí iba raz a to aj v prípade, že jeho reziduum dostatočne neklesne a meranie bude patriť do množiny s najväčšími observáciami. Metóda váhovania extrémnymi váhami je určitá alternatíva k úplnému

odstráneniu observácií podozrivých z odľahlosti. Úplné odobranie merania z výpočtu môže viesť k singularite riešenej sústavy. Je preto nutné voliť extrémne váhy tak, aby nespôsobovali lineárnu závislosť meraní, ale na druhu stranu tak, aby kontaminovaná hodnota observácie mala len minimálny vplyv na ďalší výpočet.

#### Stratégia rozdelenia váh v definovanej škále:

Druhou stratégiou je preváhovanie požadovaného počtu hodnôt observácií. Hodnoty váh sú rozdelené v škále tak, že riadok s najväčším štvorcem rezídua je prenasobený extrémnou váhou a riadok s najmenším štvorcem rezídua váhou určenou podielom extrémnej váhy a počtom riadkov. Rozdelenie váh medzi observáciu s najväčšou a najmenšou hodnotou rezídua je lineárne.

#### Stratégia násobného preváhovania:

Treťou zo stratégií je násobné preváhovanie observácii a to tak, že sa po každom kroku vynásobia riadky s observáciami vykazujúcimi najväčšie reziduá nejakou zvolenou váhou. To znamená, že čím viac krát riadok matice plánu vykazuje väčšie reziduá tak vstupuje do výpočtu s menšou váhou.

### 13.2.3 Generovanie náhodného vzoru matice plánu

Náhodne generovaný vzor matice plánu vstupuje do výpočtu na začiatku každého cyklu a s týmto vzorom sa operuje do doby, kým medián štvorcov rezidui klesá. V prípade, že medián prestal konvergovať k minimálnej hodnote, vyberie sa z matice plánu nový náhodný vzor. V algoritme výpočtu je zabezpečené, že sa náhodný riadok matice neodstraňuje násobne. Veľkosť náhodného vzoru je definovaná v konfiguračnom súbore pomocou percentuálnej hodnoty. Algoritmus výberu náhodného vzoru generuje rovnomerne rozdelenú pseudo-náhodnú sériu celých čísel reprezentujúcich čísla riadkov. Táto séria je generovaná C++ funkciou *rand()*. Pre inicializáciu generátoru je použitá hodnota systémového času. Vzhľadom ku konečnému stavu generátoru vyplýva, že generovaná náhodná postupnosť je periodická. Tento fakt ale nemá vplyv na ďalšie výpočty. Implicitná hodnota veľkosti vzoru matice je 95% z veľkosti vstupnej matice.

### 13.2.4 Algoritmus výpočtu

1. Výpočet homogenizovanej sústavy  $\tilde{\mathbf{A}} = \mathbf{P}^{\frac{1}{2}} \mathbf{A}$ ,  $\tilde{\ell} = \mathbf{P}^{\frac{1}{2}} \ell$
2. Vygenerovanie náhodného vzorku a zostrojenie matice extrémnych váh  $\tilde{\mathbf{P}}$
3. Zhotovenie vzoru matice a vektoru  $\mathbf{A}_{\text{sr}} = \tilde{\mathbf{P}} \tilde{\mathbf{A}}$ ,  $\ell_{\text{sr}} = \tilde{\mathbf{P}} \tilde{\ell}$

4. Výpočet vektoru neznámých parametrov pomocou metódy združených gradientov  $\mathbf{A}_{sr}^T \mathbf{A}_{sr} \tilde{\mathbf{x}} = \mathbf{A}_{sr}^T \ell_{sr}$
5. Výpočet rezidui  $\tilde{\mathbf{v}} = \mathbf{A} \tilde{\mathbf{x}} - \ell$
6. Výpočet mediánu štvorcov rezidui  $med(\tilde{\mathbf{v}}^T \tilde{\mathbf{v}})$
7. Určenie množiny rezidui, ktoré pripadajú meraniam podozrivým z odľahlosti
8. V prvom priebehu výpočtu sa pokračuje na ďalší krok, ináč sa testuje podmienka, či medián vygenerovanej množiny je menší ako posledný nájdený medián, a v prípade, že podmienka nieje splnená, prejde sa na krok 2, ináč sa pokračuje krokom 9.
9. Testuje sa podmienka zastavenia výpočtu a to tak, že sa testuje maximálny počet iterácií alebo viacnásobné určenie minimálneho mediánu. V prípade splnenia jednej z podmienok výpočet končí.
10. Vyhotoví sa matica extrémnych váh  $\tilde{\mathbf{P}}$  na základe posledne určenej množiny meraní podozrivých z odľahlosti. V prípade, že meranie už extrémnu váhu nastavenú má, extrémna váha sa mu znovu nenastaví
11. Zhotovenie vzoru matice a vektoru  $\mathbf{A}_{sd} = \tilde{\mathbf{P}} \tilde{\mathbf{A}}$ ,  $\ell_{sd} = \tilde{\mathbf{P}} \tilde{\mathbf{A}}$
12. Výpočet vektoru neznámých parametrov pomocou metódy združených gradientov  $\mathbf{A}_{sd}^T \mathbf{A}_{sd} \tilde{\mathbf{x}} = \mathbf{A}_{sd}^T \ell_{sd}$
13. Výpočet rezidui  $\tilde{\mathbf{v}} = \mathbf{A} \tilde{\mathbf{x}} - \ell$
14. Výpočet mediánu štvorcov rezidui  $med(\tilde{\mathbf{v}}^T \tilde{\mathbf{v}})$
15. Určenie množiny rezidui, ktoré pripadajú meraniam podozrivým z odľahlosti
16. Test podmienky, či je novo-vypočítaný medián menší než posledný medián a zároveň množina odstránených meraní nedosiahla maximálnej veľkosti. To znamená, že v podmnožine meraní ostalo iba jedno nadbytočné meranie. V prípade splnenia podmienky sa pokračuje sa krokom 10, v druhom prípade sa pokračuje krokom 2.

### 13.3 Určovanie súvislosti geodetickej siete

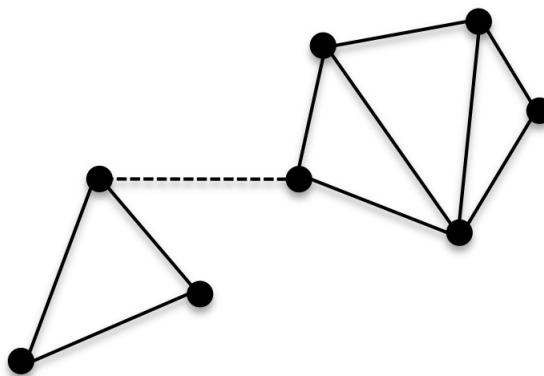
Označenie *súvislá geodetická sieť* bol použitý z terminológie teórie grafov. Pri vyrovnávaní neznámých parametrov geodetickej siete metódou LMS sa zo súboru meraní odstráni predom definovaná náhodná skupina observácií. Tá môže spôsobiť rozpad

siete na niekoľko podsieti a v tomto prípade sa stane geodetická sieť nesúvislá. Nesúvislosť geodetickej siete môže spôsobiť singularitu sústavy lineárnych rovníc, čo môže zapríčiniť ukončenie výpočtu združených gradientov zo značnou chybou riešenia. Metóda združených gradientov nebude konvergovať a zastaví sa prakticky na definovanom maximálnom počte meraní. Takýto výsledok vyrovnaní je pre určenie mediánu štvorcov reziduí nepoužiteľný. Výberu takého vzoru je možné zabrániť niekoľkými spôsobmi. Jedným z nich je priame určenie, či sústava lineárnych rovníc obsahuje lineárne závislé riadky a tým pádom je matica plánu neinvertovateľná. Existuje mnoho metód na určenie singularity. Všetky z nich sú však pomerne neefektívne a výpočtovo pomerne náročné. Druhým spôsobom ako sa potenciálne vyhnúť singularite je nepriamo analýzou geodetickej siete.

Predstavme si teda geodetickú sieť ako graf  $G$  zostavený z vrcholov  $v$  a hrán  $e$ . Vrcholy zodpovedajú vyrovnávaným bodom siete a hrany grafu meraniam v geodetickej sieti. Problém rozpadu geodetickej siete z pohľadu teórie grafov je možné previesť na úlohu hľadania súvislosti grafu. Graf je súvislý práve vtedy, keď existuje cesta medzi dvoma ľubovoľnými bodmi grafu. V súvislom grafe platí, že  $|e| \geq |v| - 1$ , kde  $|v|$  je počet vrcholov a  $|e|$  je počet hrán. Graf je súvislý práve vtedy, keď sú stupne všetkých vrcholov aspoň  $|v|/2$ . U neorientovaného grafu je stupeň vrcholu počet hrán, ktoré do daného vrcholu zasahujú. Komponenta súvislosti sú všetky indukované podgrafy na jednotlivých triedach ekvivalencie súvislosti. Súvislý graf má práve jednu komponentu súvislosti. V teórii grafov sa súvislosť grafu rozdeľuje na silnú a slabú. Silná súvislosť sa vzťahuje na orientované hrany grafu. V našom prípade postačuje určenie komponent slabej súvislosti grafu. Jedným zo spôsobov ako získať komponenty súvislosti grafu, je tzv. prehľadávaním do hĺbky (DFS). Algoritmus DFS je úplný to znamená, že vždy nájde možnú dostupnosť všetkých vrcholov z ľubovoľného počiatočného vrcholu. Algoritmus sa primárne používa pre grafy reprezentované stromovou štruktúrou. Algoritmus je, ale možné použiť aj pre všeobecné grafy. V tom prípade ale algoritmus nemusí nájsť najkratšiu cestu medzi vrcholmi. Časová zložitosť algoritmu je  $O(|v| + |e|)$ . Pre naše použitie je možné algoritmus urýchliť a to z toho dôvodu, že potrebujeme iba zistiť, či daný graf obsahuje práve jednu komponentu súvislosti, ale nepotrebujeme určovať počet týchto komponent. Cyklus sa teda zastaví pri nájdení druhej komponenty súvislosti.

### 13.3.1 Implementácia určenia súvislosti geodetickej siete.

Všetky potrebné informácie o meraných hodnotách, smerodajných odchýlkach a popise vyrovnávaných bodov geodetickej siete vyrovnávanej programom *gama-local* sú popísané súbore vo formáte XML, ktorý je vstupom do programu *gama-local*. Podrobný popis tohto formátu je dostupný na [Gama-29]. V prvom kroku vytvo-



Obrázok 13.2: Súvislosť siete [Connectivity–54]

ríme graf geodetickej siete zo všetkých dostupných meraní uvedených vo vstupnom súbore. Pre parsovanie XML dát bola použitá minimalistická efektívna knižnica dostupná na [XML–51]. Graf geodetickej siete sa vytvára iba raz pri započatí výpočtu. Určovanie komponent súvislosti grafu sa potom určuje pri každom výbere vzoru sústavy. Na základe informácie o skupine meraní, ktorá by mala byť odstránená sa vytvorí podgraf vzoru sústavy. Na tomto podgrafe sa potom určí počet komponent súvislosti pomocou algoritmu DFS. Pokiaľ graf obsahuje viac než jednu komponentu súvislosti a tá obsahuje neznámy bod, tak v tom prípade nieje možné použiť vybraný vzor sústavy a pokračuje sa výberom iného náhodného vzoru sústavy.

Rekurzívna funkcia prehľadávania grafu do hĺbky ma dve vstupné parametre a to počiatkový vrchol, z ktorého sa prehľadáva graf a pole navštívených vrcholov. Postupne sa prehľadáva spojový zoznam reprezentujúci hrany grafu. Navštívený vrchol, teda vrchol dostupný z počiatkového vrcholu je označený hodnotou true. Funkcia končí v momente navštívenia všetkých dostupných vrcholov.

```

1 void DFSUtil(int v, bool visited[])
2 {
3     visited[v] = true;
4     list<int>::iterator i;
5
6     for(i = m_adjList[v].begin(); i != m_adjList[v].end(); ++i)
7         if(!visited[*i])
8             DFSUtil(*i, visited);
9 }

```

Kód 13.1: Algoritmus prehľadávania do hĺbky

Metóda určí počet komponent slabej súvislosti grafu. Kde sa postupne označuje pole navštívených vrcholov grafu. Vrcholy sa označujú pomocou algoritmu prehľadávania grafu do hĺbky. V prípade, že vrchol nebol navštívený, tak sa zavolá funkcia prehľadávania z daného bodu. Pokiaľ sa pomocou funkcie prehľadávania do hĺbky

z daného bodu navštívia všetky vrcholy grafu, tak sa označia všetky vrcholy na hodnotu true a počet komponent grafu sa inkrementuje iba raz. To znamená, že graf má iba jednu komponentu slabej súvislosti. V prípade, že z počiatočného vrcholu nie sú dostupné všetky vrcholy grafu, tak sa pokračuje v prehľadávaní grafu z ďalšieho nedostupného vrcholu grafu a inkrementuje sa počet komponent súvislosti. Cyklus pokračuje kým nie sú označené navštívené všetky vrcholy grafu. Počet komponent slabej súvislosti je teda rovný počtu volania funkcie prehľadávania grafu do hĺbky.

```
1 int connectedComponents()
2 {
3     int countGraphs = 0;
4     bool *visited = new bool[m_countVertices];
5
6     for(int v = 0; v < m_countVertices; v++)
7         visited[v] = false;
8
9     for (int v=0; v < m_countVertices; v++)
10    {
11        if (visited[v] == false)
12        {
13            DFSUtil(v, visited);
14            countGraphs++;
15        }
16    }
17
18    return countGraphs;
19 }
```

Kód 13.2: Určenie počtu komponent slabej súvislosti

Vstupom do funkcie kontroly konektivity grafu je odkaz na množinu vylúčených meraní. Na začiatku sa vyhotoví inštancia objektu grafu a zotriedi sa množina vylúčených meraní. V cykle sa postupne iteruje od prvého merania po n-té meranie. Pokiaľ sa i-té meranie nachádza v množine vylúčených meraní tak sa hrana do grafu nevloží a posunie sa iterátor na ďalšie vylúčené meranie. V opačnom prípade sa vloží do grafu hrana reprezentujúca meranie medzi dvoma bodmi. Na záver sa zavolá metóda, ktorá určí počet komponent grafu.

```
1 int CheckNetworkConnectivity(vector<int> &drows)
2 {
3     Graph graph(m_points.size());
4
5     sort(drows.begin(), drows.end());
6
7     for( int i=0, j=0; i<m_observations.size(); i++)
8     {
9         if( j < drows.size() && i == drows[j] )
10        {
11            j++;
12        }
13        else
14        {
```

```
15         graph.addEdge(m_observations[i][0], m_observations[i][1]);
16     }
17 }
18
19 return graph.connectedComponents();
20 }
```

Kód 13.3: Metoda vytvorenia databázových schém

## 13.4 Perzistentné uloženie výsledkov do databáze

V priebehu výpočtu vyrovnania metódou LMS aplikácia kontinuálne zapisuje značné množstvo dát na zvolené blokové zariadenie. Spracovanie výsledkov uložených do jednoduchých textových súborov bolo veľmi náročne a neefektívne. Manipulácia s hodnotami pomocou programu *awk*, či *sed* bola prakticky nepoužiteľná a v mnohých prípadoch vyžadovala ďalšie programové prostriedky. Riešenie perzistentného uloženia dát a následnú jednoduchú manipuláciu ponúkajú databázové systémy. Pre túto prácu bola zvolená open-source relačná databáza PostgreSQL z dôvodu, flexibility konfigurácie výpočtových hardvérových zdrojov v závislosti na veľkosti dát, prakticky neobmedzenej kapacity databázových objektov a predchádzajúcich praktických skúseností. V nasledujúcich odstavcoch bude v skratke popísaná databáza PostgreSQL, jej konfigurácia a napojenie na aplikáciu LMS.

### 13.4.1 Databáza PostgreSQL

PostgreSQL je populárna a veľmi obľúbená open source databáza jak v sfére akademickej, tak v sfére komerčnej. PostgreSQL je multi-platformná databáza vyvíjaná v jazyku C a je možné ju skompilovať na akékoľvek zariadenie obsahujúce prekladač tohto jazyka. Podporuje MVCC (Multi-version concurrency control), cudzie kľúče, joiny, pohľady, trigger, uložené procedúry, table-partitioning a mnoho ďalších. Súčasťou je tiež fyzické a logické zálohovanie, (a)synchronná replikácia, obnova dát z bodu v čase (PITR), tablespace, možnosti logovania, transakčné logy, optimalizátor a planer SQL dotazov. V tejto práci je využitý iba zlomok jej možností. Databázu je možné nainštalovať z linuxových balíkov či kompiláciou zdrojových kódov stiahnutelných zo stránky [postgresql.org](http://postgresql.org).

### 13.4.2 Inštalácia a konfigurácia databáze na platforme Linux

Inštalácia databáze je vykonaná kompiláciou zdrojových kódov.

```
1 $ tar postgresql-9.3.4.tar.gz
2 $ cd postgresql-9.3.4
```



```
3 $ ./configure
4 $ make
5 $ su -
6 # make install
```

Kód 13.4: Inštalácia databáze

Implicitne je databáza nainštalovaná do adresára `/usr/local/pgsql/`. V ďalšom kroku je potrebné inicializovať dátový cluster a naštartovať databázový server. Príkazy sú spustené užívateľom `postgres`. V prípade, že v konkrétnom OS užívateľ `postgres` neexistuje, je vhodné ho vytvoriť. Databázový cluster by nemal vlastniť super-užívateľ `root`.

```
1 # su - postgres
2 $ mkdir data
3 $ /usr/local/pgsql/bin/initdb -D data
```

Kód 13.5: Inicializácia databázového clusteru

Pred štartom databáze je vhodné vykonať minimálnu konfiguráciu, ktorá spočíva v nastavení sieťového rozhrania, na ktorom bude databázový server prijímať kone-  
xie. Konfigurácia sa vykoná v súbore `postgresql.conf`, ktorý sa nachádza v adresári databázového clusteru po jeho inicializácii.

```
1 $ cd data
2 $ vim postgresql.conf
3 ...
4 listen_addresses = '*'
5 ...
```

Kód 13.6: Konfigurácia rozhraní

Nutné je tiež povolenie prístupových oprávnení na úrovni operačného systému. Pre jednoduchosť je povolený prístup pre užívateľa `lms` do databáze `lms` bez zadania hesla pre `localhost` a statickú IP adresu v lokálnej sieti. Povolenie sa vykoná pridaním nižšie uvedených záznamov do súboru `pg_hba.conf`

```
1 $ cd data
2 $ vim pg_hba.conf
3 ...
4 local lms lms trust
5 host lms lms 10.0.0.101/32 trust
6 ...
```

Kód 13.7: Povolenie prístupu

V tejto chvíli je možné spustiť databázový server.

```
1 $ /usr/local/pgsql/bin/pg_ctl -D data start
```

---

Kód 13.8: Spustenie databázového serveru

### 13.4.3 Založenie databáze a užívateľa

Aplikácia implicitne naväzuje spojenie s databázovým serverom cez užívateľa *lms* a pripája sa do databáze *lms*. Databázu a užívateľa je potrebné vyhotoviť pred spustením aplikácie.

```
1 # su - postgres
2 $ psql
3
4 postgres=# CREATE USER lms;
5 CREATE ROLE
6
7 postgres=# CREATE DATABASE lms OWNER lms;
```

Kód 13.9: Založenie databáze a užívateľa

### 13.4.4 Návrh databázových objektov

Databázu výpočtov tvorí šesť relácií, ktoré sú vo vzájomnom vzťahu 1:1 a 1:N. Každý výpočet má priradený jedinečný identifikátor *id* uložený v relácii *settings*. Relácia *settings* obsahuje informácie o konfigurácii výpočtu. Z každého priebehu výpočtu získame minimálny medián, priemerný medián a celkový čas priebehu výpočtu. Tieto hodnoty sú uložené v relácii *median*. Vyhodnotenie výsledkov si vyžaduje ukladanie niektorých ďalších nevyhnutných hodnôt a to odhad vyrovnaných neznámych parametrov a odhad vyrovnaných reziduí. Vo výsledku sú tieto hodnoty reprezentované ako vektory, ktoré sú uložené do dvoch databázových tabuliek *estimated\_residuals* a *estimated\_unknows*. Každý riadok obsahuje index a hodnotu vektora. Tieto tabuľky sú vo vzťahu 1:N s tabuľkou *settings* pretože ku každému výpočtu prísluší niekoľko výpočtov odhadovaných vektorov. Primárny kľúč obsiahnutý v tabuľkách zabezpečuje referenčnú integritu s reláciou *settings*. V momente, keď medián prestane konvergovať, získame množinu preváhaných riadkov. Táto množina je uložená v relácii *reweighted\_rows*. Vzťah *reweighted\_rows* relácie k relácii *settings* je 1:N. Tabuľka, kde sa ukladajú záznamy z priebehu každého výpočtu sa nazýva *median\_cmp* a slúži na podrobnú analýzu konverencie mediánu. V tabuľke je zaznamenaný každý krok výpočtu, maximálna a minimálna hodnota reziduí, absolútna hodnota súčtu reziduí, veľkosť mediánu z predošlého kroku výpočtu a veľkosť aktuálneho mediánu.

**Relácia *settings*:**

| Názov stĺpca            | Dátový typ   | Integritné obmedzenie |
|-------------------------|--------------|-----------------------|
| id                      | SERIAL       | PRIMARY KEY           |
| design_matrix_file      | VARCHAR(200) | •                     |
| measurement_vector_file | VARCHAR(200) | •                     |
| weight_matrix_file      | VARCHAR(200) | •                     |
| max_iter                | INTEGER      | •                     |
| random_sample           | INTEGER      | •                     |
| removed_lines           | INTEGER      | •                     |
| method                  | VARCHAR(50)  | •                     |
| extreme_weight          | FLOAT        | •                     |

**Relácia *median\_cmp*:**

| Názov stĺpca  | Dátový typ       | Integritné obmedzenie   |
|---------------|------------------|-------------------------|
| settings_id   | INTEGER          | REFERENCES settings(id) |
| cycle         | INTEGER          | NOT NULL                |
| step          | INTEGER          | NOT NULL                |
| median_before | DOUBLE PRECISION | NOT NULL                |
| median_after  | DOUBLE PRECISION | NOT NULL                |
| max_residual  | DOUBLE PRECISION | NOT NULL                |
| min_residual  | DOUBLE PRECISION | NOT NULL                |
| abs_residual  | DOUBLE PRECISION | NOT NULL                |

**Relácia *median*:**

| Názov stĺpca | Dátový typ       | Integritné obmedzenie   |
|--------------|------------------|-------------------------|
| settings_id  | INTEGER          | REFERENCES settings(id) |
| min_median   | DOUBLE PRECISION | NOT NULL                |
| avg_median   | DOUBLE PRECISION | NOT NULL                |
| cmp_time     | TIMESTAMP        | DEFAULT(now())          |

**Relácia *reweighted\_rows*:**

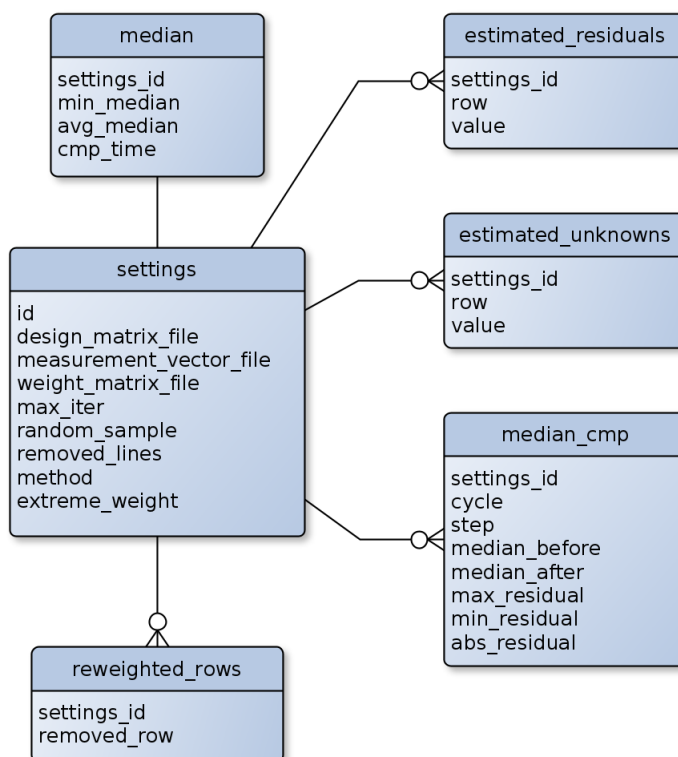
| Názov stĺpca | Dátový typ | Integritné obmedzenie   |
|--------------|------------|-------------------------|
| settings_id  | INTEGER    | REFERENCES settings(id) |
| removed_row  | INTEGER    | •                       |

**Relácia *estimated\_unknowns*:**

| Názov stĺpca | Dátový typ       | Integritné obmedzenie   |
|--------------|------------------|-------------------------|
| settings_id  | INTEGER          | REFERENCES settings(id) |
| row          | INTEGER          | NOT NULL                |
| value        | DOUBLE PRECISION | NOT NULL                |

Relácia *estimated\_residuals*:

| Názov stĺpca | Dátový typ       | Integritné obmedzenie   |
|--------------|------------------|-------------------------|
| settings_id  | INTEGER          | REFERENCES settings(id) |
| row          | INTEGER          | NOT NULL                |
| value        | DOUBLE PRECISION | NOT NULL                |



Obrázok 13.3: Databázové schéma

## 13.5 Implementácia komunikácie aplikácie LMS s databázou

V tejto sekcii sú popísané algoritmy zodpovedné za komunikáciu aplikácie LMS s databázou PostgreSQL. V triede sú volané funkcie z knižnice databázy *libpq-fe.h* vytvorenej v jazyku C. Rozhranie je definované v triede *DbResults*.

| Metóda                            | Popis   |
|-----------------------------------|---|
| <i>connect()</i>                  | Naviazanie spojenia s databázou                       |
| <i>createSchemas()</i>            | Vytvorenie objektov databáze                          |
| <i>dropSchemas()</i>              | Odstránenie všetkých objektov databáze                |
| <i>resetSchemas()</i>             | Reset objektov databáze                               |
| <i>closeConnection()</i>          | Zavretie konexie                                      |
| <i>insertSettings()</i>           | Vloženie hodnôt do tabuľky <i>settings</i>            |
| <i>insertMedian()</i>             | Vloženie hodnôt do tabuľky <i>median</i>              |
| <i>insertMedianCmp()</i>          | Vloženie hodnôt do tabuľky <i>median_cmp</i>          |
| <i>insertReweightedRows()</i>     | Vloženie hodnôt do tabuľky <i>reweighted_rows</i>     |
| <i>insertEstimatedUnknowns()</i>  | Vloženie hodnôt do tabuľky <i>estimated_unknowns</i>  |
| <i>insertEstimatedResiduals()</i> | Vloženie hodnôt do tabuľky <i>estimated_residuals</i> |

### 13.5.1 Naviazanie konexie

Naviazanie konexie aplikácie s databázou PostgreSQL sa vykonáva pomocou metódy `connect()`. Parametre funkcie tvoria tzv. *connection string*. Connection string obsahuje IP adresu, na ktorej beží databázový server, port, na ktorom databázový server počúva, názov databáze, meno a heslo užívateľa, ktorý sa do databáze prihlasuje. Vo funkcii sa zavolá funkcia z knižnice databáze PostgreSQL `PQsetdbLogin` s potrebnými údajmi a vráti pointer na konexiu databáze. V metóde sa na záver verifikuje konexia. V prípade, že sa konexiu nepodarilo naviazať na štandardný výstup, sa vypíše varovná správa.

```

1
2 int connect(const char* host, const char* port,
3             const char* db, const char* login, const char* pwd)
4 {
5     conn = PQsetdbLogin(host, port, NULL, NULL, db, login, pwd);
6
7     if(PQstatus(conn) != CONNECTION_OK)
8     {
9         cout << "Konexi do databaze: " << db
10            << "\nUzivatel: " << login
11            << "\nHost: " << host
12            << "\nPort: " << port
13            << "\nsa nepodarilo naviazat " << endl;
14         return -1;
15     }
16     else
17     {
18         cout << "=====\n";
19         cout << "    Spojenie do databaze OK " << endl;
20         cout << "=====\n";
21         cout << "Databaza:          " << PQdb(conn) << endl;
22         cout << "Uzivatel:          " << PQuser(conn) << endl;
23         cout << "Host:             " << PQhost(conn) << endl;
24         cout << "Port:            " << PQport(conn) << endl;

```

```

25         cout << "Verzia db serveru: " << PQserverVersion(conn)
26             << endl;
27         cout << "===== \n\n";
28
29         return 0;
30     }
31 }

```

Kód 13.10: Metoda naviazania spojenia s databázovým serverom

### 13.5.2 Vytvorenie a odstránenie objektov databáze

Všetky databázové objekty tzn. tabuľky, prípadne indexy sú vytvorené pomocou metódy `createSchemas`. Metóda vracia celočíselný dátový typ reprezentujúci úspešnosť vytvorenia objektov. Vytvorenie objektov sa vykonáva pomocou funkcie `PQexec()`. Funkcia má ako vstupné parametre ukazateľ na konexiu a string reprezentujúci SQL príkaz. Odstránenie objektov sa vykonáva rovnakým princípom zavolaním metódy `dropSchemas()`.

```

1
2 int createSchemas()
3 {
4     PGresult* res;
5
6     const char* settings_tab="CREATE TABLE settings("
7         " id SERIAL PRIMARY KEY, "
8         " design_matrix_file VARCHAR(200),"
9         " measurement_vector_file VARCHAR(200),"
10        " weight_matrix_file VARCHAR(200),"
11        " max_iter INTEGER,"
12        " random_sample INTEGER,"
13        " removed_lines INTEGER,"
14        " method VARCHAR(50),"
15        " extreme_weight FLOAT"
16        " );";
17        ...
18
19        PQexec(conn,settings_tab);
20        ...
21
22        return 0;
23    }

```

Kód 13.11: Metoda vytvorenia databázových schém

### 13.5.3 Zápis hodnôt do databázových objektov

K zápisu hodnôt do databázových tabuliek slúži niekoľko metód. Každý tabuľke prísluší jedna metóda `insert`. Obdobne ako pri vytváraní objektov databáze sa

volá funkcia `PQexec()`. Parametre funkcie sú pointer na konexiu do databáze a SQL príkaz.

```
1
2 int insertSettings(LmsSettings<>& set)
3 {
4     vector<string> setvalues;
5     vector<string>::iterator it;
6
7     setvalues.push_back("`"+string(set.get_file_A())+"`");
8     setvalues.push_back("`"+string(set.get_file_l())+"`");
9     setvalues.push_back("`"+string(set.get_file_p())+"`");
10    setvalues.push_back(toString(set.get_maxiter()));
11    setvalues.push_back(toString(set.get_sample_proc()));
12    setvalues.push_back(toString(set.get_remove_rows()));
13    setvalues.push_back("`"+string(set.get_method())+"`");
14    setvalues.push_back(toString(set.get_extreme_weight()));
15
16    string insert_s="INSERT INTO settings("
17                "design_matrix_file,"
18                "measurement_vector_file,"
19                "weight_matrix_file,"
20                "max_iter,"
21                "random_sample,"
22                "removed_lines,"
23                "method,"
24                "extreme_weight"
25                ") "
26                "VALUES(";
27
28    for(it=setvalues.begin(); it!=setvalues.end()-1;it++)
29    {
30        insert_s += *it+", ";
31    }
32
33    insert_s += *it+");";
34
35    const char* insert=insert_s.c_str();
36
37    PQexec(conn,insert);
38
39    const char * cmp_id_query =
40    "SELECT currval('settings_id_seq '::regclass);";
41
42    PGresult* res;
43
44    // Vratim poslednu hodnotu sekvencie v aktualnej session
45    res = PQexec(conn,cmp_id_query);
46
47    cmp_id = atoi(PQgetvalue(res,0,0));
48
49    return 0;
50 }
```

Kód 13.12: Metoda zápisu do relácia settings

## 13.6 Použitie aplikácie LMS

Aplikácia pre výpočet LMS je konzolová aplikácia primárne určená pre operačný systém Linux. Kompilácia aplikácie vyžaduje prekladač jazyka C++ a program *make*. Program používa knižnice databáze Postgres a knižnice pre prácu s maticami GNU Gama MatVec. Externé knižnice nieje nutné inštalovať, pretože sú priložené k zdrojovým kódom aplikácie. V prípade inštalácie aplikácie do systému sú vyžadované práva superužívateľa root. Spustenie aplikácie sa vykonáva binárnym súborom *lms*. Pred spustením výpočtu je potrebné upraviť dve konfiguračné súbory, ktoré slúžia na konfiguráciu pripojenia aplikácie do databáze a na konfiguráciu samotného výpočtu.

### 13.6.1 Konfiguračné súbory aplikácie LMS

Pri vykonávaní výpočtu je možné použiť dve konfiguračné súbory. Konfiguračný súbor pre nastavenie konexie do databáze je nepovinný a v prípade, že nieje použitý, aplikácia použije východzie nastavenie konexie. Na rozdiel od súboru konfigurácie databáze je súbor s nastavením výpočtu je pre aplikáciu povinný a sú v ňom nastavené hodnoty pre samotný výpočet LMS.

#### Možné hodnoty parametrov konfiguračného súboru výpočtu

DESIGN\_MATRIX\_FILE - cesta k súboru s maticou plánu

MEASUREMENT\_VECTOR\_FILE - cesta k súboru s vektorom redukovaných meraní

WEIGHT\_MATRIX - cesta k súboru s kovariančnou maticou meraní

MAX\_ITER - maximálny počet iterácií možnosti (1-n)

RANDOM\_SAMPLE - veľkosť vzoru matice plánu (%) (1-99)

REMOVED\_LINES - počet preváňovaných riadkov v každom kroku výpočtu

( $\mathbf{A} \in m \times n$   $1 - \text{int}(m/2) + 1$ )

METHOD - metóda preváňovania možnosti (ExtremeWeight, MultipleWeight, IncreaseWeight)

#### Príklad konfiguračného súboru výpočtu

```
DESIGN_MATRIX_FILE=../cesta/vystup_matvec_divided/jindriska.mat
```

```
MEASUREMENT_VECTOR_FILE=../cesta/vystup_matvec_divided/jindriska.vec
```

```
WEIGHT_MATRIX=../cesta/vystup_matvec_divided/jindriska.cov
```

```
MAX_ITER=100
```

```
RANDOM_SAMPLE=99
```

```
REMOVED_LINES=20
```



METHOD=ExtremeWeight

### Konfiguračné parametre pripojenia aplikácie k databáze

HOST - host databáze. Implicitná hodnota 127.0.0.1

PORT - port databáze. Implicitná hodnota 5432

DATABASE - názov databáze do ktorej sa budú ukládať výsledky. Implicitná hodnota lms

USER - užívateľ (s právom zápisu do databáze). Implicitná hodnota lms

PASSWORD - heslo užívateľa, ktorý sa pripojuje do databáze. Implicitná hodnota lms

### Príklad konfiguračného súboru pripojenia do databáze

HOST=192.168.122.143

PORT=5432

DATABASE=lms

USER=lms

PASSWORD=lms

## 13.6.2 Inštalácia a spustenie aplikácie

V tejto sekcii je popísaná inštalácia a spustenie aplikácie.

```
1 $ wget gyori.cz/lms.tar.gz
2 $ tar xvzf lms.tar.gz
3 $ cd lms
4 $ ./configure
5 $ make
6 # make install
```

Kód 13.13: Hromadné vyrovnanie sietí

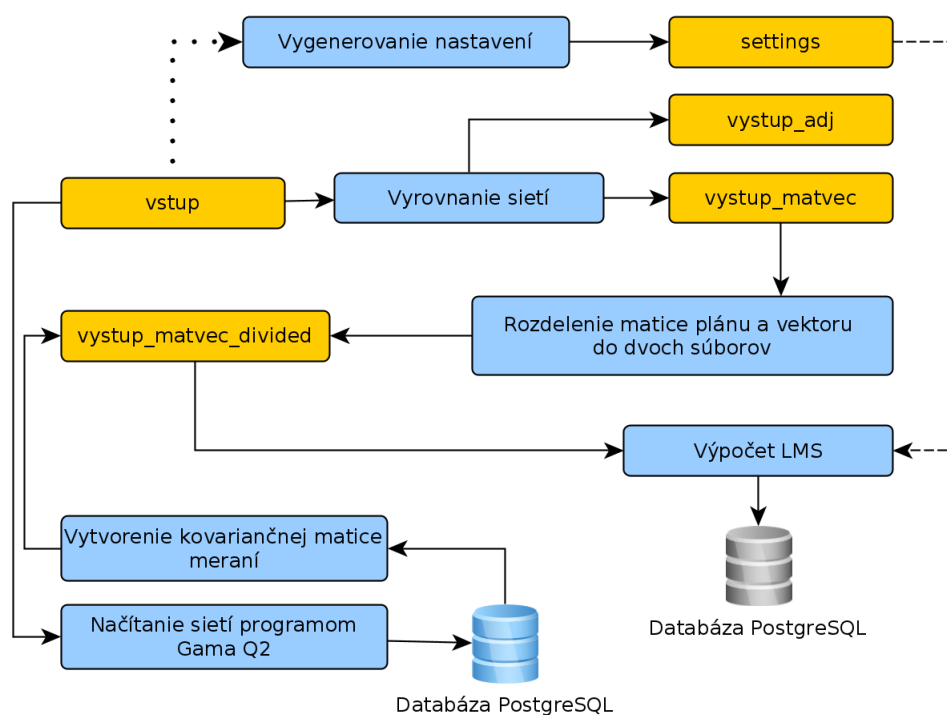
Spustenie výpočtu sa vykoná zavolaním binárneho súboru lms

```
1 $ lms LMS_SETTINGS_FILE [options]
```

Kód 13.14: Hromadné vyrovnanie sietí

## 13.7 Technológia hromadného spracovania dát

Zhodnotenie metódy LMS vyžaduje spracovanie pomerne veľkého množstva dát. Spracovanie zahŕňa vyrovnanie sietí klasickou metódou MNČ a získanie matice plánu, vektoru redukovaných meraní a kovariančnej matice meraní zo vstupných \*.gkf súborov obsahujúcich meranie v geodetických sieťach. Matica plánu a každý z vektorov by sa mal nachádzať v separátnom súbore. V závislosti na umiestnení súborov na súborovom systéme sa vygeneruje konfiguračný súbor, ktorý vyžaduje aplikácia LMS. V ďalších odstavcoch bude popísaný technologicky postup od získania matice a vektorov až po vyrovnanie dát metódou LMS.



Obrázok 13.4: Automatizácia výpočtu sietí

- *vystup\_adj* - adresár s výsledkami vyrovnania
- *vystup\_matvec* - adresár súborov matíc a vektorov, kde matica a vektor sú obsiahnuté v jednom súbore
- *vystup\_matvec\_divided* - adresár súborov matíc a vektorov, pričom matica a vektor sú oddelené

### 13.7.1 Vyrovnávanie sietí

Vyrovnávanie testovacích geodetických sietí je vykonané v programe *gama-local*. Výsledky vyrovnania slúžia pre porovnanie výsledkov klasickej metódy MNČ s metódou LMS. Primárnym výsledkom je však získanie matice plánu a vektoru redukovaných meraní zostavených taktiež programom *gama-local* z geodetických meraní v teréne. Program *gama-local* však implicitne neumožňuje potrebnú maticu a vektor vypísať na štandardný výstup, avšak v priebehu vyrovnania ich zostavuje a používa. Zdrojové kódy programu *gama-local* museli byť upravené a to tak, že matica a vektor bola presmerovaná na štandardný chybový výstup, aby dáta bolo možné odlíšiť od výsledku vyrovnania. Upravené zdrojové kódy boli následne skompilované do binárnych súborov. Tieto binárne súbory mohli byť použité v skripte uvedenom nižšie. Skript vyhľadáva v vstupnom adresári všetky súbory s príponou *\*.gkf* a jeden po druhom ich postupne vyrovnáva. Výsledok vyrovnania sú XML súbory uložené v adresári *vystup\_adj*. Matice a vektory sú uložené v adresári *vystup\_matvec*

```

1  #!/bin/bash
2
3  GAMABIN=../gama-1.11/bin/gama-local
4  INDATADIR=../data/site/vstup
5  OUTMATVECDIR=../data/site/vystup_matvec
6  OUTADJDIR=../data/site/vystup_adj
7
8  for i in $( find $INDATADIR -name "*.gkf" -printf "%f\n" )
9  do
10     $GAMABIN $INDATADIR/$i~1> $OUTADJDIR/$i~2>$OUTMATVECDIR/${i%.gkf}.mv
11     $GAMABIN $INDATADIR/$i --text $OUTADJDIR/${i%.gkf}.txt 2> /dev/null
12 done

```

Kód 13.15: Hromadné vyrovnávanie sietí

### 13.7.2 Separácia vstupných dát

V prípade, že vstupné dáta obsahujú odľahlé merania, program *gama-local* počíta vyrovnávanie v  $n$  krokoch, pričom v každom kroku vylúči jedno odľahlé meranie a z toho vyplýva, že v každom kroku je matica plánu a vektor redukovaných meraní odlišný o konkrétne vylúčené merania. Súbor teda obsahuje  $n$  matíc a vektorov. Každú maticu a vektor oddeľuje značka *Mat-A*, *Vec-v* a *End*. Značky sú doplnené v zdrojovom kóde programu *gama-local*. Vstup do programu LMS by však mala byť úplne prvá matica a vektor bez vylúčených odľahlých meraní v dvoch separovaných súboroch. Následujúci skript vezme všetky súbory s maticami a vektormi z adresára *vystup\_matvec* a rozdelí tieto údaje do dvoch nezávislých súborov s príponami *\*.mat*

a \*.vec, ktoré sa nachádzajú v adresári *vystup\_matvec\_divided*. To sa vykonáva pomocou nižšie uvedeného skriptu s využitím linuxového programu *sed*, ktorý vezme údaje od prvej značky *Mat-A* po prvú značku *Vec-v*, čomu prislúšia dáta matice plánu a presmeruje ich do súboru s príponou \*.mat. Súbor obsahuje značky *Mat-A* a *Vec-v*, ktoré už nie sú potrebné, preto sa použije príkaz *sed* s prepínačom *-i*, čo znamená, že sa zmena vykoná priamo na vstupnom súbore. Obdobne sa z vstupného súboru získa vektor a to tak, že program *sed* vyberie dáta od prvej značky *Vec-v* po prvú značku *End*. Obdobne ako u matice plánu sa potom tieto značky z výsledného súboru odstránia.

```
1 #!/bin/bash
2
3 INPUTDIR=../data/site/vystup_matvec
4 OUTDIR=../data/site/vystup_matvec_divided
5
6 for i in $( find $INPUTDIR -name "*.mv" -printf "%f\n" )
7 do
8
9     sed -n '/Mat-A/,/Vec-v/p;/Vec-v/q' $INPUTDIR/$i > $OUTDIR/${i%.mv}.mat
10
11     sed -i 's/Mat-A//g' $OUTDIR/${i%.mv}.mat
12     sed -i 's/Vec-v//g' $OUTDIR/${i%.mv}.mat
13
14     sed -n '/Vec-v/,/End/p;/End/q' $INPUTDIR/$i > $OUTDIR/${i%.mv}.vec
15
16     sed -i 's/Vec-v//g' $OUTDIR/${i%.mv}.vec
17     sed -i 's/End//g' $OUTDIR/${i%.mv}.vec
18 done
```

Kód 13.16: Separácia vstupných dát

### 13.7.3 Zostavenie konfiguračných súborov

Program pre výpočet LMS používa súbor s nastavením, v ktorom sú uvedené umiestnenia matíc a vektorov na súborovom systéme, počet iterácií, rozmer vzoru sústavy a pod.. Tento súbor sa zastavuje v závislosti na vstupných \*.gkf súboroch. Skript predpokladá, že pre každý vstupný \*.gkf súbor s nejakým menom existuje súbor matice plánu, vektoru redukovaných meraní a kovariančnej matice meraní v adresári *vystup\_matvec\_divided*. Absolútne cesty k týmto súborom sú automaticky dosadené do premenných *DESIGN\_MATRIX\_FILE* atd.. Ostatné premenné s konfiguráciami LMS je možné nastaviť ručne. Spustením skriptu sa zostavia súbory s nastavením. Pomenovanie súborov je totožné s názvami vstupných súborov. Súbory sú uložené do adresára *settings*.

```

1  #!/bin/bash
2
3  INPUTDIR=../data/site/vstup
4  RELPATH=../data/site/vystup_matvec_divided
5  OUTDIR=../settings
6
7  DESIGN_MATRIX_FILE=../data/tstd.mtx
8  MEASUREMENT_VECTOR_FILE=../data/tstm.mtx
9  WEIGHT_MATRIX=../data/tstw.mtx
10 MAX_ITER=100
11 RANDOM_SAMPLE=99
12 REMOVED_LINES=20
13 METHOD=ExtremeWeight
14
15
16 for i in $( find $INPUTDIR -name "*.gkf" -printf "%f\n" )
17 do
18
19     echo DESIGN_MATRIX_FILE=$RELPATH/${i%.gkf}.mat          \
20     > $OUTDIR/${i%.gkf}.set
21     echo MEASUREMENT_VECTOR_FILE=$RELPATH/${i%.gkf}.vec    \
22     >> $OUTDIR/${i%.gkf}.set
23     echo WEIGHT_MATRIX=$RELPATH/${i%.gkf}.cov              \
24     >> $OUTDIR/${i%.gkf}.set
25     echo MAX_ITER=$MAX_ITER                                 \
26     >> $OUTDIR/${i%.gkf}.set
27     echo RANDOM_SAMPLE=$RANDOM_SAMPLE                       \
28     >> $OUTDIR/${i%.gkf}.set
29     echo REMOVED_LINES=$REMOVED_LINES                      \
30     >> $OUTDIR/${i%.gkf}.set
31     echo METHOD=$METHOD                                     \
32     >> $OUTDIR/${i%.gkf}.set
33 done

```

Kód 13.17: Vygenerovanie nastavení

### 13.7.4 Vytvorenie kovariančnej matice meraní

Kovariančná matica meraní je štvorcová diagonálna matica, preto je možné s výhodou ukladať len jej diagonálu v podobe vektoru. Pre vytvorenie súborov s kovariančnými maticami meraní sa použije databáza PostgreSQL. Do databáze sú načítané všetky vstupné *\*.gkf* súbory pomocou programu *Gama Q2*. Pomocou príkazu SQL uvedeného na riadku 7 kódu 13.18 sa do dočasného súboru vypíšu dva stĺpce a to stĺpec s hodnotami matice a ku každej hodnote je uvedený názov súboru, ku ktorému hodnota prísluší. To znamená, že v dočasnom súbore sú uložené všetky kovariančné matice. Matice je nutné uložiť do separátnych súborov. Každý súbor však musí najprv obsahovať veľkosť matice. Pre určenie veľkosti sa použije program *wc* s prepínačom *-l*, ktorý spočíta všetky riadky presmerované z programu *grep*. Program *grep* obmedzí množinu riadkov v závislosti na názve súboru. Následne sa

do súboru zapíšu hodnoty pre konkrétny súbor. Prvý stĺpec, ktorý obsahuje hodnoty matice, je vybraný pomocou programu *awk*. Súbor s maticami sú uložené do adresára *vystup\_matvec\_divided*.

```

1  !/bin/bash
2
3  INPUTDIR=../data/site/vstup
4  OUTDIR=../data/site/vystup_matvec_divided
5
6  psql -d site -U~gabriel -t -c
7  "SELECT val,conf_name FROM gnu_gama_local_covmat a
8  JOIN gnu_gama_local_configurations b
9  ON a.conf_id=b.conf_id;" > /tmp/covmat
10
11 for i in $( find $INPUTDIR -name "*.gkf" -printf "%f\n" )
12 do
13     cat /tmp/covmat | grep ${i%.gkf} | wc -l > $OUTDIR/${i%.gkf}.cov
14     echo -e "\n" >> $OUTDIR/${i%.gkf}.cov
15     cat /tmp/covmat | grep ${i%.gkf} | awk '{print $1}' \
16     >> $OUTDIR/${i%.gkf}.cov
17 done

```

Kód 13.18: Vytvorenie kovariančnej matice meraní

### 13.7.5 Výpočet LMS

Výpočet LMS sa vykoná pomocou binárneho súboru *lms* pre každý súbor s nastavením uloženým v adresári *settings* presne sto krát. Výsledky sa kontinuálne ukladajú do databáze.

```

1  #!/bin/bash
2
3  SETDIR=../settings
4  LMSBIN=../lms-debug/lms
5
6  for j in {1..100}
7  do
8     for i in $( find $SETDIR -name "*.set" -printf "%f\n" )
9     do
10         $LMSBIN $SETDIR/$i
11     done
12 done

```

Kód 13.19: Vyrovnávanie dát aplikáciou LMS

## 13.8 Posúdenie kvality algoritmov

Skôr než boli algoritmy LMS použité na výpočet geodetických sietí, bolo vhodné vykonať overenie algoritmov na vzorových dátach. Dáta sú dostupne na stránkach <http://math.nist.gov/MatrixMarket/>, kde je možné zvoliť si typ matice formát veľkosť a pod.. Dáta boli vybrané v závislosti na testovanom algoritme.

### 13.8.1 Overenie metódy združených gradientov

Základom výpočtu sústavy lineárnych rovníc je metóda združených gradientov. Do projektu bolo naprogramovaných niekoľko alternatív, aby bolo možné relatívne posúdiť rýchlosť a presnosť výpočtu. Práca obsahuje metódy CG, CG s prepodmenením (PCG), CG bez explicitného vytvorenia normálnych rovníc a CG pracujúci so vzorom matice. Konštruktor všetkých tried združených gradientov má na vstupe ukazateľ na riedku maticu. Štruktúra riedkej matice je popísaná v kapitole 6.2. V triede je možné pomocou prístupových metód k privátnym členom nastaviť maximálny počet iterácií a presnosť výpočtu. Implicitná hodnota presnosti je  $10^{-11}$  a maximálny počet iterácií je rovný veľkosti matice. Metóda *solve()* má na vstupe konštantný iterátor na vektor meraní a iterátor na vektor neznámych. Metóda je navrhnutá všeobecne, to znamená, že v metóde je možné použiť výpočet so štandardným kontajnerom *vec* alebo napríklad vektor z knižnice *matvec*<sup>1</sup>.

Pre testovacie účely boli zvolené symetrické, pozitívne definitné matice s riedkou štruktúrou. Pretože sú matice symetrické, je logicky uložený len horný trojuholník matice. Aby trieda CG obdržala maticu s korektne naplneným objektom, museli sa pred výpočtom vykonať drobné úpravy formátu. Nižšie uvedená tabuľka ukazuje čas a počet iterácií výpočtu CG a CG s prepodmenením. Pravá strana bola zvolená ako vektor plný jednotiek. Po výpočte neznámych a spätnom vynásobení maticou plánu by sme mali zas teoreticky dostať pravú stranu s hodnotami blízkyimi jednej.

V tabuľke 13.1 hodnota *n* značí rozmer matice a *nz* je počet nenulových prvkov. Čas je uvedený v sekundách. Z tabuľky môžeme pozorovať, že predpomínenie nezvyší čas výpočtu vo všetkých prípadoch, preto je potrebné premyslieť, kde je vhodné ho použiť a kde nie. Predpomínenie však môže značne zvýšiť numerickú presnosť výpočtu.

### 13.8.2 Overenie algoritmov LMS

K overeniu algoritmov boli vybrané dáta obsahujúce maticu plánu a vektor pravej strany a sú priamo určené pre overenie MNČ. Matice sú obdĺžnikové, nesingu-

<sup>1</sup>*matvec* je knižnica výpočtov s maticami a vektormi vytvorená pre projekt GNU Gama

| Matica       | Veľkosť |        | CG    |          | PCG   |          | CG-PCG |          |
|--------------|---------|--------|-------|----------|-------|----------|--------|----------|
|              | n       | nz     | čas   | iterácie | čas   | iterácie | čas    | iterácie |
| bcsstk01.mtx | 48      | 400    | 0     | 48       | 0     | 12       | 0      | 36       |
| bcsstk15.mtx | 3948    | 117816 | 1.41  | 3948     | 0.28  | 213      | 1.13   | 3735     |
| bcsstm20.mtx | 485     | 485    | 0     | 288      | 0     | 1        | 0      | 287      |
| bcsstk07.mtx | 420     | 7860   | 0.01  | 420      | 0     | 80       | 0.01   | 340      |
| bcsstk03.mtx | 112     | 640    | 0     | 112      | 0     | 13       | 0      | 99       |
| bcsstk04.mtx | 132     | 3648   | 0     | 132      | 0     | 28       | 0      | 104      |
| bcsstk08.mtx | 1074    | 12960  | 0.06  | 1074     | 0.01  | 18       | 0.05   | 1056     |
| bcsstk05.mtx | 153     | 2423   | 0     | 153      | 0     | 30       | 0      | 123      |
| bcsstk06.mtx | 420     | 7860   | 0.01  | 420      | 0     | 80       | 0.01   | 340      |
| bcsstk11.mtx | 1473    | 34241  | 0.13  | 1473     | 0.35  | 1302     | -0.22  | 171      |
| bcsstm19.mtx | 817     | 817    | 0.01  | 478      | 0     | 1        | 0.01   | 477      |
| bcsstk16.mtx | 4884    | 290378 | 0.44  | 466      | 0.11  | 14       | 0.33   | 452      |
| bcsstk02.mtx | 66      | 4356   | 0     | 44       | 0     | 1        | 0      | 43       |
| bcsstk17.mtx | 10974   | 428650 | 15.94 | 10974    | 46.05 | 10974    | -30.11 | 0        |
| bcsstk10.mtx | 1086    | 22070  | 0.07  | 1086     | 0.17  | 1086     | -0.1   | 0        |
| bcsstk12.mtx | 1473    | 34241  | 0.13  | 1473     | 0.34  | 1302     | -0.21  | 171      |
| bcsstk14.mtx | 1806    | 63454  | 0.3   | 1806     | 0.21  | 348      | 0.09   | 1458     |
| bcsstk13.mtx | 2003    | 83883  | 0.43  | 2003     | 1.62  | 2003     | -1.19  | 0        |
| bcsstk18.mtx | 11948   | 149090 | 8.03  | 11948    | 0.58  | 307      | 7.45   | 11641    |

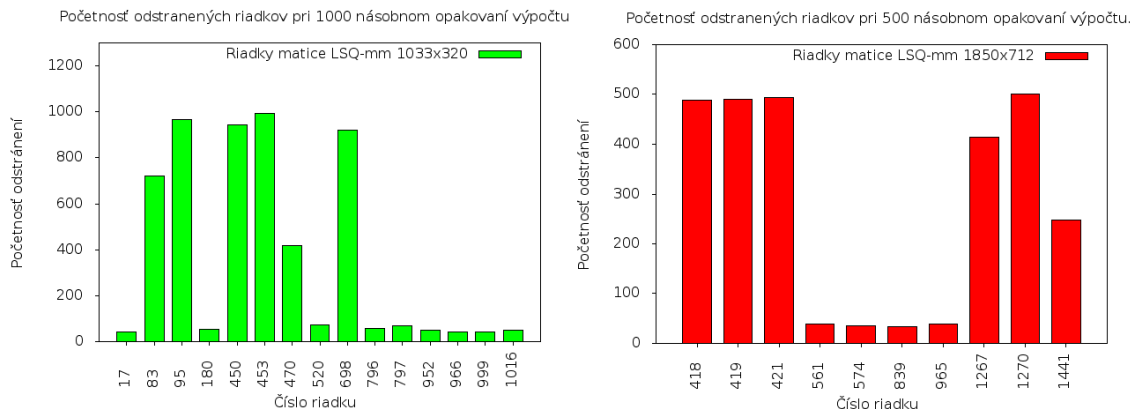
Tabuľka 13.1: Porovnanie rýchlosti rôznych typov algoritmov združených gradientov



lárne, uložené v riedkom formáte matrix marketu *\*.mm*. Predpokladá sa, že sústava je homogenizovaná. V prípade nehomogenizovanej sústavy je nutné vygenerovať váhový vektor. Pokiaľ nieje povedané ináč, konvergencia je zabezpečená preváhou riadkov extrémnymi váhami, ktoré sú určené ako recipročné hodnoty normy matice plánu. Na základe vykonaných testov bude možné odhadnúť konfigurácie pre výpočty geodetických sietí. Overenie zahŕňa test početnosti odstranených riadkov, konvergencie mediánu, veľkosť minimálneho mediánu v závislosti na vybranej stratégii váhovania.

### Test početnosti odstranených riadkov.

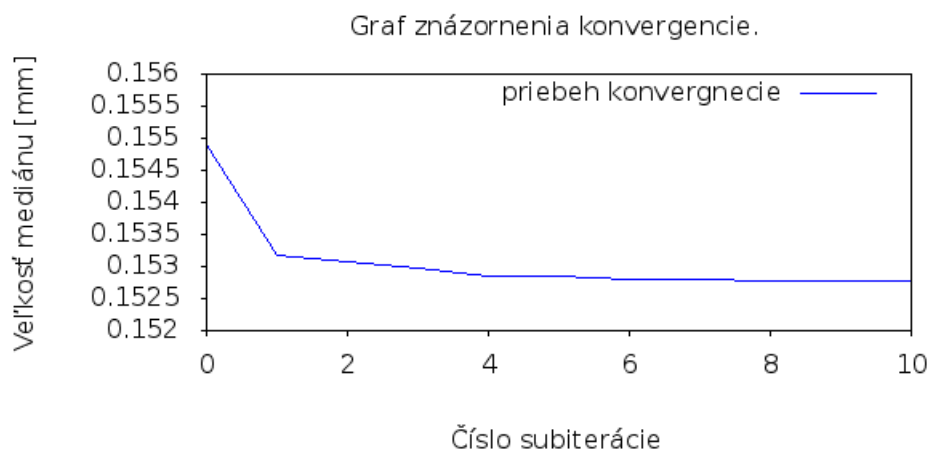
Test spočíva v určení indexov riadkov, ktoré sú odstránené resp. preváhouvané extrémnou váhou počas výpočtu v závislosti na konvergencii mediánu. Po jednom spustení výpočtu je určená jedna množina riadkov, ktorej preváhouvaním bol získaný minimálny medián v danom výpočte. V jednom výpočte je 100 krát náhodne vybraná množina dát, ktorá obsahuje 99 percent observácií. Z tejto množiny je v každom kroku preváhouvaná množina 5-tich riadkov až do chvíle, kým medián prestane konvergovať. Skutočný počet odľahlých meraní je rovný veľkosti preváhouvanej množiny. V ďalšom kroku je vygenerovaná nová pseudonáhodná množina dát z následným preváhouvaním a konvergenciou mediánu. V prípade, že minimálny medián je menší než medián z predošlej konvergencie, vezme sa množina preváhouvaných riadkov za množinu, ktorej odstránením dostaneme minimálny medián. Výpočet bol vykonaný 1000 krát v prípade matice s 1033 observáciami a 500 krát v prípade matice s 1850 observáciami. Z každého výpočtu je získaná množina riadkov, ktorých odstránením získame minimálny medián. Z dát je vyhotovený histogram, ktorý ukazuje koľko krát bol daný riadok odstránený, aby bol dosiahnutý optimálny výsledok.



Obrázok 13.5: Graf matice 1033x320 vľavo a 1850x712 vpravo

### Test konvergenzie mediánu

Na grafe 13.6 je znázornený priebeh konvergenzie mediánu. V každom kroku je vygenerovaná pseudonáhodná množina dát. Potom nasleduje postupne preváhovanie riadkov v závislosti na veľkosti štvorcov reziduí. Po každom preváhaní sa vykoná výpočet štvorcov reziduí a určí sa medián. Cyklus sa opakuje až do doby, kým medián prestane konvergovať.



Obrázok 13.6: Graf znázornenia konvergenzie mediánu pre jeden náhodný vzor

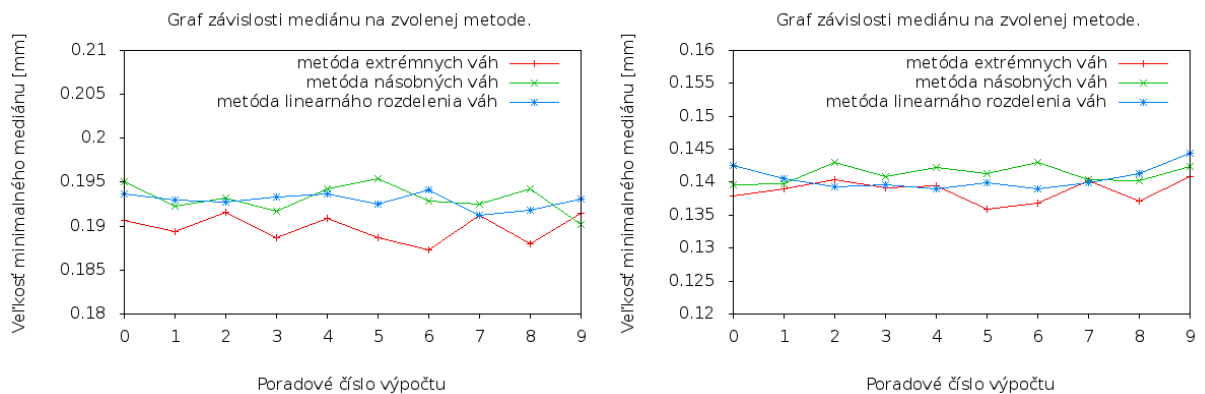
### Závislosť hodnoty minimálneho mediánu na vybranej stratégii váhovania

Výsledkom metódy LMS je primárne množina observácií podozrivých z odľahlosti. Observácie sú reprezentované číslami riadkov v matici plánu. Výsledok je možné určiť s použitím rôznych metód preváhovania týchto riadkov. Množina odľahlých meraní je závislá na veľkosti mediánu. Snahou je získať čo najmenšie hodnoty mediánu a zvýšiť rýchlosť konvergenzie. Stratégie váhovania sú podrobne popísané v kapitole 13.2.2. V tejto sekcii je popísaný vplyv stratégie váhovania na hodnotu minimalného mediánu. Konfigurácia výpočtu bola rovnaká pre každú zo stratégií. Veľkosť náhodného vzoru matice plánu bola pevne určená na 99%. Maximálny počet výberu rôznych náhodných vzorov bol zvolený na 1000, čo znamená, že pokiaľ medián prestane konvergovať, tak sa výpočet zastaví a vyberie sa nový náhodný vzor. Predošlý test ale naznačuje, že medián prestane konvergovať už po omnoho nižšom počte opakovaní. Konvergenzia mediánu ale môže byť odlišná v závislosti na zvolenej stratégii váhovania, preto bol počet iterácií zvolený na pomerne extrémnu hodnotu. Počet preváhaných riadkov matice, teda odhad počtu kontaminovaných meraní, je nastavený na hodnotu 20. Výsledne grafy 13.7 a hodnoty uvedené v tabuľkách 13.2 a 13.3 ukazujú, že pre tak malú množinu riadkov, ktoré majú byť preváhané v tomto

| Extrémne váhy |            | Násobne váhy |            | Rastuce váhy |            |
|---------------|------------|--------------|------------|--------------|------------|
| Min medián    | Avg medián | Min medián   | Avg medián | Min medián   | Avg medián |
| 0.137896      | 0.152276   | 0.139541     | 0.153119   | 0.142528     | 0.152033   |
| 0.139003      | 0.151775   | 0.139693     | 0.153107   | 0.140539     | 0.150784   |
| 0.140354      | 0.152601   | 0.142942     | 0.153154   | 0.139364     | 0.150801   |
| 0.139185      | 0.151678   | 0.140888     | 0.152748   | 0.139602     | 0.151327   |
| 0.139424      | 0.152352   | 0.142245     | 0.152684   | 0.138971     | 0.152558   |
| 0.13585       | 0.151416   | 0.141377     | 0.153066   | 0.139918     | 0.150685   |
| 0.136763      | 0.152235   | 0.142965     | 0.153011   | 0.139        | 0.151057   |
| 0.140256      | 0.152196   | 0.140333     | 0.150911   | 0.139891     | 0.151457   |
| 0.13715       | 0.151992   | 0.140295     | 0.151295   | 0.141363     | 0.153381   |
| 0.140799      | 0.152219   | 0.142463     | 0.152144   | 0.144325     | 0.153551   |

Tabuľka 13.2: Závislosť hodnoty minimálneho a mediánu na vybranej stratégii váhovania. Matica 1033x320

konkrétnom prípade, približne 2 ‰ celkového počtu meraní nemá výber stratégie takmer žiaden vplyv na veľkosť mediánu. To však nemusí platiť pre omnoho väčšie množiny.



Obrázok 13.7: Graf matice 1033x320 vľavo a 1850x712 vpravo

### Závislosť hodnoty minimalného mediánu na veľkosti množiny preváhovaných riadkov

Nastavenie metódy LMS umožňuje regulovať veľkosť množiny výsledných riadkov resp. meraní, ktoré by mali byť z výpočtu odstránené, pretože môžu obsahovať hrubé chyby. Výsledkom je teda vždy pevná množina meraní, ktorá môže a nemusí obsahovať merania s hrubými chybami. Z podstaty úlohy ale vyplýva, že pri odstránení

| Extrémne váhy |            | Násobne váhy |            | Rastuce váhy |            |
|---------------|------------|--------------|------------|--------------|------------|
| Min medián    | Avg medián | Min medián   | Avg medián | Min medián   | Avg medián |
| 0.190708      | 0.202505   | 0.195016     | 0.202773   | 0.193682     | 0.202953   |
| 0.189416      | 0.202426   | 0.192326     | 0.20235    | 0.192981     | 0.202465   |
| 0.191542      | 0.202498   | 0.193192     | 0.202569   | 0.192755     | 0.202673   |
| 0.188646      | 0.202463   | 0.191644     | 0.202248   | 0.193288     | 0.202392   |
| 0.190859      | 0.202239   | 0.194255     | 0.202707   | 0.193642     | 0.202217   |
| 0.188641      | 0.202289   | 0.19544      | 0.203032   | 0.192454     | 0.202083   |
| 0.187249      | 0.202465   | 0.192898     | 0.202571   | 0.194188     | 0.202569   |
| 0.191194      | 0.202401   | 0.192565     | 0.202466   | 0.191184     | 0.202379   |
| 0.187978      | 0.202446   | 0.194219     | 0.202473   | 0.191864     | 0.202659   |
| 0.191518      | 0.202412   | 0.190193     | 0.202615   | 0.19312      | 0.20294    |

Tabuľka 13.3: Závislosť hodnoty minimálneho a mediánu na vybranej stratégii váhovania. Matica 1850x712

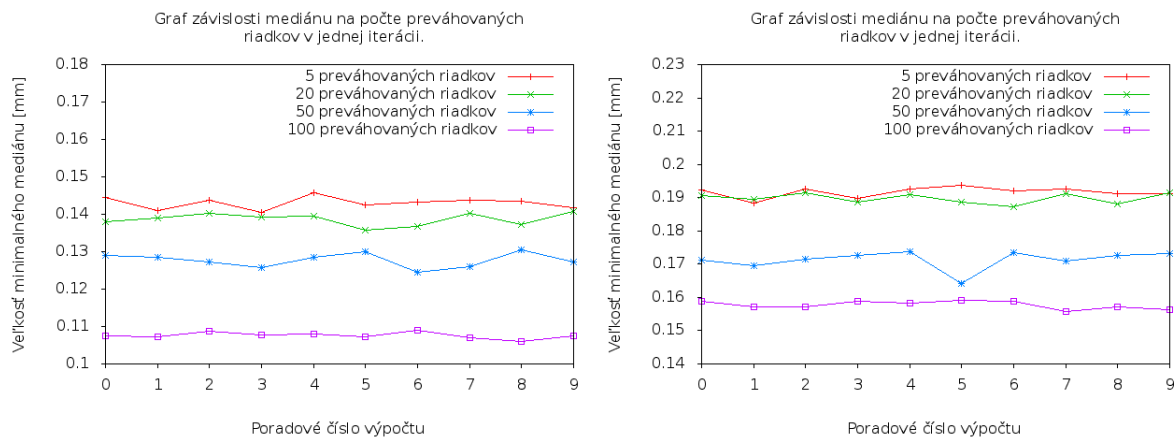
tejto množiny riadkov z výpočtu bude hodnota mediánu minimálna. V tomto prípade bola zvolená stratégia váhovania extrémnymi váhami. Testy ukazujú, že veľkosť minimalného mediánu je priamo závislá na veľkosti množiny odstránených meraní. To bol ale pomerne očakávaný výsledok, pretože čím viac meraní s väčšími štvorcami reziduí z množiny odstránime, tým sa hodnota mediánu zlepši. Čím väčšie sú rozdiely veľkosti medzi jednotlivými štvorcami reziduí tým je závislosť na počte odstránených riadkov väčšia. V ideálnom prípade, keby veľkosti štvorcov reziduí boli približne rovnaké a odľahlé merania by mali na druhu stranu štvorce reziduí omnoho vyššie, tak by sa pri odstránení odľahlých meraní hodnota mediánu zásadne nemenila. Merania sú ale v prípade geodetických sietí na sebe vzájomne závislé, takže porucha jedného merania ovplyvní štvorce reziduí meraní ostatných.

### Závislosť hodnoty minimálneho mediánu na počte opakovaní pseudonáhodného výberu

V každom kroku výpočtu sa vyberie vzor matice plánu. Vzor matice sa vytvorí odstránením množiny náhodne vybraných riadkov. Veľkosť množiny je možné nastaviť. Nový vzor matice sa vytvára vždy v momente, keď medián prestane konvergovať. V konfiguračnom súbore je možné nastaviť, koľko krát sa ma nový vzor vygenerovať. Keď sa počet iterácii vyčerpá výpočet sa zastaví do databáze sa uloží množina riadkov podozrivých z odľahlosti, pre ktorú náleží hodnota minimalného mediánu. Tento test ukazuje aký vplyv má počet vygenerovaní náhodného vzoru matice na hodnotu

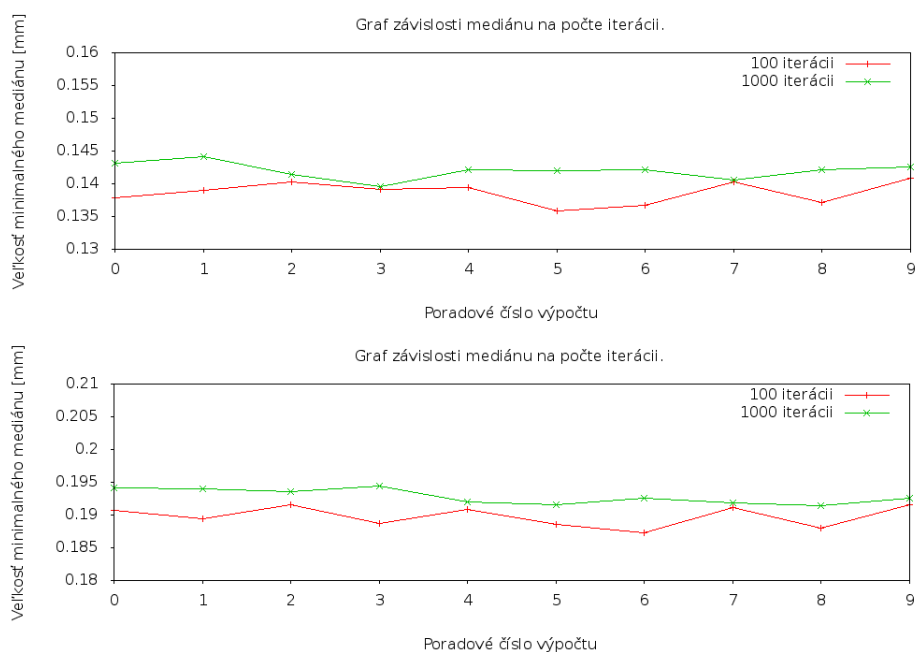
| Počet riadkov | Matica 1033x320 |            | Matica 1850x712 |            |
|---------------|-----------------|------------|-----------------|------------|
|               | Min medián      | Avg medián | Min medián      | Avg medián |
| 5             | 0.137896        | 0.152276   | 0.192285        | 0.202265   |
| 5             | 0.139003        | 0.151775   | 0.18824         | 0.202162   |
| 5             | 0.140354        | 0.152601   | 0.192611        | 0.202419   |
| 5             | 0.139185        | 0.151678   | 0.189821        | 0.202346   |
| 5             | 0.139424        | 0.152352   | 0.192488        | 0.202351   |
| 20            | 0.142611        | 0.153841   | 0.188641        | 0.202289   |
| 20            | 0.143181        | 0.1539     | 0.187249        | 0.202465   |
| 20            | 0.143839        | 0.153814   | 0.191194        | 0.202401   |
| 20            | 0.143561        | 0.15397    | 0.187978        | 0.202446   |
| 20            | 0.14173         | 0.154007   | 0.191518        | 0.202412   |
| 50            | 0.130028        | 0.14645    | 0.164072        | 0.192922   |
| 50            | 0.124461        | 0.146708   | 0.173507        | 0.193014   |
| 50            | 0.126117        | 0.146526   | 0.170846        | 0.192977   |
| 50            | 0.130396        | 0.146257   | 0.172608        | 0.19304    |
| 50            | 0.127249        | 0.146519   | 0.17313         | 0.192764   |
| 100           | 0.107238        | 0.137284   | 0.1592          | 0.178772   |
| 100           | 0.108985        | 0.137188   | 0.158868        | 0.178757   |
| 100           | 0.107053        | 0.137626   | 0.155846        | 0.178525   |
| 100           | 0.105991        | 0.137224   | 0.157237        | 0.178446   |
| 100           | 0.107445        | 0.137542   | 0.156281        | 0.178846   |

Tabuľka 13.4: Závislosť hodnoty minimalného mediánu na veľkosti množiny prevá-  
hovaných riadkov



Obrázok 13.8: Graf matice 1033x320 vľavo a 1850x712 vpravo

minimalného mediánu. V každom kroku výpočtu sa v tomto prípade odstráni 1% náhodne vybraných meraní resp. riadkov matice. Výsledky naznačujú, že pre túto úlohu má počet opakovaní pomerne malý vplyv, ale ukázalo sa, že vyšší počet opakovaní môže viesť k nižším hodnotám minimalného mediánu a určenie jeho veľkosti je pri vyššom počte opakovaní stabilnejšie. To sa ale môže meniť v závislosti na skutočnom počte kontaminovaných meraní a tiež na veľkosti množiny náhodne vylúčených meraní vzhľadom k celkovej veľkosti súboru meraní.



Obrázok 13.9: Graf matice 1033x320 vľavo a 1850x712 vpravo

## 13.9 Zhodnotenie výsledkov vyrovnaní geodetických sietí

Správnosť a robustnosť metódy najmenších mediánov štvorcov reziduí bola otestovaná na reálnych geodetických sietiach použitých pri určovaní posunov a pretvorení mostných konštrukcií či sietiach určených pre testovanie elektrooptických diaľkometerov. Definície geodetických sietí, zoznam meraní a presností boli dodané vo formáte xml, ktorý slúži ako vstupný formát pre vyrovnanie pomocou programu *gama-local*. Geodetické siete sú obecné vyrovňované ako voľné z dôvodu, aby sa sieť nedeformovala v závislosti na fixovaných bodoch. Voľné siete však vedú k zostaveniu singularnej matice plánu. Riešenie sústavy rovníc so singularnými maticami pomocou algoritmu združených gradientov nevedie k ukončeniu výpočtu nájdením minima funkcionálu. Výpočet sa zastaví pri dosiahnutí maximálneho počtu iterácií, čo vedie k značným chybám výpočtu sústavy a výsledok riešenia je prakticky nepoužiteľný. Z tohto dôvodu boli vstupné súbory upravené zafixovaním súradníc niektorých bodov. Vyrovnanie sietí bolo v prvom kroku vykonané klasickou metódou najmenších štvorcov pomocou programu *gama-local*. Aby bolo možné z programu *gama-local* získať maticu plánu a vektor redukovaných meraní, museli byť zdrojové kódy programu čiastočne upravené podľa kapitoly 13.7.1. Metóda LMS je primárne určená pre odhaľovanie observácií podozrivých z odľahlosti, preto sa do vstupných meraní zanesli hrubé chyby. V tejto kapitole bude popísaný priebeh testovania výpočtu geodetických sietí s rôznym rozdelením chýb, spôsob zhodnotenia a výsledky testovania.

### 13.9.1 Vyrovňované geodetické siete

Overenie metódy LMS bude vykonané na niekoľkých rovinných sietiach. V tabuľke je uvedený názov siete, ktorý je zhodný s názvom súboru. Počet meraní a počet neznámych určuje taktiež rozmery matíc. Na obrázku 11.1 sú znázornené grafy matíc plánu riešených úloh. Z grafov riedkych matíc 13.10 je patrné, že počet nenulových prvkov matice plánu nižšie uvedených sietí ani v jednom prípade nepresahuje ani 10% celkového počtu prvkov matice. Vo všetkých prípadoch je teda výhodné uložiť prvky matice do dátových štruktúr určených pre riedke matice. V každom kroku sú neznáme parametre lineárnych rovníc určované pomocou metódy združených gradientov bez zostavenia normálnych rovníc.

| Názov siete     | počet meraní | počet neznámych |
|-----------------|--------------|-----------------|
| jerabova draha  | 256          | 122             |
| jindriska       | 102          | 45              |
| mikrosit        | 116          | 38              |
| most klasterec  | 1899         | 610             |
| most sany       | 708          | 304             |
| rude pravo      | 165          | 74              |
| sit ocelarny    | 66           | 21              |
| stitky ocelarny | 717          | 200             |
| zakladna emeg   | 122          | 40              |

Tabuľka 13.5: Vyrovnávané siete

### 13.9.2 Overenie metódy LMS na vybraných geodetických sieťach

Podrobné výsledky overovania metódy LMS boli získané na základe vyrovnania geodetických sietí určených pre stanovenie veľkosti posunov a rozsahu pretvorení mostových konštrukcií v Klášterci nad Ohří a v obci Sány. Pre účely testovania boli meraným hodnotám týchto sietí priradené odchýlky rôznych veľkostí. Priradovanie poruch meraniam pre účely zhodnotenia LMS je možné rozdeliť do dvoch skupín a to na priradenie poruch rovnakej veľkosti variabilnému počtu meraných hodnôt a na priradenie poruch rôznej veľkosti rovnakému počtu meraní. Prvý prípad má ukázať, či metóda LMS dokáže identifikovať celú množinu porušených meraní nezávisle na jej veľkosti. V druhom prípade sa určuje veľkosť odchýlok meraných hodnôt, ktoré dokáže overovaná metóda identifikovať. Pre obe mostné konštrukcie boli zvolené rovnaké prípady porúch. Veľkosť množiny meraní s priradenou hrubou chybou v rámci decimetrov v šikmej dĺžke a v rámci desatiny gónu v zvislom uhle a smere boli zvolené na 10, 20, a 40 porušených meraní. Pri testovaní citlivosti metódy na veľkosť chyby merania bolo porušené vždy len jedno meranie a to buď smer, alebo dĺžka. Veľkosti poruchy meraní sa pre tieto meranie pohybovali v rámci desatiny, stotiny a tisíciny gónu smeru a v rámci decimetrov a centimetrov v dĺžke. Tabuľka 13.9.2 ukazuje spôsob rozdelenia odchýlok meraniam



| Názov súboru           | Rozsah porušených meraní | Veľkosť odchýlky dĺžky | Veľkosť odchýlky smeru |
|------------------------|--------------------------|------------------------|------------------------|
| most_klasterec_10.gkf  | 10                       | ± 10cm                 | ± 0.1gon               |
| most_klasterec_20.gkf  | 20                       | ± 10cm                 | ± 0.1gon               |
| most_klasterec_40.gkf  | 40                       | ± 10cm                 | ± 0.1gon               |
| most_klasterec_acm.gkf | 1                        | -                      | ± 0.1gon               |
| most_klasterec_adm.gkf | 1                        | -                      | ± 0.01gon              |
| most_klasterec_amm.gkf | 1                        | -                      | ± 0.001gon             |
| most_klasterec_dcm.gkf | 1                        | ± 1cm                  | -                      |
| most_klasterec_ddm.gkf | 1                        | ± 10cm                 | -                      |
| most_sany_10.gkf       | 10                       | ± 10cm                 | ± 0.1gon               |
| most_sany_20.gkf       | 20                       | ± 10cm                 | ± 0.1gon               |
| most_sany_40.gkf       | 40                       | ± 10cm                 | ± 0.1gon               |
| most_sany_acm.gkf      | 1                        | -                      | ± 0.1gon               |
| most_sany_adm.gkf      | 1                        | -                      | ± 0.01gon              |
| most_sany_amm.gkf      | 1                        | -                      | ± 0.001gon             |
| most_sany_dcm.gkf      | 1                        | ± 1cm                  | -                      |
| most_sany_ddm.gkf      | 1                        | ± 10cm                 | -                      |

Tabuľka 13.6: Tabuľka spôsobu rozdelenia porúch meraniam

V tabuľke 13.9.2 sú uvedené merania reprezentované číslami riadkov matíc plánu, ktorým boli rozdelené odchýlky rôznych veľkostí. Každému meraniu bola priradená odchýlka, ktorej veľkosť je uvedená v tabuľke spôsobu rozdelenia poruch meraniam. Výsledkom metódy LMS je práve množina riadkov s odľahlými meraniami, ktoré prislúžia hodnotám z tabuľky.

Tabuľka 13.8 obsahuje hodnoty nastavení metódy LMS. Nastavenia boli zvolené v závislosti na výsledkoch testovania metódy na vzorových maticiach uvedených v kapitole 13.8. Možnosti konfigurácie metódy sú podrobne popísané v 13.6.1. Každá úloha môže mať niekoľko rôznych konfigurácií výpočtu. Napríklad úloha určenia deformácií mostnej konštrukcie v Klášterci s desiatimi odľahlými meraniami (most\_klasterec\_10x) sa bude počítať s maximálnym počtom 100 iterácií, veľkosť náhodne vybraného vzoru matice bude 99, 90 a 80 percent z celkovej veľkosti matice plánu. Počet odstránených riadkov v jednej iterácii výpočtu bude 10 alebo 20. Rôzne konfigurácie výpočtu jednej úlohy majú ukázať, akou mierou vplýva daná konfigurácia na výsledky výpočtu geodetických sietí tzn. detekciu meraní podozrivých z odľahlosti, hodnoty minimálneho mediánu v jednotlivých iteráciách výpočtu či priebeh konverencie mediánu.

| Názov súboru           | Riadky s odľahlými meraniami  |
|------------------------|---|
| most_klasterec_10.gkf  | 30, 91, 92, 93, 206, 213, 215, 305, 306, 307  |
| most_klasterec_20.gkf  | 8, 11, 14, 53, 54, 59, 116, 117, 118, 178<br>179, 180, 185, 223, 227, 230, 268, 269, 271, 319   |
| most_klasterec_40.gkf  | 11, 13, 35, 54, 59, 72, 73, 145, 146, 225<br>227, 231, 236, 241, 248, 252, 253, 255, 256, 312<br>313, 317, 319, 326, 330, 332, 337, 338, 341, 343<br>480, 482, 490, 491, 503, 507, 509, 513, 526, 530 |
| most_klasterec_acm.gkf | 0   |
| most_klasterec_adm.gkf | 0   |
| most_klasterec_amm.gkf | 0   |
| most_klasterec_dcm.gkf | 30  |
| most_klasterec_ddm.gkf | 30  |
| most_sany_10.gkf       | 9, 33, 253, 329, 426, 556, 557, 558, 559  |
| most_sany_20.gkf       | 1, 3, 18, 19, 20, 70, 71, 72, 73, 110<br>111, 112, 114, 116, 209, 210, 211, 212   |
| most_sany_40.gkf       | 15, 29, 35, 42, 53, 62, 115, 118, 135, 136<br>144, 145, 165, 170, 187, 188, 200, 201, 210, 219<br>227, 230, 633, 634, 635, 636, 638, 641, 644, 654<br>659, 660, 661, 663, 664, 702, 704, 706, 707     |
| most_sany_acm.gkf      | 0   |
| most_sany_adm.gkf      | 0   |
| most_sany_amm.gkf      | 0   |
| most_sany_dcm.gkf      | 9   |
| most_sany_ddm.gkf      | 9   |

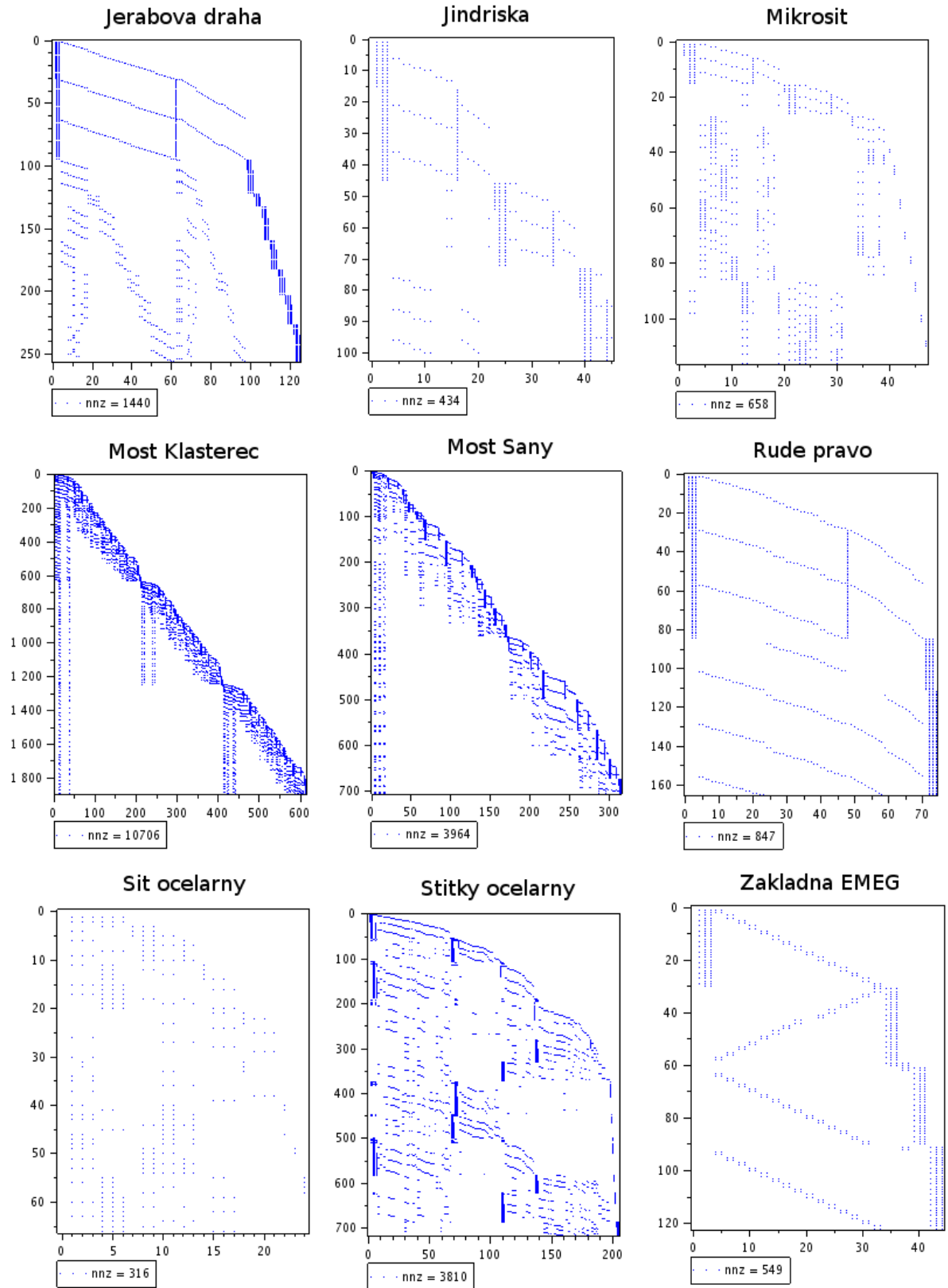
Tabuľka 13.7: Tabuľka riadkov s odľahlými meraniami

| Názov súboru           | Max. interácií | Veľkosť vzoru | Preváňované riadky | Metóda        |
|------------------------|----------------|---------------|--------------------|---------------|
| most_klasterec_10a.set | 100            | 80            | 10                 | ExtremeWeight |
| most_klasterec_10b.set | 100            | 99            | 10                 | ExtremeWeight |
| most_klasterec_10c.set | 100            | 90            | 20                 | ExtremeWeight |
| most_klasterec_10.set  | 100            | 99            | 20                 | ExtremeWeight |
| most_klasterec_20.set  | 100            | 99            | 20                 | ExtremeWeight |
| most_klasterec_40a.set | 100            | 80            | 40                 | ExtremeWeight |
| most_klasterec_40b.set | 100            | 90            | 50                 | ExtremeWeight |
| most_klasterec_40c.set | 100            | 99            | 40                 | ExtremeWeight |
| most_klasterec_40.set  | 100            | 99            | 20                 | ExtremeWeight |
| most_klasterec_acm.set | 100            | 99            | 20                 | ExtremeWeight |
| most_klasterec_adm.set | 100            | 99            | 20                 | ExtremeWeight |
| most_klasterec_amm.set | 100            | 99            | 20                 | ExtremeWeight |
| most_klasterec_dcm.set | 100            | 99            | 20                 | ExtremeWeight |
| most_klasterec_ddm.set | 100            | 99            | 20                 | ExtremeWeight |
| most_klasterec_err.set | 100            | 99            | 20                 | ExtremeWeight |
| most_klasterec_fix.set | 100            | 99            | 20                 | ExtremeWeight |
| most_klasterec.set     | 100            | 99            | 20                 | ExtremeWeight |
| most_sany_10a.set      | 100            | 80            | 20                 | ExtremeWeight |
| most_sany_10b.set      | 100            | 90            | 30                 | ExtremeWeight |
| most_sany_10c.set      | 100            | 90            | 10                 | ExtremeWeight |
| most_sany_10.set       | 100            | 99            | 20                 | ExtremeWeight |
| most_sany_20.set       | 100            | 99            | 20                 | ExtremeWeight |
| most_sany_40a.set      | 100            | 90            | 40                 | ExtremeWeight |
| most_sany_40b.set      | 100            | 99            | 50                 | ExtremeWeight |
| most_sany_40c.set      | 100            | 80            | 40                 | ExtremeWeight |
| most_sany_40.set       | 100            | 99            | 20                 | ExtremeWeight |
| most_sany_acm.set      | 100            | 99            | 20                 | ExtremeWeight |
| most_sany_adm.set      | 100            | 99            | 20                 | ExtremeWeight |
| most_sany_amm.set      | 100            | 99            | 20                 | ExtremeWeight |
| most_sany_dcm.set      | 100            | 99            | 20                 | ExtremeWeight |
| most_sany_ddm.set      | 100            | 99            | 20                 | ExtremeWeight |
| most_sany_err.set      | 100            | 99            | 20                 | ExtremeWeight |
| most_sany_fix.set      | 100            | 99            | 20                 | ExtremeWeight |
| most_sany.set          | 100            | 99            | 20                 | ExtremeWeight |

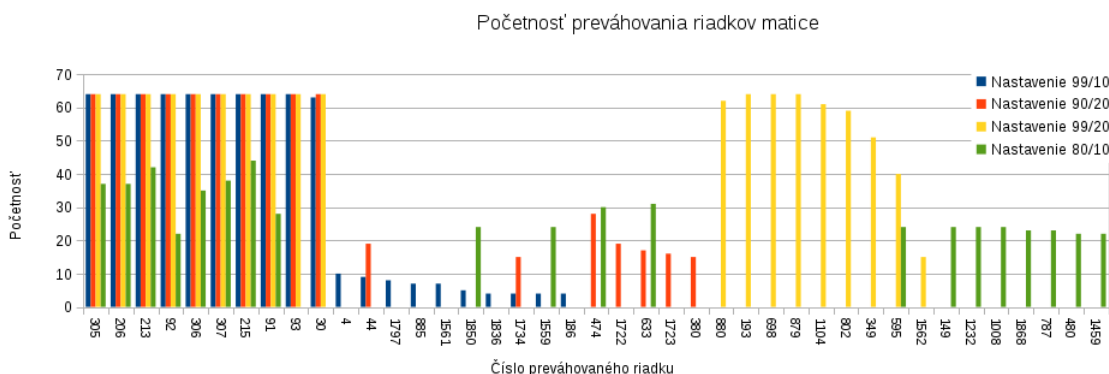
Tabuľka 13.8: Tabuľka konfigurácií metódy LMS

Histogramy obsahujú početnosť vylúčení meraní pre rôzne konfigurácie výpočtu sietí v Klášterci nad Ohří a obci Saný. Celkový súbor meraní obsahuje v oboch prípadoch 10 kontaminovaných meraní. Z histogramu 13.11 početnosti vylúčených kontaminovaných meraní v obci Klášterec nad Ohří je patrné, že pokiaľ je z matice plánu vylúčený minimálny počet náhodných observácií a nastavenie počtu vylúčených riadkov je rovný počtu kontaminovaných meraní (99/10 modrá farba v grafe 13.11) je určenie kontaminovaných meraní pomerne stabilné. Aby toho ale bolo možné dosiahnuť, musel by byť počet odľahlých meraní predom známy. Rozširovanie množiny náhodne vylúčených meraní už nevedie k tak jednoznačnému určeniu odľahlých meraní (80/10 zelená farba v grafe 13.11). Je ale zrejmé, že sa v množine riadkov podozrivých z odľahlosti nachádzajú aj skutočne kontaminované merania, ktoré ale nie je možné jednoznačne učiť. Pokiaľ sa ale množina náhodne vylúčených riadkov minimalizuje a to cca do 10% z celkového súboru meraní, tak určenie skutočných odľahlých meraní je jednoznačné aj pri nepresnom nastavení odhadu ich počtu. Červená farba v histograme ukazuje, že počiatočný odhad množstva odľahlých meraní bol 20 a kontaminované merania boli jednoznačne určené. Nastavenie prisúdené žltej farbe (99/20), ktorého množina náhodne odstránených riadkov je len 1%, sice spoľahlivo určí skutočne odľahlé merania, ale na druhú stranu vo vysokej početnosti ukáže aj na riadky, ktoré kontaminované nie sú. Z výsledkov je teda možné vyvodiť záver, že riešenie metódy LMS je silne závislé na jej konkrétnom nastavení. Je teda potrebné určiť približný odhad počtu kontaminovaných meraní ešte pred započatím výpočtu. Určenie tohto odhadu bude predmetom ďalšieho skúmania. Obdobné výsledky sú patrné aj z histogramu 13.12 patriacemu výsledkom výpočtu siete v obci Saný.

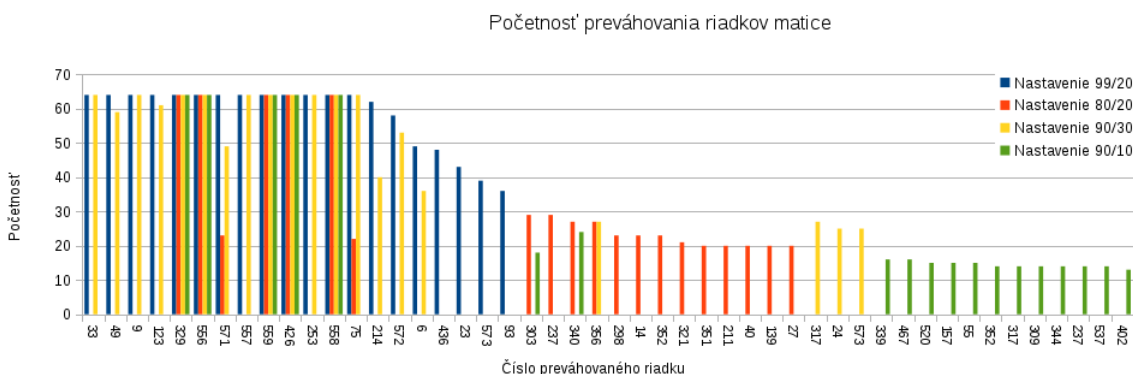
Grafy hodnôt priemerného a minimálneho mediánu pre obe siete v závislosti na konfiguráciách výpočtu jak pre obec Klášterec, tak pre obec Saný, vykazujú podobné výsledky. Z grafov je patrné, že rozdiely medzi hodnotami minimálneho či priemerného mediánu sa zväčšujú v závislosti na veľkosti množiny náhodne odstránených riadkov. Variabilitu by bolo možné zmierniť zvýšením počtu generovaní náhodného vzoru. Veľkosť množiny náhodne odstránených riadkov ovplyvňuje aj rýchlosť konvergencie mediánu. Čím je množina väčšia, tým viac cyklov je potrebné opakovať pre získanie minimálnych hodnôt mediánu 13.16. Z grafu, ktorý znázorňuje hodnoty priemerného a minimálneho mediánu v úlohe, kde je počet skutočne kontaminovaných meraní rovný 40. Z grafu 13.17 je patrné, že čím viac je meraní kontaminovaných, tým sú hodnoty mediánu mediánu viac závislé na konkrétnom nastavení. Tieto hodnoty je možné použiť pre odhad správnej konfigurácie metódy.



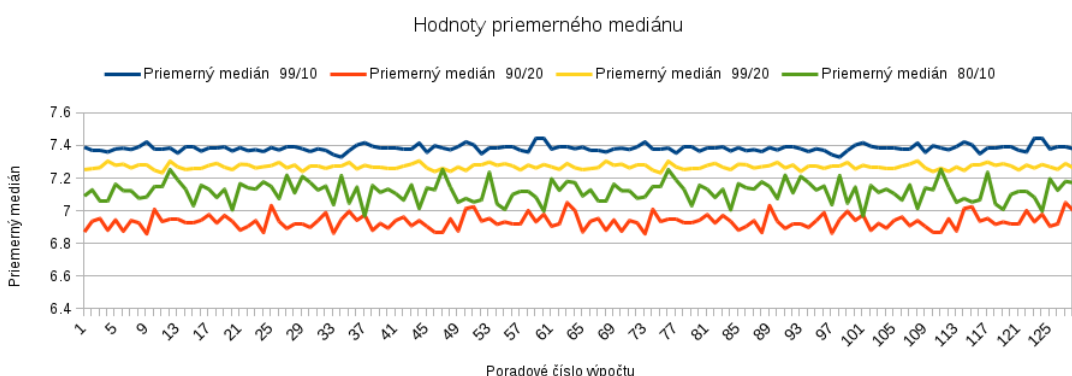
Obrázok 13.10: Grafy riedkych matíc geodetických sietí



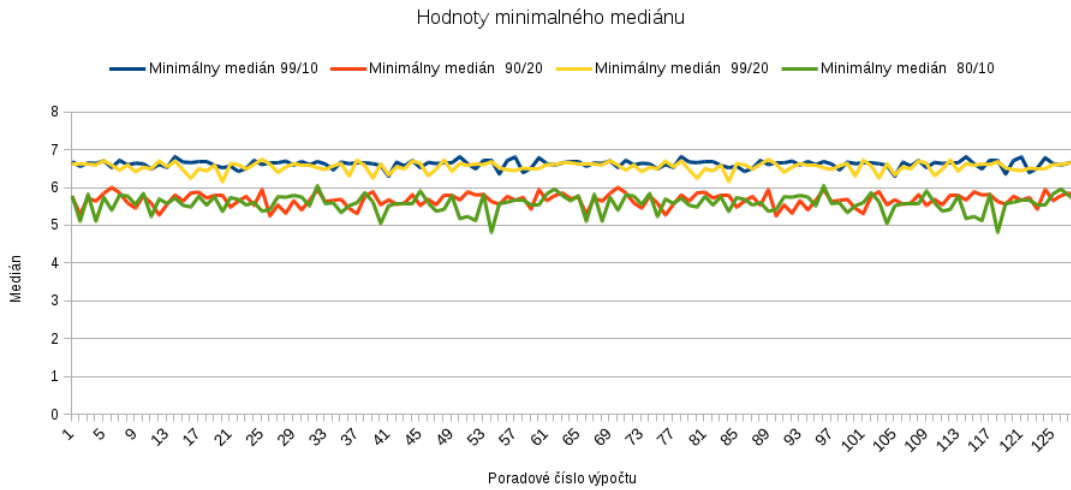
Obrázok 13.11: Početnosť odstránení meraní podozrivých z odľahlosti mostnej konštrukcii v obci Klášterec nad Ohří



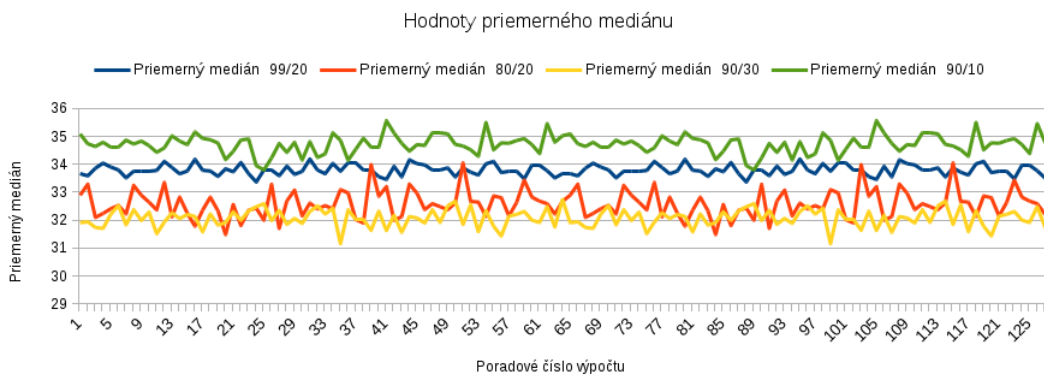
Obrázok 13.12: Početnosť odstránení meraní podozrivých z odľahlosti mostnej konštrukcii v obci Saný



Obrázok 13.13: Hodnoty priemerného mediánu LMS mostnej konštrukcii v obci Klášterec nad Ohří pre 10 vylúčených meraní

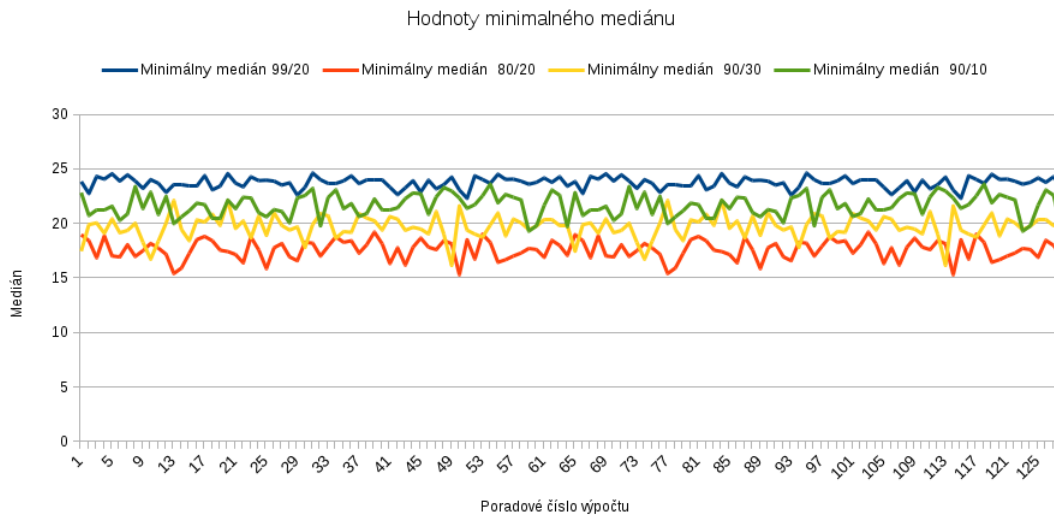


Obrázok 13.14: Hodnoty minimalného mediánu metódy LMS mostnej konštrukcii v obci Klášterec nad Ohří pre 10 vylúčených meraní

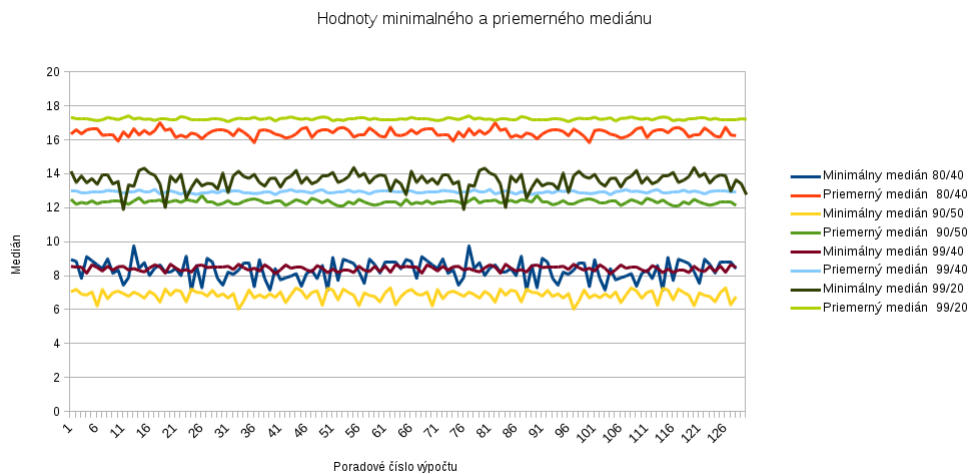


Obrázok 13.15: Hodnoty priemerného mediánu metódy LMS mostnej konštrukcii v obci Saný pre 10 vylúčených meraní

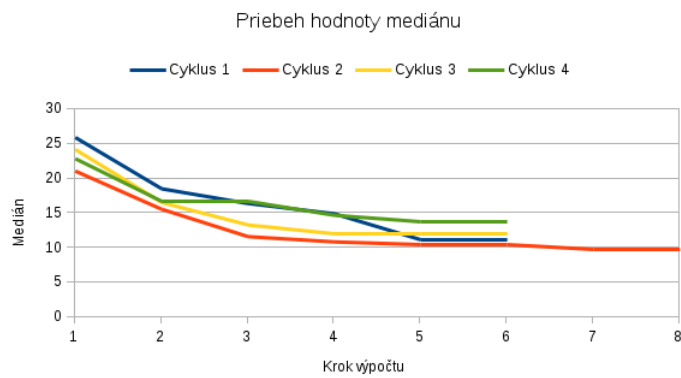




Obrázok 13.16: Hodnoty minimalného mediánu metódy LMS mostnej konštrukcii v obci Saný pre 10 vylúčených meraní



Obrázok 13.17: Hodnoty minimalného a priemerného mediánu metódy LMS mostnej konštrukcii v obci Klášterec nad Ohří pre 40 vylúčených meraní



Obrázok 13.18: Priebek poklesu mediánu metódy LMS aplikovanej pri výpočte mostnej konštrukcii v obci Klášterec nad Ohří

## Záver

Predkladaná dizertačná práca zhrňuje doterajšie poznatky o možnostiach použitia štatistických a numerických metód aplikovaných na rôzne výpočty v geodetických aplikáciach. Prvá časť dizertačnej práce obsahuje rešerš doterajších metód použitých k riešeniu LMS. Z naštudovanej literatúry vyplýva, že exaktné riešenie LMS je možné prakticky vykonať iba na pomerne malých vstupných dátach. Dôvodom je, že minimalizovaná funkcia má mnoho lokálnych extrémov. Mnoho autorov preto pristúpilo k aproximácii riešenia rôznymi heuristickými metódami. Aj navzdory použitiu približného výpočtu odhadu neznámych parametrov, či určenia množiny kontaminovaných meraní je nutné opakovane riešiť sústavu lineárnych algebraických rovníc. Tento problém bol dôvodom skúmania možnosti zvýšenia efektivity výpočtu využitím vstupných predpokladov špecifických pre geodetické merania. Prvým predpokladom je riedkosť vstupnej matice diferenciálnych prírastkov. Matica má v mnohých prípadoch riedku štruktúru a to ako bolo naznačené v kap. 10.4 v prípade vyrovnania GNSS meraní, či v prípade terestrických meraní v kap. 13.9.1. V kap. 10.4.4 bolo ukázané, že výberom vhodnej permutácie môžeme pri vyrovnaní GNSS meraní zachovať riedku štruktúru aj v prípade riešenia lineárnych rovníc Choleského rozkladom. Pre opakované riešenie lineárnych rovníc bola zvolená metóda združených gradientov, ktorá patrí do skupiny iteračných metód. Metóda nevyžaduje vyhotovenie normálnych rovníc pre riešenie sústavy, a preto je možné vyhnúť sa vzájomnému násobeniu matíc. Keďže metóda združených gradientov je iteračná metóda ponúkalo sa zistenie možnosti urýchlenia konvergencie nejakou metódou prepodmienevania. Ako prepodmienovač sa často používa neúplný Choleského rozklad. Choleského rozklad ale zaplňuje riedku maticu, preto bol v tejto práci zvolený špeciálny typ neúplného Choleského rozkladu tzv. ILU0, teda rozklad, v ktorom sa pri vzniku nenulového prvku v priebehu ekvivalentných riadkových úprav tento vypočítaný prvok nepoužije a ponechá sa na danom mieste nulová hodnota. Na testovanie numerickej stability a rýchlosti algoritmov boli mimo iné použité matice generované zo stránok *Matrix Marketu*, ktorá je spravovaná matematickým oddelením Amerického národného inštitútu štandardov a technológií. Po implementácii všetkých potrebných algoritmov a prebehnutí zdĺhavých výpočtov bolo v kap. 13.8.1 zhodnotené značné množstvo dát, na základe ktorého bola posúdená vhodnosť zvolenej metódy. Po vykonaní potrebných overení funkčnosti numerických metód bola skúmaná možnosť použitia paralelného výpočtu. Výsledky v kap. 3.1.4 ukázali, že v prípade LMS je efektívnejšie paralelizovať minimalizáciu mediánu ako takého oproti paralelizácii metódy združených gradientov. Dôvodom bol fakt, že v metóde združených gradientov sa opakovanie vykonáva operácia násobenia dvoch vektorov a režijne náklady na vykonanie paralelizácie vzhľadom k veľkosti vektorov prevyšujú samotný para-

lélny výpočet. Popis efektívnej paralelizácie metódy LMS je naznačený v kap. 8.6.3. V tomto prípade bola paralelizácia vykonaná na metóde LMS s vyhotovením všetkých kombinácií výpočtu. Jednoduchou modifikáciou je možné paralelizovať taktiež aproximovaný výpočet LMS so stratégiou polovičných vzorov či vzorov variabilnej veľkosti.

V ďalšom kroku bolo nutné porovnať metódu LMS s vybranými robustnými metódami. Pre tento účel bol použitý exaktný výpočet odhadu neznámych parametrov. Metóda LMS bola porovnaná s metódami M-odhadov, s metódou LTS a RANSAC. Výsledky podrobne popísané v kap. 9.5 ukázali prípady kedy je vhodné použitie metód LMS oproti ostatným metódam. Z výsledkov vyplýva, že metóda LMS je čo sa týka rýchlosti výpočtu jednou z najmenej efektívnych. Nepatrne časovo náročnejšia je z posudzovaných metód iba metóda LTS, kde výpočet sumy štvorcov reziduí je časovo menej efektívny než určenie mediánu. Metóda LMS ale na druhú stranu oproti metóde M-odhadov dáva potenciálne správny odhad neznámych parametrov aj pri vysokej kontaminácii meraní.

Po porovnaní metódy LMS s ostatnými robustnými metódami bola metóda použitá pre určenie odľahlých geodetických meraní. Vstupnými dátami boli geodetické merania vyhotovené pre účel určovania posunov monolitických konštrukcií. Do týchto dát boli zámerne zanesené hrubé chyby. Výsledky ukázali, že pri použití metódy LMS je možné pomerne spoľahlivo určiť odhad neznámych parametrov aj v prípade 40% meraní. Cenou za to je ale strata 50% pôvodnej informácie aj v prípadoch, keď meranie hrubé chyby neobsahuje. Metóda bola preto použitá iba na detekciu odľahlých meraní. Po ich rozpoznaní budú potenciálne kontaminované hodnoty z merania odstránené a k odhadu neznámych parametrov bude použitá metóda MNČ. K čiastočným overeniam správnosti určenia odľahlých meraní, hlavne v prípade vysokého počtu kontaminácie, slúžila metóda testu štatistickej hypotézy kvadratickej formy vyrovnaných reziduí.

Odstránenie takto veľkého množstva meraní môže spôsobiť singularitu pôvodnej sústavy lineárnych rovníc. Tento problém je pomerne ťažko odhaliť. Jednou z možností určenia potenciálnej singularity je analýza geodetických meraní pred započatím riešenia sústavy lineárnych rovníc tzv. určením súvislosti geodetickej siete. Implementovaný algoritmus dokáže rozpoznať, že sa z určovaného bodu vylúčili všetky merania. Vstupným súborom pre analýzu merania je XML programu *gama-local*. Nevýhodou je nízka efektivita algoritmu. Ďalším zo spôsobov je singularitu z počiatku ignorovať. Ako už bolo spomenuté, na výpočet sústavy lineárnych rovníc je použitá iteračná metóda združených gradientov. Tá v prípade singularných sústav nebude konvergovať. To znamená, že sa nebude prejavovať pokles chyby výpočtu algoritmu. Výpočet sa napokon zastaví na maximálnom dovolenom počte iterácii. Keď tento prípad nastane, tak je takmer isté, že tento dielčí výsledok je nepresný

a z riešenia bude vylúčený.

V kapitole 12 bol popísaný návrh algoritmu a jeho implementácia pre metódu LMS s rozdelením pravdepodobnosti diskretných náhodných veličín generovania náhodných vzorov sústavy. Na základe výsledkov získaných z vykonaných experimentov je možné tvrdiť, že táto metóda dokáže značne vylepšiť konvergenciu mediánu a s jej použitím je možné získať správny odhad už pri pomerne nízkom počte opakovaní výberu náhodného vzoru. Metóda nijakým spôsobom neupravuje pôvodné merané hodnoty. Metóda iba modifikuje pravdepodobnosť výberu merania v závislosti na veľkosti vyrovnaných opráv v každom kroku výpočtu. Pre túto metódu bolo navrhnutých niekoľko stratégií diskretného rozdelenia pravdepodobnosti a to lineárna, exponenciálna a metóda rozdelenia pravdepodobnosti skupinám meraní. Najlepšie výsledky vykazovala stratégia lineárneho rozdelenia pravdepodobností meraniam podozrivým z odľahlosti.

V dizertačnej práci boli taktiež skúmané možnosti riešenia metódy LMS preváhaním meraní podozrivých z odľahlosti. V podstate sa jedná o akúsi kombináciu LMS a M-odhadov. V práci bolo navrhnutých niekoľko stratégií váhovania popísaných v kap. 13.2.2. Program popísaný v kap. 13.6 na výpočet LMS vyhotovený v C++ používa práve na výpočet LMS tieto stratégie. S pomocou tohto programu bol vykonaný veľký počet výpočtov s rôznymi konfiguráciami metódy. Konfiguráciami sa myslí veľkosť náhodného vzoru, počet iterácií, zvolená stratégia váhovania a pod.. Pre opakovanie výpočtov s rôznymi konfiguráciami boli použité skripty jazyka *bash*. Program vygeneroval mohutné množstvo výsledkov a preto jediným možným nástrojom na ich spracovanie bola databáza. Výsledky boli preto uložené do databázy *PostgreSQL* a na ich spracovanie bol použitý jazyk SQL. Vypočítané hodnoty z kapitoly 13.8 naznačujú, že veľkosť mediánu závisí na počte odstránených observácií v jednom kroku výpočtu, ale na druhu stranu zásadne nezávisí na počte pseudo-náhodných výberov. To znamená, že zvyšovaním iterácií sa zásadne nemení minimum mediánu a to preto, že pseudo-náhodný výber väčšinou neobsahuje dáta z vysokými štvorcami rezíduí. To ale platí iba pre tieto konkrétne testované dáta a to z toho dôvodu, že vyššie rezíduum vykazujú len jednotky observácií. To znamená, že kontaminované je iba promile observácií. Z histogramov 13.11 a 13.12 ďalej vyplýva, že opakovaným výpočtom je možné spoľahlivo určiť observácie s vysokými reziduami a z výpočtu ich odstrániť. Pri zadaných dátach a testovanej konfigurácii sa ukázalo, že výber stratégie váhovania zásadne neovplyvní výsledky mediánu, ale môže ovplyvniť celý výpočet. To znamená, že pokiaľ sa rozhodne, že sa dáta zo súboru nevyľúčia úplne, je možné ich preváhať lineárne v závislosti na veľkosti získaných štvorcach rezíduí a tieto výsledky použiť.

Metódu LMS je v geodézii výhodné použiť práve vtedy, keď očakávame veľké množstvo odľahlých meraní vo veľkom množstve vstupných dát, kde počet meraných

hodnôt značne prevyšuje počet hodnôt určovaných. Pre úplné využitie kladných vlastností LMS, to znamená vysokého bodu zvratu, je možné na základe výsledkov z tejto práce doporučiť použitie LMS s generovaním náhodných polovičných vzorov. Dôvodom je, že už pri pomerne nízkom počte opakovaní generovania náhodných vzorov bude pravdepodobnosť určenia správneho odhadu neznámych parametrov pomerne vysoká. Cenou za túto výhodu bude ale strata značnej časti pôvodnej informácie získanej meraním.

## Referencie

- [Steele–1] STEELE J.M., STEIGER W.L.: *Algorithms and complexity for least median of squares regression*  
Elsevier Science Publishers B.V., 1986, Discrete Applied Mathematics 14, 93-100 s
- [Souvaine–2] SOUVAINÉ D. L., STEELE J. M.: *Time- and Space-Efficient Algorithms for Least Median of Squares Regression*  
Journal of the American Statistical Association, Vol. 82, No. 399 (Sep., 1987), 794- 801 s
- [Joss–3] JOSS J., MARAZZI A.: *Probabilistic algorithms for Least Median of Squares regression*  
Computational Statistics and Data Analysis 9, 0167-9473/90, 1990, 123-133 s
- [Shather–4] HETTMANSPERGER T. P., SHATHER S. J.: *A Cautionary Note on the Method of Least Median Squares*  
The American Statistician, 46:2, ISSN: 0003-1305, 79-83 s
- [Erickson–5] ERICKSON J., HAR-PELED S., MOUNT D.: *On the Least Median Square Problem*  
June 8–11, 2004, Brooklyn, New York, USA., Copyright 2004 ACM 1581138857/ 04/0006
- [Hawkins–6] HAWKINS D. M.: *The feasible set algorithm for least median of squares regression*  
Elsevier Science Publishers B.V., Computational Statistics and Data Analysis 16, 0167-9473/93/, 1993, 81-101 s
- [Olson–7] OLSON C. F.: *An approximation algorithm for least median of squares regression*  
Elsevier Science B.V., Information Processing Letters 63, 1997,0020.0190/97, 237-241 s
- [Cook–8] COOK R. D., HAWKINS D. M., WEISBERG S.: *Comparison of Model Misspecification Diagnostics Using Residuals From Least Mean of Squares and Least Median of Squares Fits*  
Journal of the American Statistical Association, 1992, 87:418, 419-424 s

- [Dallal–9] DALLAL G. E., ROUSSEEUW P. J.: *LMSMVE: A Program for Least Median of Squares Regression and Robust Distances* Academic Press. Inc., COMPUTERS AND BIOMEDICAL RESEARCH 25, 1992, 0010-480919, 384-391 s
- [Karr–10] KARR C. L., WECK B., MASSART D. L., VANKEERBERGHEN P.: *Least Median Squares Curve Fitting Using a Genetic Algorithm* Elsevier Science B.V., EngngAppl. Artif. Intell. Vol. 8, No. 2, 1995, 0952-1976(94)011064-6, 177-189 s
- [Massart–11] DESIRE L., MASSART D., KAUFMAN L., ROUSSEEUW P., LERROY A.: *Least Median of Squares: A Robust Method for Outlier and Model Error Detection in Regression and Calibration*, Elsevier Science Publishers, Atlantica Chimica Acta, 1986, 0003-2670/86, 171-179 s,
- [Vandev–12] VANDEV D. L.: *Stochastic Optimization Algorithm Applied to Least Median of Squares Regression* Pliska Studia Mathematica Bulgar. 17, ISSN 0204-9805, 2005, 323-335 s Dostupný z WWW: <http://www.math.bas.bg/~pliska/>
- [Rusiecki–13] RUSIECKI A.: *Robust Learning Algorithm Based on Iterative Least Median of Squares* Springer DOI 10.1007/s11063-012-9227-z, Neural Process Lett 36, 2012, 145–160 s
- [Chong-wei–14] CHONG-WEI XU, WEI-KEY SHIUE: *Parallel algorithms for least median of squares regression* Elsevier Science B.V., Computational Statistics and Data Analysis 16, 1993, 0167-9473/93, 349-362 s
- [Stromberg–15] STROMBERG A. J.: *Computing the exact least median of squares estimate and stability diagnostics in multiple linear regression* SIAM J. SCI. COMPUT. Vol. 14, No. 6, November 1993, 1289-1299 s
- [Rousseeuw–16] ROUSSEEUW P.: *Least Median of Squares Regression*, Journal of the American Statistical Association, December 1984, Volume 79, Number 388, Theory and Methods Section
- [Saad–17] SAAD Y.: *Iterative Methods for Sparse Linear Systems*, Second Edition, 2000, 439 s, [online]



- Dostupný z WWW: <http://www-users.cs.umn.edu/~saad/books.html>
- [Golub–18] GOLUB G.H., VAN LOAN CH.F. : *Matrix Computations*, Third Edition, The John Hopkins University Press, 1996, ISBN 0-8018-5431-X, 687 s.
- [Marek–19] MAREK I., MEYER P., SEKERKA B.: *Úvod do numerické matematiky*, Prednáška pro posluchače matematiky, 101 s.
- [Koc–20] KOČANDRLOVÁ M.: *Úvod do lineární algebry*, Studijní text pro obor G+K, ČVUT Praha, 2003, 45 s.
- [Liu–21] GEORGE A., LIU J., NG E.: *Comuter Solution of Sparse Linear System*, 1994, 401 s.
- [Koch–22] KOCH K.: *Parameter Estimation and Hypothesis Testing in Linear Models*, Second updatet and enlarged Edition, Springer-Verlag, 1997, ISBN 3-540-65257-4, 327 s.
- [Jerabek–23] JEŘÁBEK M.: *Využití paralelních výpočtů v geodézii*, Doktorská disertační práce, ČVUT Praha, Zář 2001
- [Čepek–24] ČEPEK A., PYTEL J.: *A Note on Numerical Solutions of Least Squares Adjustment in GNU Project Gama* In: *Interfacing Geostatistics and GIS*. Berlin: Springer-Verlag, 2009, pp. 179-193. ISBN 978-3-540-33235-0.
- [Mikhail–25] MIKHAIL E., ACKERMAN F.: *Oservations and Least Squares*, Thomas Y. Crowell Company, 1976, ISBN 0-7002-2481-5, 327 s.
- [Davis–26] DAVIS T.: *Direct Methods for Sparse Linear Systems*, Siam, 2006, ISBN-10: 0-89871-613-6, 211 s.
- [SparseLib–27] DONGARRA J., LUMSDAINE A., NIU X., POZO R., REMINGTON K.: *A Sparse Matrix Library in C++ for High Performance Architectures*, Dostupný z WWW: <http://math.nist.gov/sparselib++/>
- [Eigen–28] Dostupný z WWW: <http://eigen.tuxfamily.org/>
- [Gama–29] ČEPEK A.: *Manuál k programu Gama 1.10*, 1. vyd. Praha, 2009. 56 s. Dostupný z WWW: <http://www.gnu.org/software/gama/>

- [Trasak–30] TŘASÁK P.: *Simulace, modelování a statistické zpracování geodetických měření*, Doktorská disertační práce, ČVUT Praha, Březen 2013
- [Kanani–31] KANANI E.: *Robust Estimators dor Geodetic Transformations and GIS*, Doktorská disertační práce číslo 13521, Institut fur Geodäsie and Photogrametric, ETH Curych, 2000, ISBN 3-906467-26-0
- [Yang–32] YANG L., WNAG J.: *Reliability analysis of robust estimation for GNSS positioning*, International Global Navigation Satellite Systems Society IGNSS Symposium 15-17 nowember 2011, University of New South Wales, Sydney, Australia
- [Mervart–33] MERVART L., LUKEŠ Z.: *Adjustment csclculus*, 1. vyd., ČVUT Praha, 2007, ISBN 978-80-01-02593-1, 165 s.
- [Cepek–34] ČEPEK A.: *Implementace Givensovy transformace pro vyrovnání měření zprostředkujících s řídkou maticí soustavy rovnic oprav*, VUGTK Praha, 1992, 123 s.
- [Charamza–35] CHARAMZA F.: *Řešení základních úloh vyrovnávacího počtu ortogonalizační metodou*, VUGTK Praha, 1972
- [Cplusplus–36] Dostupný z WWW: <http://en.cppreference.com/w/cpp/atomic/atomic>
- [Ruckstuhl–37] RUCKSTUHL A.: *Robust Fitting of Parametric Models Based on M-Estimation*, IDP Institute of Data Analysis and Process Design ZHAW Zurich University of Applied Sciences in Winterthur, Version 2016, 64 s
- [Ruppert–38] RUPPERT D., SIMPSON D. G.: *Comment on: Unmasking multivariate outliers and leverage points* Journal of the American Statistical Association, 85, 1990, 644-646.
- [Portnoy–39] PORTNOY S.: *Directions in Robust Statistics and Diagnostics: Part II* New York Springer-Verlag, 1991, 145-157.
- [Kulkarni–40] KULKARNI S.M., BORMANE D.S., NALBALWAR S.L.: *RANSAC Algorithm for Matching Inlier Correspondences in Video Stabilization* Societ for science and education, UK, 2017, Vil 5, Issue 1, ISSN 2054-7412 Dostupný z WWW: <http://dx.doi.org/10.14738/aivp.51.2692>

- [Traa–41] TRAA J, SMARAGDIS P.: *Multichannel Source Separation and Tracking with RANSAC and Directional Statistics* Dostupný z WWW: <http://dx.doi.org/10.1109/TASLP.2014.2365701>
- [Kaspi–42] KASPI O., YOSIPOF A.: *RANdom SAmples Consensus (RANSAC) algorithm for material-informatics: application to photovoltaic solar cells* Journal of Cheminformatics 2017, 2017; 9:34, DOI: 10.1186/s13321-017-0224-0
- [Kuba–43] KOUBA J., HEROUX P.: *Precise point positioning using IGS orbit and clock product* Geodetic Survey Division, Natural Resources Canada Dostupný z WWW: <http://igsceb.jpl.nasa.gov/components/usage.html>
- [Hofmann–44] HFMANN-WALLENHOF B., LICHTENEGGER H., WALSE E.: *GNSS-Global Navigation Systems* Springer-Verlag Wien 2008, ISBN 978-3-211-73012-6, 127 s
- bibitem[Cepek–45]cepek ČEPEK A.: *Informatika, Úvod do C++*, 1. vyd. Praha : Nakladatelství ČVUT, 2004. 265 s. ISBN 80-01-03074-1.
- [Demel–46] BAI Z., DEMMEL J., DONGARRA J., RUHE A., VORST H.: *Templates for the Solution of Algebraic Eigenvalue Problems*, [online], 2001, Dostupný z WWW: <http://www.cs.ucdavis.edu/~bai/ET/contents.html>
- [Rousseew–48] ROUSSEEUW P.: *Robust regression and outlier detection* Applied probability and statistics, 1987, ISBN 0471-85233-3
- [Fischler–49] FISCHLER M. A., BOLLES R. C.: *Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography*. Comm. of the ACM, Vol 24, pp 381-395, 1981.
- [Simpson–50] SIMPSON D.G., YOHAI V.J.: *Functional stability of one-step gm-estimators in approximately linear regression* The Annals of Statistics 1998, Vol. 26, No. 3, 1147–1169
- [XML–51] <http://www.applied-mathematics.net/tools/IXMLParser.html>

- [Kavouras–52] KAVOURAS M.: *On the detection of outliers and the determination of reliability in geodetic networks* Technical report NO. 87. University of New Brunswick 1982
- [Outliers–53] STAT 501 Regression Methods - online course materials website <https://onlinecourses.science.psu.edu/stat501/node/337> Copyright © 2017 The Pennsylvania State University
- [Connectivity–54] [https://en.wikipedia.org/wiki/Connectivity\\_\(graph\\_theory\)](https://en.wikipedia.org/wiki/Connectivity_(graph_theory))

## Zoznam skratiek

|        |                                    |
|--------|------------------------------------|
| LMS    | Least Median Squares               |
| ILU    | Incomplete Cholesky Decomposition  |
| GPL    | General Public License             |
| STL    | Standard Template Library          |
| GNU    | Unix-like Operating System         |
| MNČ    | Metóda najmenších čtverců          |
| GIS    | Geografic Information System       |
| GNSS   | Global Navigation Satellite System |
| GPS    | Global Positioning System          |
| CG     | Conjugate Gradients                |
| PCG    | Precoditioned Conjugate Gradients  |
| LTS    | Least Trimmed Squares              |
| RBP    | Resilient Backpropagation          |
| MSE    | Mean Squared Error                 |
| GSO    | Gramm-Schmidt Ortogonalization     |
| SVD    | Singular Value Decomposition       |
| IGS    | International GPS service          |
| PPP    | Precise Point Positioning          |
| XML    | Extensible Markup Language         |
| DFS    | Depth-first Search                 |
| MVCC   | Multiversion Concurrency Control   |
| PITR   | Point in Time Recovery             |
| SQL    | Structured Query Language          |
| RANSAC | Random Sample Consensus            |

MSS Minimal Sample Set

CS Consensus Set

VUGTK Výskumný ústav geodetický topografický a kartografický

# Zoznam príloh

|          |  |            |
|----------|--|------------|
| <b>A</b> | <b>Zápisníky geodetických meraní</b>                       | <b>207</b> |
| <b>B</b> | <b>Vybrané zdrojové kódy</b>                               | <b>209</b> |
| B.1      | Algoritmus združených gradientov s predpodmiením . . . . . | 209        |
| B.2      | Algoritmus ILU0 . . . . .                                  | 213        |
| B.3      | Funkcie súborových operácií . . . . .                      | 215        |

# A Zápisníky geodetických meraní

| ===== Observations ===== |      |              |             |           |            |
|--------------------------|------|--------------|-------------|-----------|------------|
| [from]                   | [to] | [s-distance] | [direction] | [z-angle] | [distance] |
| =====                    |      |              |             |           |            |
| 9001                     | 2001 | 33.5134      | 77.02196    | 93.64073  | 33.3463    |
| 9001                     | 2001 | 33.5146      | 277.022     | 306.35921 | 33.3475    |
| 9001                     | 2002 | 35.20971     | 287.98446   | 305.13542 | 35.0952    |
| 9001                     | 2002 | 35.20791     | 87.98489    | 94.86446  | 35.0934    |
| 9001                     | 2004 | 79.28005     | 123.30044   | 101.12989 | 79.2675    |
| 9001                     | 2004 | 79.28005     | 323.30065   | 298.87023 | 79.2675    |
| 9001                     | 2005 | 32.9815      | 335.42429   | 322.85975 | 30.8779    |
| 9001                     | 2005 | 32.981       | 135.42406   | 77.14016  | 30.8774    |
| 9001                     | 2006 | 30.5567      | 156.16933   | 74.88016  | 28.2086    |
| 9001                     | 2006 | 30.5588      | 356.16972   | 325.11926 | 28.2106    |
| 9001                     | 2007 | 37.72722     | 368.96383   | 319.97134 | 35.8859    |
| 9001                     | 2007 | 37.72512     | 168.96335   | 80.02837  | 35.8839    |
| 9001                     | 2008 | 37.63652     | 173.99006   | 80.2618   | 35.8419    |
| 9001                     | 2008 | 37.63852     | 373.98977   | 319.73822 | 35.8438    |
| 9001                     | 2010 | 55.12327     | 11.79037    | 310.38486 | 54.3914    |
| 9001                     | 2010 | 55.12327     | 211.79088   | 89.61589  | 54.3915    |
| 9001                     | 201  | 31.8908      | 133.3937    | 108.73893 | 31.5908    |
| 9001                     | 201  | 31.8894      | 333.39348   | 291.26064 | 31.5893    |
| 9001                     | 202  | 29.62099     | 345.3919    | 291.43444 | 29.3532    |
| 9001                     | 202  | 29.62279     | 145.39157   | 108.56566 | 29.3550    |
| 9001                     | 203  | 28.47439     | 159.5347    | 108.01018 | 28.249     |
| 9001                     | 203  | 28.47329     | 359.53446   | 291.98983 | 28.2481    |
| 9001                     | 2001 | 33.514       | 77.02246    | 93.64077  | 33.3469    |
| 9001                     | 2001 | 33.5145      | 277.02263   | 306.35885 | 33.3474    |
| 9001                     | 2002 | 35.20631     | 287.98505   | 305.1355  | 35.0918    |
| 9001                     | 2002 | 35.20491     | 87.98441    | 94.86475  | 35.0904    |
| 9001                     | 2004 | 79.27975     | 123.30021   | 101.13037 | 79.2672    |
| 9001                     | 2004 | 79.27985     | 323.30128   | 298.87003 | 79.2673    |
| 9001                     | 2005 | 32.9818      | 335.42437   | 322.86005 | 30.8781    |
| 9001                     | 2005 | 32.9809      | 135.42431   | 77.14059  | 30.8774    |
| 9001                     | 2006 | 30.557       | 156.16978   | 74.88031  | 28.2089    |
| 9001                     | 2006 | 30.5589      | 356.16936   | 325.12023 | 28.2106    |
| 9001                     | 2007 | 37.72702     | 368.96362   | 319.97056 | 35.8859    |
| 9001                     | 2007 | 37.72522     | 168.96382   | 80.0289   | 35.8841    |
| 9001                     | 2008 | 37.63642     | 173.98936   | 80.26147  | 35.8418    |
| 9001                     | 2008 | 37.63892     | 373.99014   | 319.73902 | 35.844     |
| 9001                     | 2010 | 55.12427     | 11.79043    | 310.38581 | 54.3923    |
| 9001                     | 2010 | 55.12257     | 211.79051   | 89.61567  | 54.390     |
| 9001                     | 201  | 31.8905      | 133.39475   | 108.73862 | 31.5905    |
| 9001                     | 201  | 31.8894      | 333.39388   | 291.26008 | 31.5893    |
| 9001                     | 202  | 29.62109     | 345.3916    | 291.43438 | 29.3533    |
| 9001                     | 202  | 29.62279     | 145.39151   | 108.56633 | 29.3550    |
| 9001                     | 203  | 28.47449     | 159.53513   | 108.0102  | 28.2493    |
| 9001                     | 203  | 28.47329     | 359.53439   | 291.98958 | 28.2481    |
| =====                    |      |              |             |           |            |

Kód A.1: Merané hodnoty



```

===== Points =====
[id]          [x]          [y]          [z]
=====
2001  1040799.2748  738841.3913  209.5454
2002  1040805.1813  738843.0552   209.04
2003  1040821.3558  738846.4995  205.1056
2004  1040857.9602  738842.4922  204.7959
2005  1040814.4011  738865.5741  217.7938
2006  1040811.9126  738875.1981  217.9508
2007  1040817.7351  738883.3087   217.845
2008  1040816.6234  738885.9116  217.6871
2009  1040814.4749  738895.6651  216.3706
  201  1040814.9094  738864.4595  201.8397
2010  1040812.4015  738917.4317  215.1551
  202  1040813.3587  738870.4123  202.2293
  203  1040811.6919  738876.6739  202.6301
9001  1040784.008   738871.03757    0
=====
  
```

Kód A.2: Súravnice bodov

## B Vybrané zdrojové kódy

Ukážky zdrojových kódov obmedzené na vybrané numerické algoritmy, ktoré boli použité v dizertačnej práci.

### B.1 Algoritmus združených gradientov s predpodmienením

```

1 #ifndef PRECONJUNGATE_GRADIENTS_H
2 #define PRECONJUNGATE_GRADIENTS_H
3
4 #include <smatrix.h>
5 #include <precondition.h>
6
7 template<typename Float=double, typename Index=std::size_t>
8 class PreConjugateGradients
9 {
10 public:
11
12     enum { noError, badMatrix, badDimension };
13
14     PreConjugateGradients(const GNU_gama::SparseMatrix<Float, Index>* Ap)
15     {
16         A = Ap;
17
18         eps      = Float(1e-12);
19         maxiter  = A->rows();
20
21         size     = A->rows();
22         itercount = Index();
23
24         lasterror = noError;
25
26         if (A->rows() != A->columns() || A->rows() <= Float())
27         {
28             lasterror = badMatrix;
29         }
30     }
31
32     ~PreConjugateGradients()
33     {
34     }
35
36     void setMax(int max) { maxiter = max; }
37     int  getMax() const { return maxiter; }
38     void setEps(int tol) { eps = tol; }
39     int  getEps() const { return eps; }
40     int  getIterations() const { return itercount; }
41
42     template<typename const_iterator, typename iterator>
43     int solve(const_iterator bbegin, const_iterator bend,
44              iterator          xbegin, iterator          xend )
45     throw()

```

```

46     {
47
48         if (size != Index(bend-bbegin) // size != Index(xend-xbegin) //
49             size <= Index())
50         {
51             lasterror = badDimension;
52         }
53         if (lasterror != noError) return lasterror;
54
55         Precondition<> prec(A);
56         prec.chol0();
57         L = prec.get_preconditioner();
58
59         Float* p = new Float[size];
60         Float* r = new Float[size];
61         Float* ap = new Float[size];
62         Float* z = new Float[size];
63
64         Float alfa, r0, beta, temp;
65
66         // initialization for residuals r = b - A*x
67         // vector x is not initialized (not set to zero vector!)
68
69         r0 = Float();
70         for (Index j=Index(), k=1; k<=size; k++, j++)
71         {
72             Float* n = A->begin(k);
73             Float* e = A->end (k);
74
75             Float t = *(bbegin + j);
76             for(Index* i=A->ibegin(k) ; n!=e; n++, i++)
77             {
78                 t -= *n * *(xbegin + *i - 1);
79             }
80
81             z[j] = r[j] = t;
82             r0 += t*t;
83         }
84
85         // initialization for precondition z = inv(M) - r, M=LU
86         // initialization for direction p = z
87
88         //Forward substitution column version
89
90         for(Index k=1; k<=size; k++)
91         {
92             Float* n=L->begin(k);
93             Float* e=L->end(k);
94             Index* i=L->ibegin(k);
95
96             for( ; n!=e; n++, i++)
97             {
98                 if(*i==k)
99                 {
100                     break;
101                 }
102             }
103
104             z[k-1] /= *n;

```

```

105
106         n++; i++;
107
108         for( ; n!=e; n++, i++)
109         {
110             z[*i-1] -= z[k-1] * *n;
111         }
112     }
113
114     //Backward substitution
115
116     for (Index k=size; k>0; k--)
117     {
118         Float* e=L->end(k)-1;
119
120         Float t = z[k-1];
121
122         for(Index* i=L->iend(k)-1 ; *i!=k ; e--, i--)
123         {
124             t -= *e * z[*i-1];
125         }
126
127         p[k-1] = z[k-1] = t / *e;
128     }
129
130     // main loop
131
132     itercount = Index();
133     for(Index i=Index(); i<maxiter; i++, itercount++)
134     {
135         temp = dot_product(r,z);
136
137         if(temp <= r0*eps) break;
138
139         // axp = A*p
140         Float* axp = ap;
141         for (Index k=1; k<=size; k++, axp++)
142         {
143             Float* n = A->begin(k);
144             Float* e = A->end (k);
145
146             *axp = Float();
147             for(Index* i=A->ibegin(k) ; n!=e; n++, i++)
148             {
149                 *axp += *n * p[*i - 1];
150             }
151         }
152
153         alfa = temp/dot_product(p,ap);
154
155         for(Index i=Index(); i<size; i++)
156         {
157             *(xbegin+i) += alfa * p[i];
158             z[i] = r[i] -= alfa * ap[i];
159         }
160
161         for(Index k=1; k<=size; k++)
162         {
163             Float* n=L->begin(k);

```

```

164         Float* e=L->end(k);
165         Index* i=L->ibegin(k);
166
167         for( ; n!=e; n++, i++)
168         {
169             if(*i==k)
170             {
171                 break;
172             }
173         }
174
175         z[k-1] /= *n;
176
177         n++; i++;
178
179         for( ; n!=e; n++, i++)
180         {
181             z[*i-1] -= z[k-1] * *n;
182         }
183     }
184
185     for (Index k=size; k>0; k--)
186     {
187         Float* e=L->end(k)-1;
188
189         Float t = z[k-1];
190
191         for(Index* i=L->iend(k)-1 ; *i!=k ; e--, i--)
192         {
193             t -= *e * z[*i-1];
194         }
195
196         z[k-1] = t / *e;
197     }
198
199     beta = dot_product(r,z)/temp;
200
201     for(Index i=Index(); i<size; i++)
202     {
203         p[i] = z[i] + beta * p[i];
204     }
205 }
206
207 delete[] p;
208 delete[] r;
209 delete[] ap;
210 delete[] z;
211
212     return lasterror;
213 }
214
215 private:
216
217     const GNU_gama::SparseMatrix<Float,Index>* A;
218     GNU_gama::SparseMatrix<Float,Index>* L;
219
220     Index size;
221     Index itercount;
222     Float eps;

```

```

223     Index  maxiter;
224     int    lasterror;
225
226     Float dot_product(const Float* a, const Float* b)
227     {
228         Float dp = Float();
229         for (Index i=Index(); i<size; i++) dp += a[i]*b[i];
230         return dp;
231     }
232 };
233
234 #endif // CONJUGATE_GRADIENTS_H

```

Kód B.1: Algoritmus združených gradientov s predpodmiením

## B.2 Algoritmus ILU0

```

1  #ifndef PRECONDITION_H
2  #define PRECONDITION_H
3
4  #include <smatrix.h>
5  #include <cmath>
6
7  using namespace GNU_gama;
8
9  template<typename Float=double, typename Index=std::size_t>
10 class Precondition
11 {
12 public:
13
14     Precondition(const GNU_gama::SparseMatrix<Float,Index>* Ap)
15     {
16         A   = Ap;
17         L   = Ap->replicate();
18
19         dim = Ap->rows();
20         nz  = Ap->nonzeroes()/2+dim;
21     }
22
23     ~Precondition(){}
24
25     // Matrix A is sorted !
26
27     Index chol0()
28     {
29         Index dim_alfa = Index();
30         Float d = Float();
31
32         for(Index i = 1; i <= dim; i++)
33         {
34             Float* n = L->begin(i);
35             Float* e = L->end(i);
36             Index* j = L->ibegin(i);
37             Index* je = L->iend(i);
38

```

```

39         dim_alfa = e-n;
40
41         Float* alfa = new Float[dim_alfa];
42
43         // Finding pointer to diagonal element
44
45         for(Index* ii=L->ibegin(i); n!=e; n++, ii++, j++)
46         {
47             if(*ii==i)
48             {
49                 break;
50             }
51         }
52
53         // Diagonal element cant be equal to zero or lower
54
55         if(*n > 1e-12)
56         {
57             d = std::sqrt(*n);
58
59             *n = d; n++; j++;
60
61             for(Index l=0; n!=e; n++, l++)
62             {
63                 *n /= d;
64                 alfa[l] = - *n;
65             }
66
67             for(Index l=0; j!=je; j++, l++)
68             {
69                 // Linear combination of vectors v = alfa*a+b
70
71                 Float* n1 = L->begin(i);
72                 Float* e1 = L->end(i);
73
74                 Float* n2 = L->begin(*j);
75                 Float* e2 = L->end(*j);
76
77                 Index* i1 = L->ibegin(i);
78                 Index* i2 = L->ibegin(*j);
79
80                 while(true)
81                 {
82                     if(*i1<*i2)
83                     {
84                         i1++; n1++;
85                     }
86                     else if(*i1>*i2)
87                     {
88                         i2++; n2++;
89                     }
90                     else
91                     {
92                         *n2 += alfa[l] * *n1;
93
94                         n1++; n2++; i1++; i2++;
95                     }
96
97                     if(n1==e1 || n2==e2)

```

```

98         {
99             break;
100        }
101    }
102    }
103    }
104    delete[] alfa;
105    }
106    return 0;
107    }
108
109    SparseMatrix<>* get_preconditioner() const
110    {
111        return L;
112    }
113
114 private:
115
116    const GNU_gama::SparseMatrix<>* A;
117    GNU_gama::SparseMatrix<>* L;
118
119    Index dim;
120    Index nz;
121
122 };
123
124 #endif // PRECONDITION_H

```

Kód B.2: Algoritmus združených gradientov s predpodmieniením

## B.3 Funkcie súborových operácií

```

1
2 void check(char* subor,int &m, int &n, int &nonzero){
3
4     ifstream dat(subor);           //otvori subor
5     if(!dat)
6     {
7         cerr << "Can't read input matrix";
8     }
9
10    double d;
11    m = n = nonzero = 0;
12
13    dat >> m >> n;
14
15    while(dat >> d)
16    {
17        if(d != 0.0)
18        {
19            nonzero++;
20        }
21    }
22    dat.close();
23 }

```



```

24
25 void check_sparse(const std::string subor, int &m, int &n, int &nonzero)
26 {
27
28     ifstream dat(subor.c_str());           //otvori subor
29     if(!dat)
30     {
31         cerr << "Can't read input matrix";
32     }
33     m = n = nonzero = 0;
34
35     dat >> n >> m >> nonzero;
36
37     dat.close();
38 }
39
40 std::istream& operator>>(std::istream &data, SparseMatrix<> *A)
41 {
42     double val;
43     int m,n;
44     data >> m >> n;
45
46     for(int i=1; i<=m; i++)
47     {
48         A->new_row();
49         for(int j=1; j<=n; j++)
50         {
51             data >> val;
52
53             if(val != 0.0)
54             {
55                 A->add_element(val, j);
56             }
57         }
58     }
59     return data;
60 }
61
62 std::ostream& operator<<(std::ostream &data, SparseMatrix<> *A)
63 {
64     data << std::endl;
65     for (unsigned long k=1; k<=A->rows(); k++)
66     {
67         data << k << " : ";
68
69         double* n = A->begin(k);
70         double* e = A->end (k);
71
72         for(std::size_t* i=A->ibegin(k) ; n!=e; n++, i++)
73         {
74             data << *n << " [" << *i << "]" ";
75         }
76         data << std::endl;
77     }
78     return data;
79 }
80
81 std::istream& read_sparse(istream &data, SparseMatrix<> *A)
82 {

```

```
83     int m,n,nonzero;
84     int i,j;
85     double val;
86
87     int j0=0;
88
89     data >> m >> n >> nonzero;
90
91     while(data >> i >> j >> val)
92     {
93         if(j!=j0)
94         {
95             A->new_row();
96             j0=j;
97         }
98         A->add_element(val,i);
99     }
100     return data;
101 }
```

Kód B.3: Funkcie súborových oprácií