

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ  
Fakulta stavební  
Katedra mapování a kartografie

# DIPLOMOVÁ PRÁCE

VÝPOČET PŘIBLIŽNÝCH SOUŘADNIC BODŮ V C++

červen 1999

Ing. Jiří Veselý

Tímto bych chtěl poděkovat vedoucímu diplomové práce Doc.Ing. Aleši Čepkovi, CSc. za cenné rady a připomínky v průběhu mé práce, poskytnutou literaturu a za pomoc při konečných úpravách tohoto textu.

Prohlašuji, že jsem diplomovou práci vypracoval samostatně s použitím literatury uvedené v seznamu.

Jiří Veselý

# Obsah

Úvod	4
<b>1 Výpočet polohy bodu</b>	<b>5</b>
1.1 Základní typy bodů a měření	5
1.1.1 Určující prvky	5
1.1.2 Orientační posun osnovy vodorovných směrů	6
1.2 Geodetické úlohy	7
1.2.1 Určení parametrů kružnice z dvou bodů a obvodového úhlu	8
1.2.2 Protínání vpřed ze směrů	9
1.2.3 Protínání z délek	11
1.2.4 Průsečík orientované polopřímky s kružnicí	13
1.2.5 Spojené protínání	14
1.2.6 Protínání zpět z délky a úhlu	15
1.2.7 Protínání zpět	15
1.3 Podobnostní transformace v rovině	16
<b>2 Návrh tříd pro výpočet souřadnic bodu</b>	<b>19</b>
2.1 Programovací jazyk C++	19
2.2 Třídy	20
2.3 Knihovna Median	20
2.3.1 Základní datové typy	20
2.3.2 GeodetickaUloha	21
2.3.3 Kruznice	22
2.3.4 Delka_delka	23
2.3.5 Smer_smer	23

---

2.3.6	Smer_delka . . . . .	24
2.3.7	Smer_uhel . . . . .	24
2.3.8	Delka_uhel . . . . .	25
2.3.9	Uhel_uhel . . . . .	25
2.3.10	Výběr řešení . . . . .	26
2.3.11	Určení mediánu souřadnic . . . . .	27
2.3.12	Transformace souřadnic . . . . .	27
2.3.13	Výpočet přibližných souřadnic jednoho bodu . . . . .	29
2.3.14	Výpočet přibližných souřadnic seznamu zadaných bodů . . . . .	30
<b>3</b>	<b>Praktické použití třídy Median</b>	<b>34</b>
3.1	Formát vstupních dat . . . . .	34
3.2	Příklady . . . . .	36
	<b>Literatura</b>	<b>38</b>
<b>A</b>	<b>Ukázkové příklady</b>	<b>39</b>
A.1	Hansenova úloha . . . . .	40
A.2	Volná síť . . . . .	43
A.3	Vázaná síť . . . . .	48
A.4	Vázaná síť – zhoršená konfigurace . . . . .	53
<b>B</b>	<b>Hlavičkové soubory knihovny Median</b>	<b>72</b>
B.1	zmeny.h . . . . .	73
B.2	vyjimky.h . . . . .	74
B.3	g_fce.h . . . . .	75
B.4	g_ulohy.h . . . . .	80
B.5	pribl_b.h . . . . .	86
B.6	pribl_s.h . . . . .	89

# Seznam obrázků

1.1	Orientační posun . . . . .	7
1.2	Určení parametrů kružnice . . . . .	9
1.3	Protínání ze směrnic . . . . .	10
1.4	Protínání z délek . . . . .	11
1.5	Průsečík orientované polopřímky s kružnicí . . . . .	13
1.6	Spojené protínání . . . . .	14
1.7	Protínání zpět z délky a úhlu . . . . .	15
1.8	Protínání zpět . . . . .	16
1.9	Podobnostní transformace v rovině . . . . .	17
A.1	Hansenova úloha . . . . .	42
A.2	Volná síť . . . . .	47
A.3	Vázaná síť . . . . .	52
A.4	Vázaná síť – zhoršená konfigurace . . . . .	71

# Úvod

V současné době je při zpracování výsledků geodetických měření kladen stále větší důraz na automatizaci. Díky modernímu přístrojovému vybavení není výjimkou nadbytečný počet měření pro určení polohy bodů. Tato měření vstupují do procesu vyrovnání spolu s přibližnými souřadnicemi bodů. Cílem programátora je vytvořit takové algoritmy, které by umožňovaly projít celým procesem vyrovnání s minimem zásahů od uživatele. K tomu je zapotřebí získat přibližné souřadnice bodů, které se použijí jako vstupní hodnoty pro další výpočty.

V naší republice je dlouhá tradice automatizovaného zpracování dat, především zásluhou Ing. Františka Charamzy, CSc., dlouholetého vědeckého pracovníka VÚGTK. Rád bych připomněl jeho práci v oblasti využití Gramm-Schmidtovy ortogonalizace a ve vyrovnávacím počtu.

Předkládaná diplomová práce navazuje na jednu z dílčích úloh, které byly řešeny při tvorbě programového vybavení počítačů SMEP - jedná se o výpočet přibližných souřadnic bodů. Tato práce se snaží uvedený problém řešit prostředky objektového programování, které v době vzniku původního programového vybavení (70. léta) nebylo k dispozici.

Vzhledem k tomu, že na naší fakultě se stále učí jednotlivé typy „protínání“ jako samostatné úlohy, věřím, že mnou předložená práce naznačí poněkud odlišný přístup k algoritmizaci výpočtu souřadnic bodu.

# Kapitola 1

## Výpočet polohy bodu

V této kapitole se budu zabývat výpočtem polohy bodu z měřených (a následně upravených) hodnot. Do výpočtu vstupují měření po všech potřebných redukcích, přičemž problematika úprav měřených veličin není předmětem této práce. V dalším budou uvažovány pouze výpočty v rovině (souřadnice  $[Y ; X]$ ).

### 1.1 Základní typy bodů a měření

Pro účely výpočtu přibližných souřadnic lze body rozdělit do dvou skupin:

- **dané** – body, jejichž souřadnice jsou známy,
- **určované** – body, jejichž přibližné souřadnice se počítají.

Při měření v terénu se zjišťují šikmé délky, zenitové vzdálenosti a osnova vodorovných směrů. Po aplikování příslušných fyzikálních a matematických redukcí získáme *vodorovnou délku v rovině zobrazení* (viz např. [1]). Obecně tedy vstupují do výpočtu vodorovné délky a vodorovné směry; třetí – zřídka se vyskytující – možností je přímo měřený vodorovný úhel.

#### 1.1.1 Určující prvky

Pod určujícími prvky rozumíme reprezentaci měřených veličin, vstupující přímo do výpočtu polohy. Pro určení souřadnic lze použít libovolnou kombinaci dvou prvků z následujících tří skupin [2]:

- **vnější směrník [s]** – orientovaná polopřímka s počátkem v *daném* bodě a procházející *určovaným* bodem,

- **délka [d]** – délka úsečky, spojující *daný* a *určovaný* bod,
- **vnitřní úhel [u]** – úhel s vrcholem v *určovaném* bodě a rameny procházejícími *danými* body.

Vnější směrník se vypočte podle vztahu

$$\sigma_{ij} = \psi_{ij} + z_i \quad (1.1)$$

kde

- $\sigma_{ij}$  – vnější směrník z bodu  $i$  na bod  $j$ ,
- $\psi_{ij}$  – měřený směr z bodu  $i$  na bod  $j$ ,
- $z_i$  – orientační posun na bodě  $i$  (viz kap. 1.1.2).

Vnitřní úhly se vypočtou z osnovy vnitřních směrů

$$u_{ij} = \psi_i - \psi_j \quad i \in \langle 1; n-1 \rangle \quad j \in \langle i+1; n \rangle \quad (1.2)$$

kde

- $n$  – počet vnitřních směrů.

### 1.1.2 Orientační posun osnovy vodorovných směrů

Pro výpočet hodnoty vnějšího směrníku je nutné znát velikost orientačního posunu na bodě  $P_0$ , což je úhel, který svírá nulový směr stroje s rovnoběžkou s osou  $X$ . K řešení je nutné znát souřadnice stanoviště  $P_0$ , souřadnice alespoň jednoho bodu  $P_i$  se známými souřadnicemi a směr na něj měřený  $\psi_{0i}$ .

Pokud jsou měřeny směry na více ( $n$ ) daných bodů, může dojít k *vyrovnání* osnovy podle požadavků MNČ (viz např. [3]). Pokud označíme  $\sigma_{0i}$  za vypočtený směrník spojnice  $P_0 - P_i$  a  $z$  za výsledný orientační posun (obr. 1.1), bude oprava pro  $i$ -tý směr

$$v_i = (\sigma_{0i} - \psi_{0i}) - z. \quad (1.3)$$

Úkolem vyrovnání je nalézt takový orientační posun  $z$ , pro který platí

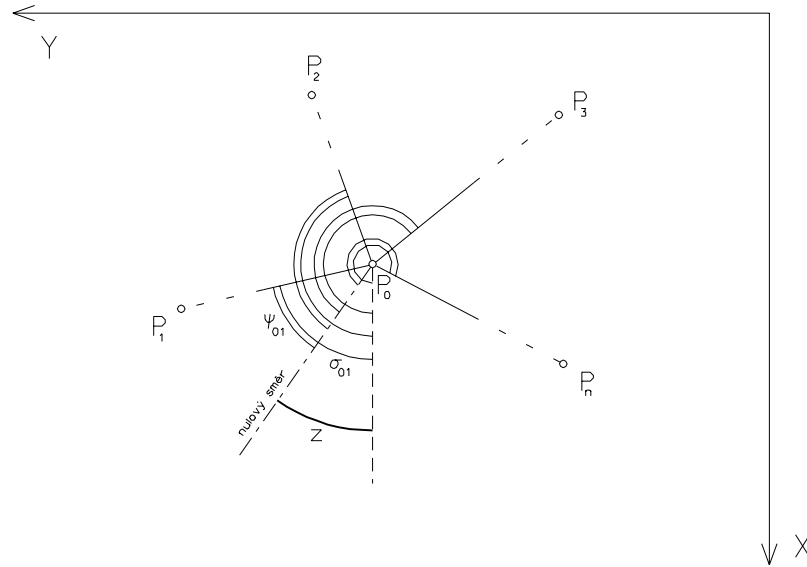
$$[p v] = \min.$$

Odvozením lze dokázat, že řešením je obecný aritmetický průměr a jeho jednotková střední chyba

$$z = \frac{[p(\sigma_{0i} - \psi_{0i})]}{[p]}, \quad (1.4)$$

$$m_0 = \sqrt{\frac{[p v]}{n-1}}. \quad (1.5)$$





Obrázek 1.1: Orientační posun

Při vypuštění  $j$ -tého orientačního bodu se změní orientační posun a jeho jednotková střední chyba podle vztahů [2]

$$z' = \left( z - \frac{p_j(\sigma_{0j} - \psi_{0j})}{[p]} \right) \frac{[p]}{[p] - p_j}, \quad (1.6)$$

$$m'_0 = \sqrt{\frac{[p v v] - \Delta_j}{n - 2}} \quad (1.7)$$

kde

$$\Delta_j = \frac{[p] p_j v_j^2}{[p] - p_j}$$

Při praktických výpočtech lze použít pro odhad výsledné hodnoty také medián orientačních posunů, vypočtených z jednotlivých směrniců. Medián je robustní odhad (na rozdíl od průměru), protože není příliš ovlivněn jednou nebo i více vybočujícími hodnotami („hrubými chybami“).

## 1.2 Geodetické úlohy

V závislosti na kombinaci určujících prvků se pro výpočet souřadnic použijí následující metody protínání:

- **protínání vpřed ze směrniců [s-s]** – dva vnější směrniců (kap. 1.2.2),

- **protínání z délky a směrníku [s-d]** – vnější směrník a délka; zvláštním případem je polární metoda (kap. 1.2.4),
- **spojené protínání [s-u]** – vnější směrník a vnitřní úhel; zvláštním případem je protínání stranou (kap. 1.2.5),
- **protínání z délek [d-d]** – dvě délky (kap. 1.2.3),
- **protínání zpět z délky a úhlu [d-u]** – délka a vnitřní úhel (kap. 1.2.6),
- **protínání zpět [u-u]** – dva vnitřní úhly (kap. 1.2.7).

Každá z úloh může mít jednoznačné řešení, dvojnásobné řešení (např. u protínání z délek) nebo žádné (případně nejisté – malý úhel protnutí a pod.).

Úlohy lze řešit pomocí rozkladu na základní výpočetní operace, což je

- určení parametrů kružnice z dvou bodů na obvodu a obvodového úhlu (kap. 1.2.1)
- protínání vpřed ze směrníků (kap. 1.2.2)
- protínání z délek - průsečík dvou kružnic (kap. 1.2.3)
- průsečík orientované polopřímky s kružnicí (kap. 1.2.4)

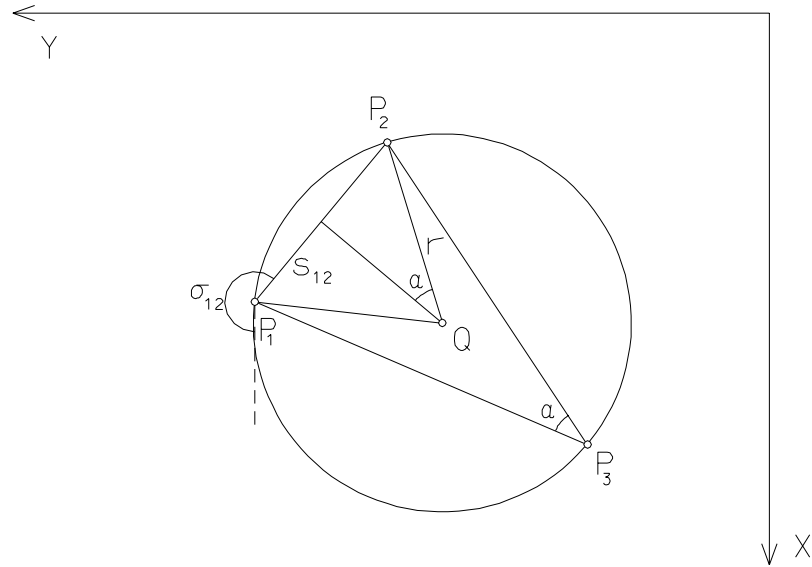
#### Dále použitá symbolika:

daný bod	P $[y_P; x_P]$
hledaný bod	Q $[y_Q; x_Q]$
vnitřní směr	$\psi$
vnější směrník	$\sigma$
délka	d
vnitřní úhel	u

### 1.2.1 Určení parametrů kružnice z dvou bodů a obvodového úhlu

Tato úloha řeší výpočet parametrů kružnice (poloměr  $r$  a střed  $S$ ) ze známých souřadnic bodů na kružnici  $P_1[y_1; x_1]$  a  $P_2[y_2; x_2]$  a obvodového úhlu nad těmito body  $\alpha$ . Úloha nemá řešení, jestliže jsou dané body totožné ( $s_{12} = 0$ ). Nejisté řešení nastává, pokud se úhel  $\alpha$  liší od celočíselného násobku  $\pi$  o méně než 10 gon:

$$|\sin \alpha| < 0,15 \quad (1.8)$$



Obrázek 1.2: Určení parametrů kružnice

Podle obr. 1.2 platí vztahy

$$r' = \frac{s_{12}}{2 \sin \alpha}, \quad (1.9)$$

$$y_Q = y_1 + r' \sin\left(\sigma_{12} + \frac{\pi}{2} - \alpha\right) = y_1 + r' \cos(\sigma_{12} - \alpha), \quad (1.10)$$

$$x_Q = x_1 + r' \cos\left(\sigma_{12} + \frac{\pi}{2} - \alpha\right) = x_1 - r' \sin(\sigma_{12} - \alpha), \quad (1.11)$$

$$r = |r'| \quad (1.12)$$

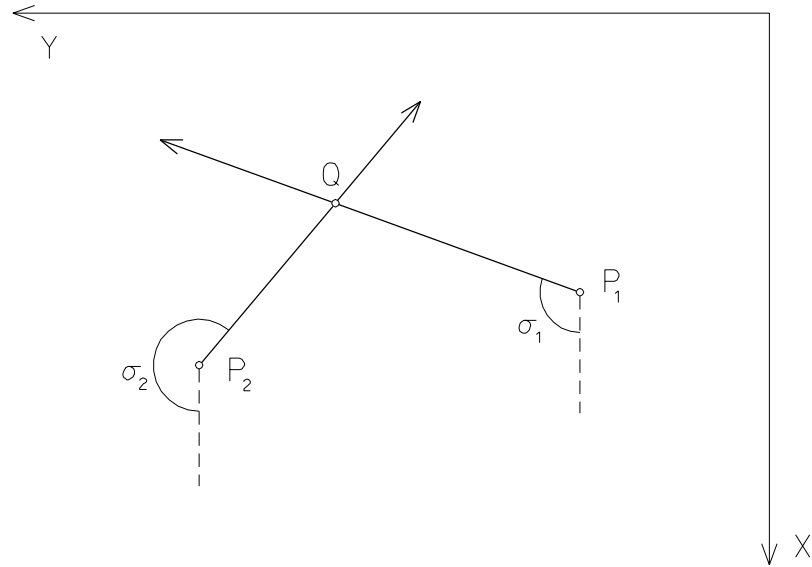
### 1.2.2 Protínání vpřed ze směrniců

Pomocí této úlohy se řeší výpočet polohy bodu ze směrniců určených na dvou daných bodech  $P_1[y_1; x_1]$ ,  $P_2[y_2; x_2]$  (obr. 1.3). Singulární řešení nastává tehdy, jestliže se směry neprotínají, protínají v některém z daných bodů nebo jsou dané body totožné.

Při odvození lze vyjít z rovnice přímky v rovině  $y = k \cdot x + q$ . Pro bod  $Q$  tedy platí

$$y_Q = k_1 \cdot x_Q + q_1, \quad (1.13)$$

$$y_Q = k_2 \cdot x_Q + q_2 \quad (1.14)$$



Obrázek 1.3: Protínání ze směrniců

a zároveň

$$y_1 = k_1 \cdot x_1 + q_1, \quad (1.15)$$

$$y_2 = k_2 \cdot x_2 + q_2 \quad (1.16)$$

kde

$$k_i = \tan \sigma_i$$

Úpravou lze odvodit vztahy

$$x_Q = \frac{\cos \sigma_1 \cos \sigma_2 (y_2 - y_1) + \sin \sigma_1 \cos \sigma_2 \cdot x_1 - \sin \sigma_2 \cos \sigma_1 \cdot x_2}{\sin \sigma_1 \cos \sigma_2 - \sin \sigma_2 \cos \sigma_1}, \quad (1.17)$$

$$y_Q = y_1 + \frac{\sin \sigma_1 \cos \sigma_2 (y_2 - y_1) - \sin \sigma_1 \sin \sigma_2 (x_2 - x_1)}{\sin \sigma_1 \cos \sigma_2 - \sin \sigma_2 \cos \sigma_1} \quad (1.18)$$

Tyto vztahy lze dále zjednodušit, např. podle [2] se pro výpočet volí indexy daných bodů tak, že platí nerovnost

$$|\sin \sigma_1| > |\sin \sigma_2| \quad (1.19)$$

a souřadnice určovaného bodu se poté určí podle vztahů:

$$y_Q = y_1 + \Delta Y, \quad x_Q = x_1 + \frac{\Delta Y \cos \sigma_1}{\sin \sigma_1} \quad (1.20)$$

kde

$$\Delta Y = \frac{\sin \sigma_2 \sin \sigma_1 (x_2 - x_1) - \cos \sigma_2 \sin \sigma_1 (y_2 - y_1)}{\cos \sigma_2 \sin \sigma_1 - \sin \sigma_2 \cos \sigma_1} \quad (1.21)$$

Singulární řešení nastává při splnění alespoň jedné z podmínek

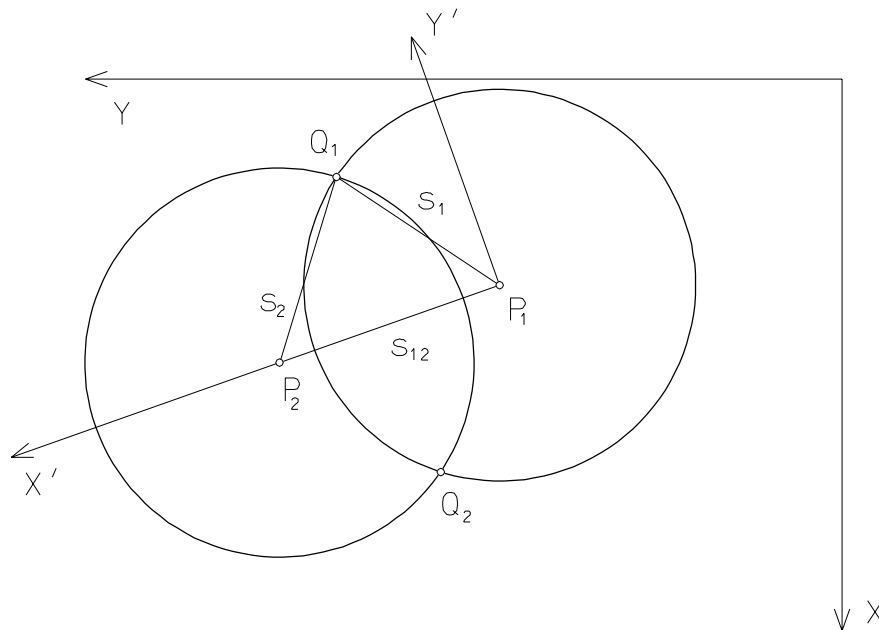
$$\begin{aligned} \text{sign}(\Delta Y) &\neq \text{sign}(\sin \sigma_1), \\ \text{sign}(x_Q - x_2) &\neq \text{sign}(\cos \sigma_2). \end{aligned} \quad (1.22)$$

Nejisté protnutí (úhel protnutí menší než cca 10 gon) nastává při splnění nerovnosti

$$|\cos \sigma_2 \sin \sigma_1 - \sin \sigma_2 \cos \sigma_1| < 0,15 \quad (1.23)$$

### 1.2.3 Protínání z délek

Protínání z délek je vlastně řešením průsečíku dvou kružnic, u nichž jsou známy jak poloměry (měřené délky  $s_1$  a  $s_2$ ), tak poloha středů (dané body  $P_1 [y_1; x_1]$  a  $P_2 [y_2; x_2]$ ). Pro výpočet se použije řešení v pomocné souřadnicové soustavě, která má počátek v bodě  $P_1$  a osu  $+X'$  ve spojnici bodů  $P_1$  a  $P_2$ .



Obrázek 1.4: Protínání z délek

Podle obr. 1.4 platí vztahy

$$s_1^2 = y_Q'^2 + x_Q'^2, \quad (1.24)$$

$$s_2^2 = y_Q'^2 + (s_{12} - x_Q')^2 \quad (1.25)$$

kde

$$s_{12} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (1.26)$$

Po dosazení  $y_Q'$  ze vztahu (1.24) do (1.25) získáme souřadnici  $x_Q'$

$$x_Q' = \frac{s_1^2 - s_2^2 + s_{12}^2}{2s_{12}} \quad (1.27)$$

Pro  $y_Q'$  se použije zpětné dosazení do (1.24)

$$y_Q' = \sqrt{s_1^2 - x_Q'^2} \quad (1.28)$$

Takto vypočtený bod Q' je jedním z dvou řešení (obecně), druhé řešení je symetrické k ose X'. Konečné souřadnice získáme transformací z místní souřadnicové soustavy

$$y_{Q1,2} = y_1 + \frac{y_2 - y_1}{s_{12}} x_Q' \pm \frac{x_2 - x_1}{s_{12}} y_Q', \quad (1.29)$$

$$x_{Q1,2} = x_1 + \frac{x_2 - x_1}{s_{12}} x_Q' \mp \frac{y_2 - y_1}{s_{12}} y_Q'. \quad (1.30)$$

Úloha nemá řešení při splnění těchto podmínek

$$s_1 \leq 0 \vee s_2 \leq 0 \quad \text{záporná nebo nulová měřená délka,}$$

$$s_{12} = 0 \quad \text{dané body jsou totožné,}$$

$$(s_1 + s_2) < s_{12} \quad \text{průsečík neexistuje.}$$

Pro  $(s_1 + s_2) = s_{12}$  existuje pouze jeden průsečík (dotykový bod).

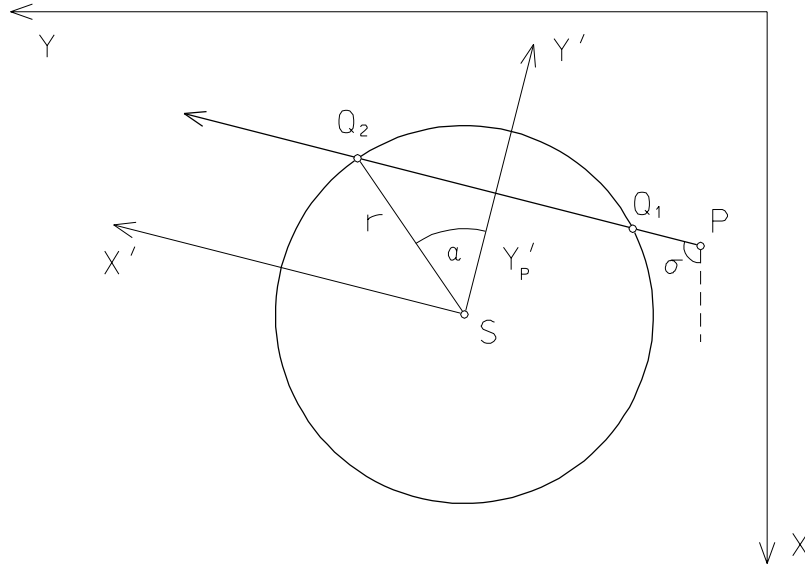
Nejisté protnutí (úhel protnutí menší než cca 10 gon) je detekováno splněním nerovnosti [2]

$$\frac{y_Q'}{s_{12}} < 0, 15s_1s_2 \quad (1.31)$$

### 1.2.4 Průsečík orientované polopřímky s kružnicí

Tato úloha řeší případ průsečíku orientované polopřímky dané souřadnicemi počátečního bodu  $P[x_P, y_P]$  a směrníkem  $\sigma$  s kružnicí, danou souřadnicemi středu  $S[x_S, y_S]$  a poloměrem  $r$ . Pokud jsou body  $P$  a  $S$  totožné, jedná se o polární metodu.

Úloha nemá řešení, pokud polopřímka míjí kružnici. V opačném případě má úloha jedno nebo dvě řešení.



Obrázek 1.5: Průsečík orientované polopřímky s kružnicí

Princip výpočtu spočívá v použití pomocné souřadnicové soustavy s počátkem ve středu kružnice  $S$  a s poloosou  $+X'$  orientovanou rovnoběžně s danou polopřímkou (obr. 1.5). V této soustavě budou mít body  $P$  a  $Q$  souřadnice

$$y'_P = (y_P - y_S) \cos \sigma - (x_P - x_S) \sin \sigma, \quad (1.32)$$

$$x'_P = (x_P - x_S) \cos \sigma + (y_P - y_S) \sin \sigma, \quad (1.33)$$

$$y'_{Q_1} = y'_{Q_2} = y'_P, \quad (1.34)$$

$$x'_{Q_1} = \sqrt{r^2 - y'^2_P}, \quad (1.35)$$

$$x'_{Q_2} = -x'_{Q_1}. \quad (1.36)$$

Na závěr se provede transformace souřadnic bodů  $Q_1$  a  $Q_2$  z místní souřadnicové soustavy

$$\begin{aligned} y_{Q_i} &= y_S + y'_{Q_i} \cos \sigma + x'_{Q_i} \sin \sigma, \\ x_{Q_i} &= x_S + x'_{Q_i} \cos \sigma - y'_{Q_i} \sin \sigma \end{aligned} \quad i = 1, 2. \quad (1.37)$$

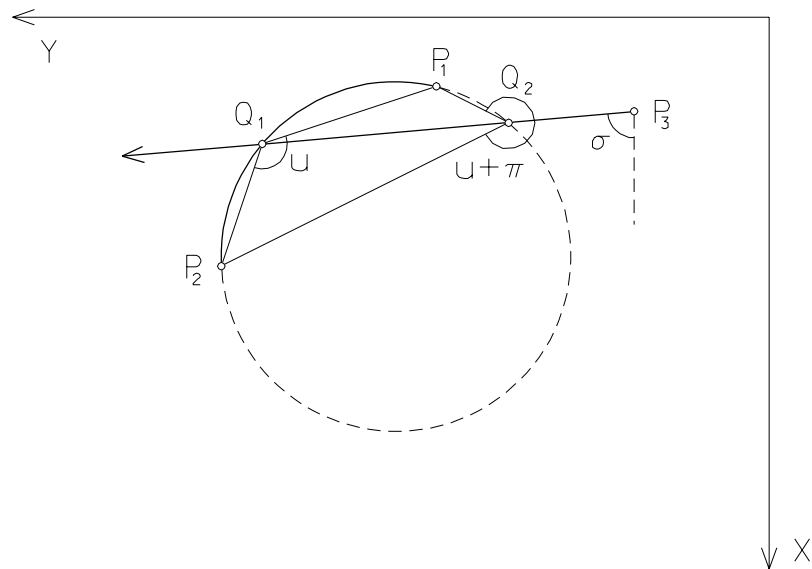
Singulární řešení lze identifikovat pomocí následujících podmínek

$$\begin{aligned} r &\leq 0 && \text{daný poloměr není kladný,} \\ |y'_P| &> r && \text{polopřímka mívá kružnici - průsečík neexistuje,} \\ x'_{Q_1} &\leq x'_P && \text{polopřímka mívá kružnici, průsečík neexistuje,} \end{aligned}$$

Pokud nejsou tyto podmínky splněny a zároveň platí nerovnost  $x'_{Q_2} \leq x'_P$ , má úloha pouze jedno řešení.

### 1.2.5 Spojené protínání

Při této úloze jsou dány souřadnice bodů  $P_1$ ,  $P_2$  a  $P_3$ , vnitřní úhel  $u$  z určovaného bodu  $Q$  na body  $P_1$  a  $P_2$  a vnější směrnik  $\sigma$  na bodě  $P_3$  (obr. 1.6).



Obrázek 1.6: Spojené protínání

Pomocí úlohy *Parametry kružnice* (kap. 1.2.1) se určí střed a poloměr kružnice dané body  $P_1$  a  $P_2$  a obvodovým úhlem  $u$ . Dále se vypočtou průsečíky této kružnice s orientovanou polopřímkou určenou bodem  $P_3$  a směrníkem  $\sigma$  (kap. 1.2.4). Řešením úlohy jsou ty průsečíky, mající obvodový úhel rovný  $u$ . Na obr. 1.6 představuje bod  $Q_1$  řešení; bod  $Q_2$  řešením není, neboť obvodový úhel se od hodnoty  $u$  liší o  $\pi$ .

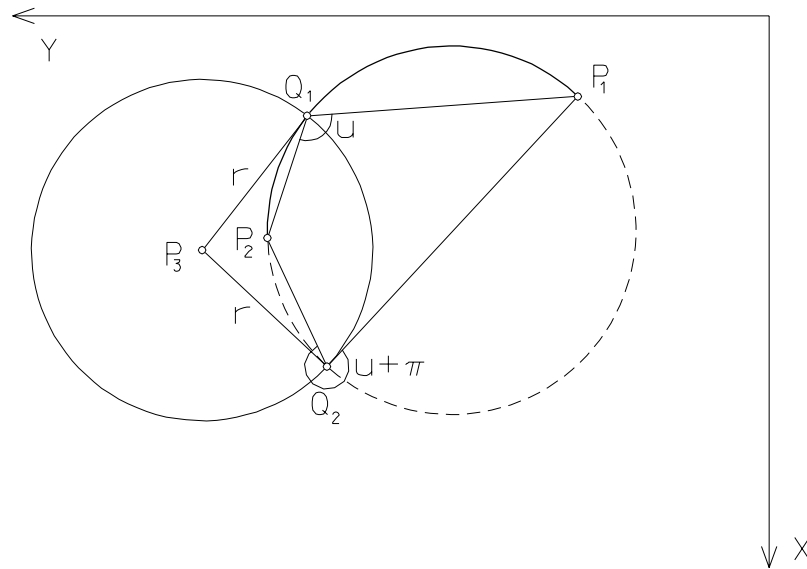


Zvláštním případem je protínání stranou, kdy bod  $P_3$  je totožný s jedním z bodů  $P_1$  nebo  $P_2$ .

Singulární řešení nastává za podmínek definovaných jednotlivými dílčími úlohami – např. malý obvodový úhel při určení parametrů kružnice.

### 1.2.6 Protínání zpět z délky a úhlu

Jsou dány souřadnice bodů  $P_1$ ,  $P_2$  a  $P_3$ , vnitřní úhel  $u$  z určovaného bodu  $Q$  na body  $P_1$  a  $P_2$  a délka  $d$  mezi body  $P_3$  a  $Q$  (obr. 1.7).



Obrázek 1.7: Protínání zpět z délky a úhlu

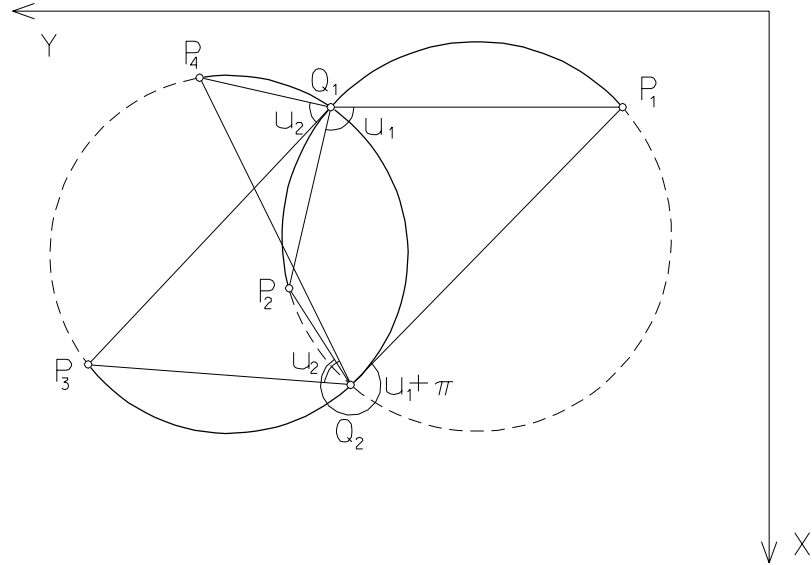
Podle kap. 1.2.1 se vypočtou parametry kružnice  $k_1(P_1, P_2, u)$  a úloha se převede na výpočet protínání z délek. Řešením je průsečík, jehož obvodový úhel má velikost  $u$ . Z řešení jsou vyloučeny průsečíky s vrcholovým úhlem  $u \pm \pi$ . Podle obr. 1.7 tedy bod  $Q_1$  je řešením, bod  $Q_2$  řešením není.

Speciálním případem je protínání zpět z délky a úhlu, kdy bod  $P_3$  je totožný s jedním z bodů  $P_1$  nebo  $P_2$ .

Singulární situace opět závisí na dílčích úlohách – určení parametrů kružnice a protínání z délek.

### 1.2.7 Protínání zpět

Jsou dány souřadnice bodů  $P_1$  až  $P_4$  a obvodové úhly  $u_1$  a  $u_2$  (obr. 1.8).



Obrázek 1.8: Protínání zpět

Pro oba obvodové úhly a odpovídající dané body se vypočtou kružnice  $k_1$  a  $k_2$  a úloha se převede na výpočet protínání z délek. Z řešení se vyloučí průsečíky, které mají obvodový úhel  $u_{1,2} \pm \pi$ . Na obr. 1.8 je bod  $Q_1$  řešením, bod  $Q_2$  řešením není.

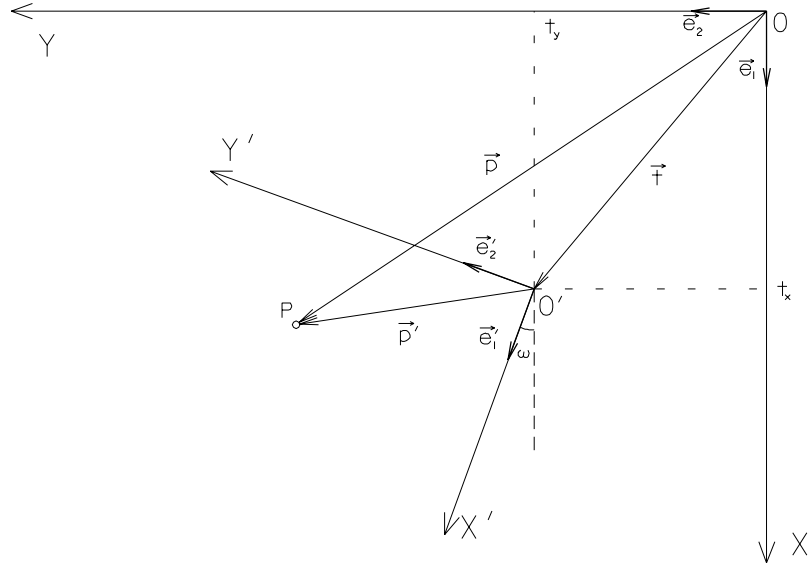
Pro singulární řešení platí podmínky z úloh určení parametrů kružnice a protínání z délek. Například u známé situace kdy všechny body leží poblíž jedné kružnice, bude identifikováno nejednoznačné řešení „malý úhel protnutí“ při výpočtu protínání z délek.

### 1.3 Podobnostní transformace v rovině

Problematika transformací těsně souvisí s výpočty polohy bodu, neboť některé úlohy jsou bez výpočtu v pomocné souřadnicové soustavě a následné transformace neřešitelné – typickým příkladem je vetknutý polygonový pořad. Pro výpočet přibližných souřadnic postačuje transformace podobnostní, při které dochází k posunu a potočení souřadnicové soustavy a ke změně měřítka; změna je stejná ve směru obou os.

Pro odvození transformačních rovnic vyjdeme z obr. 1.9, na kterém je pomocná souřadnicová soustava  $[Y';X']$  definována jednotkovými vektory  $e'_1$  a  $e'_2$ , tvořícími *kanonickou bázi*. Cílovou soustavu definují jednotkové vektory  $e_1$  a  $e_2$ . Pro polohový vektor libovolného bodu v cílové soustavě platí vektorový součet

$$\vec{p} = \vec{p}' + \vec{t} \quad (1.38)$$



Obrázek 1.9: Podobnostní transformace v rovině

Tento vztah lze rozepsat pro jednotlivé souřadnice

$$xe_1 + ye_2 = x'e'_1 + y'e'_2 + t_x e_1 + t_y e_2 \quad (1.39)$$

a po úpravě a uvažování změny měřítka získáme transformační rovnice pro podobnostní transformaci v rovině

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} t_x \\ t_y \end{pmatrix} + q \begin{pmatrix} \cos \omega & -\sin \omega \\ \sin \omega & \cos \omega \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix} \quad (1.40)$$

$$\mathbf{X} = \mathbf{t} + q \cdot \mathbf{R} \cdot \mathbf{X}' = \mathbf{t} + \mathbf{A} \cdot \mathbf{X}' \quad (1.41)$$

kde

$$\mathbf{A} = \begin{pmatrix} a_1 & -a_2 \\ a_2 & a_1 \end{pmatrix} = q \begin{pmatrix} \cos \omega & -\sin \omega \\ \sin \omega & \cos \omega \end{pmatrix} \quad (1.42)$$

$q$  – měřítko transformace

Transformační klíč je tedy tvořen prvky  $a_1$ ,  $a_2$ ,  $t_x$ ,  $t_y$  a pro jeho určení je nutné znát souřadnice dvou identických bodů. Pro výpočet transformačních koeficientů se do rovnice (1.41) dosadí postupně souřadnice identických bodů  $P_1$  a  $P_2$  a rovnice se od sebe odečtou. Identické body  $P_1$  a  $P_2$  mají souřadnice  $[y'_{1,2}; x'_{1,2}]$  v místní souřadnicové soustavě a  $[y_{1,2}; x_{1,2}]$  v soustavě cílové.

$$\begin{pmatrix} \Delta x_{21} \\ \Delta y_{21} \end{pmatrix} = \begin{pmatrix} a_1 & -a_2 \\ a_2 & a_1 \end{pmatrix} \begin{pmatrix} \Delta x'_{21} \\ \Delta y'_{21} \end{pmatrix} \quad (1.43)$$

a tedy

$$\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \frac{1}{\Delta x'_{21}{}^2 + \Delta y'_{21}{}^2} \begin{pmatrix} \Delta x'_{21} & \Delta y'_{21} \\ -\Delta y'_{21} & \Delta x'_{21} \end{pmatrix} \begin{pmatrix} \Delta x_{21} \\ \Delta y_{21} \end{pmatrix} \quad (1.44)$$

Posun počátku pomocné soustavy se vypočte podle vztahu

$$\begin{pmatrix} t_x \\ t_y \end{pmatrix} = \mathbf{X} - \mathbf{A} \cdot \mathbf{X}' \quad (1.45)$$

# Kapitola 2

## Návrh tříd pro výpočet souřadnic bodu

V následující části bude rozebrána problematika návrhu tříd v jazyce C++ pro výpočet souřadnic bodu.

### 2.1 Programovací jazyk C++

Autorem programovacího jazyka C++ je Bjarne Stroustrup, který na počátku 80. let navrhl jazyk „C with Classes“, rozšiřující možnosti jazyka C. V roce 1983 byl název změněn na C++. Výraz C++ značí změny, ke kterým při vývoji jazyka došlo (C++ znamená inkrement C).

Potřeba nového jazyka byla vyvolána snahou zpřehlednit zápis programů a tím zjednodušit i jejich údržbu. K tomuto cíli již směřovalo strukturované programování, jehož použití však neřešilo potíže s úpravami již napsaných programů. Při objektovém řešení je pouze definováno rozhraní objektu (vstupní/výstupní parametry). V okamžiku vzniku požadavku na změnu chování objektu se opraví pouze objekt samotný bez nutnosti zásahu do ostatních částí zdrojového textu.

Jazyk C byl vybrán jako základ z několika důvodů [4]

- všestranný, relativně nízkourovňový,
- vhodný pro většinu úloh,
- dostupný na naprosté většině platform.

Hlavním rozšířením jsou *třídy* (class), *výjimky* (exception), *šablony* (template), *prostory jmen* (namespace), *přetížení* operátorů a funkcí, *inline* funkce a rozsáhlé knihovny. Celkově poskytuje C++ programátorovi mohutný nástroj pro řešení

úloh. Překladače jsou dostupné prakticky pro všechny operační systémy, pro většinu z nich i pod GPL licencí<sup>1</sup> jako volně šiřitelné.

Stejně jako k jazyku C patří neoddělitelně kniha K&R [5], pro programátora C++ je „biblí“ „The C++ Programming Language“ [4], jejímž autorem je Bjarne Stroustrup.

## 2.2 Třídy

Prvním zcela objektově orientovaným jazykem byl SmallTalk72, vyvíjený firmou Xerox. Po něm následovaly další jazyky, jako C++ (s odnožemi např. Borland C++, Turbo C++), Objective-C a Pascal.

*Třída (class)* je základním nástrojem objektového programování. Představuje reprezentaci reálných nebo abstraktních objektů. Každá třída obsahuje datové struktury a řídicí část, rozhodující o způsobu nakládání s daty. Třídy jsou charakterizovány třemi základními vlastnostmi

**zapouzdření** (*encapsulation*) – datové struktury uvnitř třídy jsou chráněny proti přímému přístupu zvenčí, lze k nim přistupovat pouze prostřednictvím funkcí (metod) třídy.

**dědičnost** (*inheritance*) – je možné vygenerovat následníka (potomka) třídy. Ten dědí všechny vlastnosti (datové typy i metody) svého předka a dále k nim může přidat své vlastní.

**polymorfismus** (*polymorphism*) – mnohotvárnost. Tato vlastnost umožňuje pojmenovat jedním jménem metodu, kterou obsahují všechny třídy na dané větvi dědičného stromu, přičemž v každé třídě může mít tato metoda jinou implementaci.

## 2.3 Knihovna Median

V této kapitole popíšeme jednotlivé třídy knihovny `Median` pro výpočet přibližných souřadnic. Uvedené definice tříd neobsahují těla *inline* metod. Kompletní výpis hlavičkových souborů je uveden v příloze.

### 2.3.1 Základní datové typy

Pro řešení byla využita knihovna `GaMaLib`, obsahující základní datové typy pro řešení souřadnicových výpočtů

---

<sup>1</sup>GNU Public License (GNU's Not Unix), viz <http://www.gnu.org/copyleft/gpl.html>

**Bod** - objektová reprezentace bodu. Obsahuje metody pro manipulaci se souřadnicemi bodu (vlození souřadnic, zjištění jejich velikosti, test existence definovaných souřadnic), seznamem orientačních posunů (na jednom bodě je možné měřit několik osnov vodorovných směrů s různým počátkem) a s dalšími parametry, využívanými během vyrovnání.

**Mereni** - společná (bázová) třída pro měření, potomky jsou **Delka**, **Smer** a **Uhel**. Bázová třída obsahuje metody pro identifikaci typu měření (**PtrSmer()**, **PtrDelka()**, **PtrUhel()**), zjišťování čísel bodů, měřené hodnoty a její střední chyby (**stan()**, **cil()**, **mer()**, **str()**) a další pomocné metody. Každá odvozená třída zajišťuje kontrolu zadávaných hodnot, tisk měření a další operace, specifické pro konkrétní typ měření. Třída **Uhel** obsahuje navíc číslo bodu druhého ramene úhlu (**cil2()**).

**CisBod** - číslo bodu (datový typ `std::string`),

**seznamy** - kontejnery, obsahující seznamy bodů, měření a čísel bodů

**SezBod** – seznam bodů, obsahuje číslo bodu a bod, umožňuje vyhledávání podle čísla bodu

```
class SezBod : public std::map<CisBod, Bod, std::less<CisBod> >
```

**SezMer** – seznam měření

```
class SezMer : public std::vector<Mereni*>
```

**SezCB** – seznam čísel bodů

```
class SezCB : public std::list<CisBod>
```

### 2.3.2 GeodetickaUloha

Základním požadavkem pro návrh je jednotný přístup ke vstupu dat a přebírání výstupních hodnot. Od toho je odvozena existence jedné základní třídy a jejích potomků pro jednotlivé geodetické úlohy. Základní (bázová) třída definuje rozhraní, pomocí kterého bude probíhat komunikace s okolím.

**GeodetickaUloha** je základní třídou, od ní jsou odvozeny další třídy pro konkrétní výpočty.

```
class GeodetickaUloha
{
protected:
    int pocet_reseni;
    Bod* bod1;
    Bod* bod2;
    SezBod* SB;
    virtual void Prevod_mereni(Mereni*, Mereni*) = 0;
```

```
public:
    GeodetickaUloha(SezBod* sb) : pocet_reseni(-1), SB(sb);
    virtual ~GeodetickaUloha();
    virtual void Vypocet() = 0;
    int Pocet_reseni() const;
    Bod Reseni_1() const;
    Bod Reseni_2() const;
};
```

Jako společný znak všech typů výpočtů lze stanovit, že úloha bude pracovat s dvěma body o známých souřadnicích, bude podávat informaci o počtu řešení (obecně žádné, jedno nebo dvě) a podle počtu řešení bude poskytovat i výsledné hodnoty. Metody `Prevod_mereni` a `Vypocet` jsou definovány jako *čistě virtuální* (*pure virtual function*). V důsledku toho je třída `GeodetickaUloha` *abstraktní* a není možné tvořit její instance. U odvozených tříd je nutné čistě virtuální metody definovat.

Chybové stavy jsou ošetřeny prostřednictvím výjimek (*exceptions*), které zajišťují korektní ukončení programu při výskytu dané chyby.

### 2.3.3 Kružnice

Třída `Kružnice` je určena pro výpočet parametrů kružnice, dané dvěma body a obvodovým úhlem (kap. 1.2.1). Výsledný poloměr vrací metoda `r()`, souřadnice středu metoda `Reseni_1()`. Úloha nemá řešení pokud jsou dané body totožné nebo se obvodový úhel blíží celočíselnému násobku  $\pi$  podle vztahu (1.8).

```
class Kružnice : public GeodetickaUloha
{
private:
    Uhel* h1;
    Bod B1, B2;
    Double R;
    void Prevod_mereni(Mereni*, Mereni*) {}
public:
    Kružnice() : GeodetickaUloha(NULL), h1(NULL) {}
    Kružnice(Uhel* u, SezBod* sb) : GeodetickaUloha(sb), h1(u) {}
    ~Kružnice() {}
    void Vypocet();
    void Novy_vypocet(Uhel* u, SezBod* sb);
    Double r() const;
};
```



### 2.3.4 Delka\_delka

Třída `Delka_delka` reprezentuje úlohu protínání z délek (kap. 1.2.3). Při výpočtu jsou detekovány singularity – totožné dané body, některá z délek nulová nebo záporná, neexistující průsečík nebo nejisté protnutí (úhel protnutí menší než cca 10 gon).

```
class Delka_delka : public GeodetickaUloha
{
private:
    Delka* h1;
    Delka* h2;
    CisBod CB;
    Double r1, r2;
    Bod B1;
    Bod B2;
    void Prevod_mereni(Mereni*, Mereni*);
public:
    Delka_delka():GeodetickaUloha(NULL),h1(NULL),h2(NULL),r1(-1) {}
    Delka_delka(Mereni* m1, Mereni* m2, SezBod* sb, CisBod cb) :
        GeodetickaUloha(sb), CB(cb), r1(-1);
    Delka_delka(Double& m1,Double& m2,Bod b1,Bod b2,SezBod* sb) :
        GeodetickaUloha(sb), r1(m1), r2(m2), B1(b1), B2(b2) {}
    ~Delka_delka() {}
    void Vypocet();
    void Novy_vypocet(Mereni* m1, Mereni* m2, SezBod* sb, CisBod cb);
};
```

### 2.3.5 Smer\_smer

V třídě `Smer_smer` se provádí výpočet protínání ze směrnic (kap. 1.2.2). Během výpočtu jsou testovány singulární situace (totožné dané body, směry se neprotínají nebo se protínají v jednom z daných bodů) i nejisté protnutí.

```
class Smer_smer : public GeodetickaUloha
{
private:
    Smer* h1;
    Smer* h2;
    void Prevod_mereni(Mereni*, Mereni*);
public:
    Smer_smer() : GeodetickaUloha(NULL), h1(NULL), h2(NULL) {}
    Smer_smer(Mereni* m1, Mereni* m2, SezBod* sb) :
        GeodetickaUloha(sb);
```

```

~Smer_smer() {}
void Vypocet();
void Novy_vypocet(Mereni* m1, Mereni* m2, SezBod* sb);
};

```

### 2.3.6 Smer\_delka

Tato třída zajišťuje výpočet průsečíku orientované polopřímky s kružnicí (kap. 1.2.4). Řešení neexistuje pro záporný nebo nulový poloměr nebo pokud polopřímka míjí kružnici.

```

class Smer_delka : public GeodetickaUloha
{
private:
    Smer* h1;
    Delka* h2;
    Bod B;
    Double r;
    void Prevod_mereni(Mereni*, Mereni*);
public:
    Smer_delka() : GeodetickaUloha(NULL),h1(NULL),h2(NULL),r(-1) {}
    Smer_delka(Mereni* m1,Mereni* m2,SezBod* sb)
        : GeodetickaUloha(sb),r(-1);
    Smer_delka(Smer* m1, Double m2, Bod b, SezBod* sb) :
        GeodetickaUloha(sb), h1(m1), B(b), r(m2) {}
    ~Smer_delka() {}
    void Vypocet();
    void Novy_vypocet(Mereni* m1, Mereni* m2, SezBod* sb);
};

```

### 2.3.7 Smer\_uhel

Úloha spojeného protínání (kap. 1.2.5) se řeší rozkladem na dílčí výpočty – určení kružnice a následně průsečík orientované polopřímky s kružnicí. Existence řešení se odvozuje od podmínek řešitelnosti těchto úloh. Závěrečný výběr z (obecně) dvou průsečíků se provádí porovnáním měřeného a vypočteného obvodového úhlu – viz obr. 1.6 na straně 14.

```

class Smer_uhel : public GeodetickaUloha
{
private:
    Smer* h1;
    Uhel* h2;

```

```
void Prevod_mereni(Mereni*, Mereni*);  
public:  
    Smer_uhel() : GeodetickaUloha(NULL),h1(NULL),h2(NULL) {}  
    Smer_uhel(Mereni* m1, Mereni* m2, SezBod* sb) :  
        GeodetickaUloha(sb);  
    ~Smer_uhel() {}  
    void Vypocet();  
    void Novy_vypocet(Mereni* m1, Mereni* m2, SezBod* sb);  
};
```

### 2.3.8 Delka\_uhel

Kombinace délky a vnitřního (kap. 1.2.6) se vypočte protínáním z délek po odvození parametrů kružnice z vnitřního úhlu. Podmínky řešitelnosti závisí na dílčích úlohách. Konečný výběr z průsečíků se provede podle obr. 1.7 na straně 15.

```
class Delka_uhel : public GeodetickaUloha  
{  
private:  
    Delka* h1;  
    Uhel* h2;  
    void Prevod_mereni(Mereni*, Mereni*);  
public:  
    Delka_uhel() : GeodetickaUloha(NULL), h1(NULL), h2(NULL) {}  
    Delka_uhel(Mereni* m1, Mereni* m2, SezBod* sb) :  
        GeodetickaUloha(sb);  
    ~Delka_uhel() {}  
    void Vypocet();  
    void Novy_vypocet(Mereni* m1, Mereni* m2, SezBod* sb);  
};
```

### 2.3.9 Uhel\_uhel

Protínání zpět z úhlů se vypočte pomocí protínání z délek (pro každý úhel jedna kružnice). Singulární řešení tedy opět závisí na úlohách určení parametrů kružnice a protínání z délek. Výběr průsečíků, odpovídajících měřenému vnitřnímu úhlu, se provede podle obr. 1.8 na straně 16.

```
class Uhel_uhel : public GeodetickaUloha  
{  
private:  
    Uhel* h1;  
    Uhel* h2;
```

```
void Prevod_mereni(Mereni*, Mereni*);
public:
    Uhel_uhel() : GeodetickaUloha(NULL), h1(NULL), h2(NULL) {}
    Uhel_uhel(Mereni* m1, Mereni* m2, SezBod* sb) :
        GeodetickaUloha(sb);
    ~Uhel_uhel() {}
    void Vypocet();
    void Novy_vypocet(Mereni* m1, Mereni* m2, SezBod* sb);
};
```

### 2.3.10 Výběr řešení

Obecně je výsledkem výpočtu geodetické úlohy žádné, jedno nebo dvě řešení. Třída `Vyber_reseni` má za úkol pokusit se v případě dvojnásobného řešení vybrat správné. Výběr se provádí pomocí porovnání dalších určujících prvků s hodnotami vypočtenými ze souřadnic těchto řešení. Pro úspěšný výběr je tedy nutný nadbytečný počet měření.

Metoda `Reseni()` vrací vybrané řešení – pokud bylo možno na základě dalších určujících prvků odstranit nejednoznačnost. V opačném případě vrací metoda `Stav()` hodnotu `neni_reseni` a pokus o volání metody `Reseni()` skončí vyvoláním výjimky.

```
class Vyber_reseni
{
private:
    Navesti_stavu stav;
    Bod B1, B2;
    SezBod* SB;
    SezMer* SM;
public:
    Vyber_reseni(SezBod* sb, SezMer* sm)
        : stav(neproveden_vypocet), SB(sb), SM(sm) {}
    Vyber_reseni(Bod& b1, Bod& b2, SezBod* sb, SezMer* sm) :
        stav(neproveden_vypocet), B1(b1), B2(b2), SB(sb), SM(sm) {}
    void Vypocet();
    void Vypocet(Bod b1, Bod b2);
    Bod Reseni();
    int Stav() const ;
}; // class Vyber_reseni
```

### 2.3.11 Určení mediánu souřadnic

Během výpočtu získáme množinu bodů, z nichž se vybírají konečné souřadnice. Pro výběr nejpravděpodobnější hodnoty jsem zvolil medián, neboť je minimálně ovlivněn hrubými chybami. Medián tedy představuje robustní odhad, na rozdíl od aritmetického průměru. Třída `Statistika` provádí výběr mediánu z množiny bodů, výsledné souřadnice vrací metoda `Median()`.

```
class Statistika
{
private:
    Pom_seznam* PS;
    Bod median;
    Navesti_stavu stav;
public:
    Statistika() : stav(neni_init) {}
    Statistika(Pom_seznam* ps) : PS(ps),stav(neproveden_vypocet) {}
    void Vypocet();
    void Vypocet(Pom_seznam* ps);
    Navesti_stavu Stav() const { return stav; }
    // vysledna souradnice
    Bod Median();
}; // class Statistika
```

### 2.3.12 Transformace souřadnic

Pro některé konfigurace – např. Hansenovu úlohu – je nutný výpočet v místním souřadnicovém systému a následná transformace. Třída `Transformace` je bázovou čistě virtuální třídou, jejím potomkem je třída `PodobnostniTr2D`, která zajišťuje výpočet podobnostní transformace v rovině (kap. 1.3).

```
class Transformace
{
protected:
    // seznam bodu v cilove soustave
    SezBod& SB;
    // seznam bodu v mistni soustave
    SezBod& mistni;
    // seznam urcovanych bodu
    SezCB& urcovane;
    Navesti_stavu stav;
    // body transformovane z mistni do cilove soustavy
    SezBod transf_body;
    // v kazdem potomkovi se musi definovat - bude se tam minimalne
```

```

// cistit transformacni klic
virtual void Reset() = 0;
public:
    Transformace(SezBod& sb, SezBod& mist, SezCB& urc)
        : SB(sb),mistni(mist),urcovane(urc),stav(neproveden_vypocet) {}
    virtual ~Transformace();
    void Reset(SezBod& sb, SezBod& mist, SezCB& urc);
    virtual void Vypocet() = 0;
    Navesti_stavu Stav() const { return stav; }
    SezBod Transf_body() const;
}; // class Transformace

// -----
// Transformace v rovine
// Transformace z nutneho poctu bodu - vybira nejvhodnejši dvojici
// - podobnostni transformace
/*
 * transformacni klic pro podobnostni transformaci:
 *
 * | x |   |tx | |a1  -a2| |x'|
 * |   | = |   | +|       |*| |
 * | y |   |ty | |a2   a1| |y'|
 *
 * transf_klic[0] = a2
 * transf_klic[1] = a1
 * transf_klic[2] = ty
 * transf_klic[3] = tx
 *
 */
class PodobnostniTr2D : public Transformace
{
private:
    std::vector<Double> transf_klic;
    void Reset();
    bool Dany_bod(const CisBod& cb);
    void Identicke_body(SezBod::iterator& b1, SezBod::iterator& b2);
    void Transformacni_klic(SezBod::iterator& b1,
                           SezBod::iterator& b2);
public:
    PodobnostniTr2D(SezBod& sb, SezBod& mist, SezCB& urc)
        : Transformace(sb, mist, urc) { Reset(); }
    void Vypocet();
    std::vector<Double> Transf_klic() const;
}; // class PodobnostniTr2D

```

### 2.3.13 Výpočet přibližných souřadnic jednoho bodu

Třída `PriblBod` zastřešuje všechny předešlé třídy a provádí výpočet přibližných souřadnic zadaného bodu. Výpočet probíhá tak, že se nejdříve ze seznamu měření vyberou ta, která souvisí se zadaným bodem, převedou se na určující prvky a následně se vypočtou všechny kombinace dvojic určujících prvků. Tím se získá seznam souřadnic řešení jednotlivých dvojic a z něj se výsledná hodnota vybere jako medián souřadnic. Pokud se nepodaří vyřešit dvojznačnost, může třída vrátit obě možná řešení.

```
class PriblBod
{
private:
    Pom_seznam Vyp_body;
    SezBod SB;
    CisBod CB;
    SezMer SM;
// Navesti_stavu -> viz g_fce.h
    SezMer SM_U;           // pro vnitřní uhly
    SezMer SM_S;         // pro vnější směry
    Bod v_bod;           // konečný bod - výstup
    Bod v_bod2;         // --"-- - pro dvojznačné řešení
    SezBod* SB_puv;     // opakování výpočtu s jiným CB
    SezMer* SM_puv;
    Navesti_stavu stav;
    void CistiSeznamy(); // vyprázdnění pomocných seznamů
    Uhel* VyrobuUhel(const SezMer::iterator i,
                    const SezMer::iterator j);
    Smer* VyrobuSmernik(const Uhel* u, const CisBod& cb);
    Smer* VyrobuSmernik(const Smer* s, const CisBod& cb);
    bool ZnamyCil(const SezMer::iterator i);
    bool ZnamyCil1(const SezMer::iterator i);
    bool ZnamyCil2(const Uhel* u);
    bool ZnameStan(const SezMer::iterator i);
    void SrovnejMereni(const SezMer&);
    void Reset(const SezBod*, const SezMer*, const CisBod&);
public:
    PriblBod(const SezBod* sb, const SezMer* sm, const CisBod& cb) :
        SB_puv(sb), SM_puv(sm), stav(neni_init);
    PriblBod(const SezBod* sb, const SezMer* sm)
        : SB_puv(sb), SM_puv(sm), stav(neni_init) {};
    ~PriblBod();
    void Vypocet(const SezBod* sb, const SezMer* sm, const CisBod& cb);
    void Vypocet(const CisBod& cb);
    void Vypocet();
};
```

```
Navesti_stavu Stav() const { return stav; }  
Bod Reseni();  
Bod Reseni_2();  
};
```

### 2.3.14 Výpočet přibližných souřadnic seznamu zadaných bodů

Třída `PriblSour` je vrcholovou třídou knihovny `Median`. Zajišťuje veškerou komunikaci s ostatními částmi programu a definuje rozhraní pro výpočet přibližných souřadnic.

Konstruktor třídy má dva parametry, seznam bodů a seznam měření. Již vytvořenou instanci třídy lze inicializovat pro jiné vstupní hodnoty metodou `Reset(SezBod& ,SezMer&)`.

Seznam určovaných bodů lze zadat několika způsoby

- `Vypocet()` – vypočte přibližné souřadnice všech bodů, které existují v seznamu bodů ale nemají souřadnice a všech bodů, které jsou uvedeny pouze jako stanovisko nebo cíl v seznamu měření a neexistují v seznamu bodů,
- `Vypocet(CisBod)` – určí přibližné souřadnice zadaného bodu,
- `Vypocet(SezCB)` – vypočte přibližné souřadnice bodů, uvedených v seznamu.

Všechny tři metody vrací `true`, pokud se podařilo určit přibližné souřadnice všech zadaných bodů.

Vypočtené body jsou doplněny do seznamu bodů, navíc lze získat seznam vypočtených bodů (metoda `Vypoctene()`) a seznam čísel bodů, u nichž byl výpočet neúspěšný (metoda `Nevypoctene()`). Metoda `Vypocteno_vse()` vrací `true`, pokud se podařilo určit souřadnice všech zadaných bodů.

```
class PriblSour  
{  
private:  
// seznam bodu, v nem se predavaji vysledky  
    SezBod& SB;  
    SezMer& SM;  
// Navesti_stavu -> viz g_fce.h  
    Navesti_stavu stav;  
// seznam pozadovanych bodu (pocitanych)  
    SezCB urcovane;
```



```

// seznam vypoctenych bodu
SezBod vypoctene;
int vnoreni;
bool Neobsahuje(CisBod cb)
{
    return (std::find(urcovane.begin(),urcovane.end(),cb) ==
            urcovane.end());
}
bool Mistni_mereni(SezMer::iterator sm, SezCB sb)
{
    bool pom = false;
    pom = (std::find(sb.begin(),sb.end(),(*sm)->stan()) !=
            sb.end()) &&
            (std::find(sb.begin(), sb.end(), (*sm)->cil()) != sb.end());
    if((*sm)->PtrUhel())
    {
        Uhel* u = static_cast<Uhel*>(*sm);
        pom = pom && (std::find(sb.begin(), sb.end(), u->cil2()) !=
                    sb.end());
    }
    return pom;
}
void Reset()
{
    stav = neproveden_vypocet;
    urcovane.erase(urcovane.begin(), urcovane.end());
    vypoctene.erase(vypoctene.begin(), vypoctene.end());
}
bool Mereni_obsahujeCB(SezMer::iterator m, SezCB::iterator cb)
{
    bool pom = (((*m)->stan() == (*cb)) || ((*m)->cil() == (*cb)));
    if((*m)->PtrUhel())
    {
        Uhel* u = static_cast<Uhel*>(*m);
        pom = pom || (u->cil2() == (*cb));
    };
    return pom;
}
// true - existuji alespon dva body s danymi souradnicemi
bool Resitelna_data(SezBod& b);
// true - existuji minimalne 2 mereni s bodem cb
bool Nutna_mereni(CisBod cb);
// v SM a SB najde body bez souradnic a jejich cisla da do urcovane
void Najdi_nevypoctene();
// presun bodu cb z seznamu odkud do kam

```

```

    void Presun(SezBod& odkud, SezBod& kam, CisBod& co);
// vypocet klasickym protinanim; true - vypocetlo alespon jeden bod
    bool Pocitej_protinani(SezBod& body, SezCB& co);
// vypocet bodu, které nejdou beznym protinanim - napr. vetknuty PGP
// true - urceny souradnice alespon jednoho bodu
    bool Pocitej_vetknuti();
// kombinace predeslych dvou metod
    void Vypocetni_cyklus();
public:
    PriblSour(SezBod& b, SezMer& m) : SB(b), SM(m), vnoreni(0)
    {
        Reset();
    }
    PriblSour(SezBod& b, SezMer& m, int vn)
        : SB(b), SM(m), vnoreni(vn)
    {
        Reset();
    }
    void Reset(SezBod& b, SezMer& m)
    {
        SB = b;
        SM = m;
        vnoreni = 0;
        Reset();
    }
// jeden bod - i uz vypocteny; true - podarilo se urcit souradnice
    bool Vypocet(CisBod cb);
// body v SezCB - i uz vypoctene;
// true - podarilo se urcit vsechny souradnice
    bool Vypocet(SezCB cb);
// vsechny nezname body; true - podarilo se urcit vsechny souradnice
    bool Vypocet()
    {
// hledam vsechny body, které nejsou v seznamu bodu (jen u mereni)
// a body nemajici souradnice
        stav = probehl_vypocet;
        Najdi_nevypoctene();
        if(!Resitelna_data(SB))
            return false;
        Vypocetni_cyklus();
        return Vypocteno_vse();
    }
    bool Vypocteno_vse() const
    {
        if(stav <= neproveden_vypocet)

```

```
    throw ps_chyba(“PriblSour::Vypocteno_vse: neproveden vypocet”);
return urcovane.empty();
}
SezCB Nevypoctene() const
{
    if(stav <= neproveden_vypocet)
        throw ps_chyba(“PriblSour::Nevypoctene: neproveden vypocet”);
    return urcovane;
}
SezBod Vypoctene() const
{
    if(stav <= neproveden_vypocet)
        throw ps_chyba(“PriblSour::Vypoctene: neproveden vypocet”);
    return vypoctene;
}
int Celkem_bodu () const
{
    return SB.size();
}
int Celkem_mereni () const
{
    return SM.size();
}
};
```

# Kapitola 3

## Praktické použití třídy Median

Výpočet přibližných souřadnic byl zapracován do programu pro vyrovnání rovinné sítě GaMa. Přístup k programu je volný přes rozhraní elektronické pošty. Po zaslání dopisu na adresu `gama-sit-ukazka@gama.fsv.cvut.cz` je vrácen ukázkový soubor se vstupními daty. Pro výpočet vyrovnání je nutné poslat na adresu `gama-sit@gama.fsv.cvut.cz` dopis, v jehož těle je soubor vstupních dat. Protokol o výpočtu je zaslán automaticky na adresu odesilatele.

### 3.1 Formát vstupních dat

Obecný tvar vstupních dat pro program GaMa je následující

```
<sit>

    <popis>
    </popis>

    <parametry>
    </parametry>

    <body-mereni>
    </body-mereni>

</sit>
```

V celém souboru je znak # považován za začátek komentáře a všechny znaky za ním až do konce řádky se ignorují. Všechny znaky před úvodní značkou `<sit>`

jsou ignorovány a nemají žádný význam. Jednotlivé sekce mohou být řazené v libovolném pořadí a mohou být uvedeny vícekrát. Za koncovou značkou `</sit>` jsou povoleny pouze komentářové řádky.

Text uvnitř sekce `<popis>` je vypsan na začátek protokolu a slouží uživateli k rozlišení jednotlivých úloh. Řádky nesmí začínat znakem `#`, neboť by byly považovány za komentář.

Značka `<parametry>` zahajuje část definující parametry vyrovnání a cesty k výstupním souborům. Každý parametr se umísťuje na samostatnou řádku.

**vystup\_txt** *soubor* – jméno souboru pro výstupní protokol,

**vystup\_stx** *soubor* – jméno souboru pro výstupní seznam souřadnic,

**vystup\_opr** *soubor* – jméno souboru pro výstupní soubor koeficientů rovnic oprav,

**m0apr** *hodnota* – apriorní jednotková střední chyba,

**konf\_pr** *hodnota* – konfidenční pravděpodobnost,

**tol\_abs** *hodnota* – tolerance pro testování absolutních členů,

**typ\_m0** *empiricka|apriorni* – typ jednotkové střední chyby.

Sekce `<body-mereni>` slouží k vstupu typů bodů (pevný, opěrný, určovaný), souřadnic daných bodů a seznamu měření.

**číslo bodu** / **pb|ob|ub** – definice typu bodu – pevný (*pb*), opěrný (*ob*), určený bod (*ub*)

**číslo bodu** **yx = y x** – vstup souřadnic bodu, může být doplněn o výšku (**v = výška**) nebo o orientační posun (**z = or\_posun**)

**stanovisko** **s = cíl směr střední\_chyba** – měřený směr

**stanovisko** **d = cíl délka střední\_chyba** – měřená délka

**stanovisko** **u = cíl cíl2 úhel střední\_chyba** – měřený úhel

U vstupu měření je možné vynechat stanovisko u druhého a dalších měření, pokud jsou seřazena za sebou.

## 3.2 Příklady

Programem GaMa bylo zpracováno několik příkladů pro ukázkou funkčnosti knihovny Median. Zobrazení sítí a výpočetní protokoly jsou uvedeny v příloze. Jednotlivé příklady byly vybrány tak, aby demonstrovaly možnosti knihovny.

### Hansenova úloha

Hansenova úloha je typickým zástupcem konfigurací, při nichž je nutno provést výpočet souřadnic v pomocné souřadnicové soustavě a výsledek transformovat. Při transformaci se výrazně mění měřítko, neboť rozměr sítě není měřením definován a při volbě pomocné souřadnicové soustavy se definuje libovolně. V úloze je nutný počet určujících prvků (4 úhly pro 2 body) a tedy nedochází k vyrovnání.

### Volná síť

Při výpočtu volné sítě jsou za dané považovány pouze dva body (platí pro výpočet přibližných souřadnic; u vyrovnání postačuje např. jeden bod a směr – obecně je síť umístěna za platnosti nutného počtu podmínek), ostatní body se určují. Příklad je převzat ze zaměření areálu zahrad kláštera Bosých karmelitánů ve Slaném.

### Vázaná síť

Vázaná síť je umístěna při platnosti nadbytečného počtu podmínek – počet daných bodů pro výpočet přibližných souřadnic je tedy větší. Jako příklad byla použita ukázková síť pro program GaMa, kterou lze získat zasláním dopisu na adresu `gama-sit-ukazka@gama.fsv.cvut.cz`.

### Vázaná síť – zhoršená konfigurace

Uvedený příklad pochází ze zaměření drážního tělesa bývalé vlečky a přilehlého území v Libušíně u Kladna. Z hlediska výpočtu přibližných souřadnic má síť nevhodnou konfiguraci, protože mezi danými body se nachází vždy velké množství bodů určených a lze tedy očekávat velké rozdíly mezi přibližnými a vyrovnávanými souřadnicemi. To se také projevilo v prvním protokolu s výpočtem přibližných souřadnic. Maximální opravy souřadnic u bodu č.4002 činí 129 a 137 mm. Následné opakované vyrovnání, při kterém byly jako přibližné souřadnice použity vyrovnané z prvního výpočtu, prokázalo zanedbatelné rozdíly řádu setin milimetru. Přesnost výpočtu přibližných souřadnic lze tedy v tomto případě prohlásit za dostatečnou.

# Závěr

Spojením s programem pro vyrovnání rovinné sítě vzniknul silný nástroj pro geodetické výpočty. Jako vstupní hodnoty totiž slouží pouze seznam měření, seznam souřadnic a informace o typech bodů. Ostatní operace již probíhají automaticky a geodet pouze rozhoduje o případném vyřazení některých bodů nebo měření z procesu vyrovnání.

Na závěr bych rád poděkoval všem lidem pracujícím v oblasti tvorby *free software* za vývoj kvalitních a volně dostupných programů jako je například GNU C++, Emacs a další. Tato diplomová práce byla vysázena v systému  $\text{\LaTeX}$  pod operačním systémem Linux, distribuce Debian 2.1.

Elektronická podoba diplomové práce je dostupná ve formátu PDF na adrese <http://gama.fsv.cvut.cz/~vesely/diplomka/diplomka99.pdf>.

# Literatura

- [1] Blažek, R., Jandourek, J.: Geodézie (úpravy měřených veličin a výškopis), skriptum, vydavatelství ČVUT, Praha, 1994
- [2] Charamza, F., Stádníková, A.: Knihovna základních procedur pro počítač EC 1030, edice VÚGTK, Praha, 1979
- [3] Ryšavý ,J.: Praktická geometrie (nižší geodesie), 1. vydání, Česká matice technická, Praha, 1941, 710 str.
- [4] Stroustrup, B.: The C++ Programming Language, 3rd ed., Addison–Wesley, Reading, Mass., June 1997, 910 p., ISBN 0-201-88954-4  
<http://www.aw.com/cp/stroustrup3d>
- [5] Kernighan, B.W., Ritchie, D.M.: The C Programming Language, Prentice-Hall Inc. New Jersey 1978  
slovenský překlad: Programovací jazyk C, Alfa – SNTL, Bratislava – Praha, 1978, 249 str.



# Příloha A

## Ukázkové příklady

## A.1 Hansenova úloha

Davkove vyrovnani mistni trigonometricke site verze: 0.6.3 / 0.8.0 / GNU C++  
 \*\*\*\*\*

Vypocet pribliznych souradnic bodu  
 \*\*\*\*\*

Pocet bodu se znamymi souradnicemi: 2  
 Pocet urcovanych bodu : 2  
 Pocet mereni : 6

-----  
 Uspesne vypoctenych bodu : 2  
 Zustalo nevypocteno bodu : 0

Popis site  
 \*\*\*\*\*

Hansenova uloha

Zakladni parametry reseni ulohy  
 \*\*\*\*\*

Pocet urcovanych bodu : 2  
 Pocet pevnych bodu : 2  
 Bodu celkem : 4

Pocet smeru : 6 Pocet osnov : 2  
 Celkem pozorovani : 6

Pocet rovnic oprav : 6 Pocet neznamych: 6  
 Pocet nadbyt. pozorovani: 0 Defekt site : 0  
 m0' [mm/cc] : 0.00 [pvv] : 1.23630e-47

Pri statisticke analyze se pracuje

- s empirickou jednotkovou stredni chybou 0.00 [mm/cc]  
 - s konfidencni pravdepodobnosti 95 %

Prehled vyrovnanych pozorovani  
 \*\*\*\*\*

i	stanovisko	cil	merena hodnota	vyrovnana [m/g]	str.ch. [mm/cc]	konf.i. [mm/cc]
1	4001	1 smer	0.000000	0.000000	0.0	0.0
2		2 smer	112.756500	112.756500	0.0	0.0
3		4002 smer	163.728400	163.728400	0.0	0.0

4	4002	4001 smer	0.000000	0.000000	0.0	0.0
5		1 smer	18.377000	18.377000	0.0	0.0
6		2 smer	113.502000	113.502000	0.0	0.0

Prehled neznamych - orientacni posuny

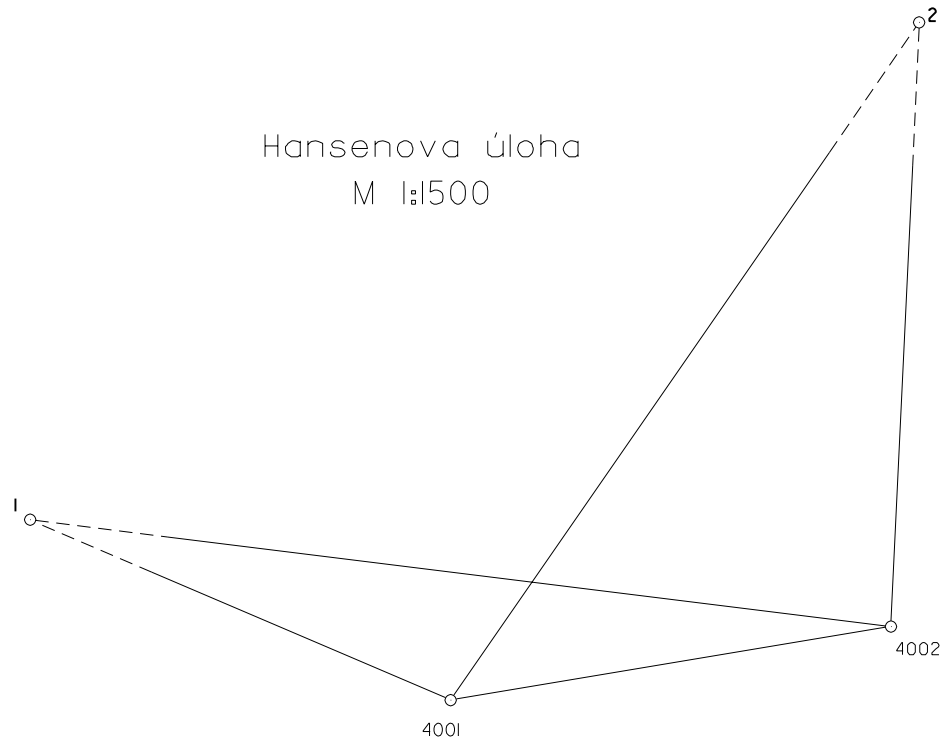
\*\*\*\*\*

i	stanovisko	priblizna hodn. [g]	korekce [g]	vyrovnana hodn. [g]	str.ch. +-[cc]	konf.i.
1	4001	125.726036	-0.000000	125.726036	0.0	0.0
6	4002	89.454436	-0.000000	89.454436	0.0	0.0

Prehled neznamych - souradnice

\*\*\*\*\*

i	bod	priblizna hodnota	korekce [m]	vyrovnana hodnota	str.ch. +-[mm]	konf.i.
	4001	typ: U		mp =	0.0	
2	y	916.46041	-0.000000	916.46041	0.0	0.0
3	x	5035.72477	0.000000	5035.72477	0.0	0.0
	4002	typ: U		mp =	0.0	
4	y	829.13563	0.000000	829.13563	0.0	0.0
5	x	5021.12571	-0.000000	5021.12571	0.0	0.0



Obrázek A.1: Hansenova úloha

## A.2 Volná síť

Davkové vyrovnání místní trigonometrické sítě verze: 0.6.3 / 0.8.0 / GNU C++

\*\*\*\*\*

Výpočet přibližných souřadnic bodů

\*\*\*\*\*

Počet bodů se známými souřadnicemi: 2  
 Počet určených bodů : 10  
 Počet měření : 64

-----  
 Úspěšně vypočtených bodů : 10  
 Zůstalo nevypočteno bodů : 0

Popis sítě

\*\*\*\*\*

Klášter Bosých karmelitánů Slaný

Základní parametry řešení úlohy

\*\*\*\*\*

Počet určených bodů	: 11	Operní body	: 1
Počet pevných bodů	: 1		
Bodů celkem	: 12		
Počet směrů	: 32	Počet osnov	: 12
Počet delek	: 32		
Celkem pozorování	: 64		
Počet rovnic oprav	: 64	Počet neznámých:	34
Počet nadbytl. pozorování:	31	Defekt sítě	: 1
$m_0'$ [mm/cc]	: 18.26	[pvv]	: 1.03334e+04

Při statistické analýze se pracuje

- s empirickou jednotkovou střední chybou 18.26 [mm/cc]
- s konfidencí pravděpodobnosti 95 %

Pomer  $m_0'$  empirická /  $m_0$  apriorní: 1.826

95 % interval (0.752, 1.247) neobsahuje hodnotu  $m_0'/m_0$

$m_0'/m_0$  (delky): 1.617  $m_0'/m_0$  (smery): 2.534

Maximální pokles  $m_0''/m_0$  při vyloučení jednoho pozorování: 1.548

Maximální studentizovaná oprava 3.07

presahuje kritickou hodnotu 1.95 na hladině významnosti 5 %

52. pozorování - stanovisko 4010 d = 4011 30.799 10.0

## Přehled vyrovnaných pozorování

\*\*\*\*\*

i	stanovisko	cíl	merena hodnota	vyrovnana [m/g]	str.ch. +-[mm/cc]	konf.i. ==
1	4001	4005 delka	91.66600	91.66432	7.3	14.9
2		4005 smer	385.498800	385.498606	31.2	63.6
3		4006 delka	42.07260	42.06272	5.9	11.9
4		4006 smer	35.087800	35.086198	32.3	65.8
5		4002 delka	57.62507	57.62664	7.4	15.0
6		4002 smer	65.422800	65.426082	30.5	62.1
7		4007 delka	50.60405	50.59825	11.2	22.9
8		4007 smer	299.862600	299.861115	35.6	72.7
9	4002	4001 delka	57.63344	57.62664	7.4	15.0
10		4001 smer	345.309000	345.304528	30.8	62.8
11		4006 delka	27.97562	27.97201	4.6	9.4
12		4006 smer	393.765100	393.767369	32.4	66.1
13		4003 delka	54.94537	54.96366	6.3	12.9
14		4003 smer	83.192400	83.192181	31.3	63.8
15		4004 delka	88.68182	88.68590	7.1	14.5
16		4004 smer	51.609300	51.611722	29.0	59.2
17	4003	4002 delka	54.95651	54.96366	6.3	12.9
18		4002 smer	223.831600	223.832871	32.5	66.4
19		4004 delka	48.08953	48.08649	5.6	11.3
20		4004 smer	355.632500	355.631229	32.5	66.4
21	4004	4003 delka	48.06631	48.08649	5.6	11.3
22		4003 smer	100.149200	100.149870	31.3	63.8
23		4002 delka	88.68397	88.68590	7.1	14.5
24		4002 smer	136.776000	136.771053	29.6	60.3
25		4005 delka	37.97908	37.98962	6.6	13.5
26		4005 smer	228.606500	228.611227	31.2	63.6
27		4008 delka	27.74674	27.73359	4.4	9.0
28		4008 smer	381.651100	381.650650	32.2	65.7
29	4008	4005 delka	61.42826	61.41478	8.6	17.5
30		4005 smer	292.767400	292.765350	31.4	64.1
31		4004 delka	27.74792	27.73359	4.4	9.0
32		4004 smer	265.450400	265.449921	27.5	56.2
33		4003 delka	62.01763	62.01335	6.6	13.4
34		4003 smer	212.136100	212.138629	33.0	67.3
35	4006	4001 delka	42.06110	42.06272	5.9	11.9
36		4001 smer	282.900000	282.902038	32.9	67.0
37		4002 delka	27.95553	27.97201	4.6	9.4
38		4002 smer	161.706800	161.704762	32.9	67.0
39	4005	4009 delka	54.49492	54.47667	11.3	23.1
40		4009 smer	72.594200	72.594839	35.7	72.8
41		4004 delka	37.98679	37.98962	6.6	13.5
42		4004 smer	284.583200	284.578418	28.4	57.9
43		4008 smer	264.930700	264.933270	29.0	59.2
44		4001 delka	91.65400	91.66432	7.3	14.9

45		4001 smer	6.502000	6.503574	32.2	65.7
46	4009	4005 delka	54.48167	54.47667	11.3	23.1
47		4005 smer	339.255500	339.255220	36.4	74.3
48		4010 delka	22.07755	22.04751	11.4	23.3
49		4010 smer	170.421200	170.421480	36.4	74.3
50	4010	4009 delka	22.03306	22.04751	11.4	23.3
51		4009 smer	275.141300	275.141300	36.5	74.5
52		4011 delka	30.79860	30.75423	11.2	22.7
53	4011	4010 delka	30.73454	30.75423	11.2	22.7
54		4010 smer	111.148600	111.148724	36.4	74.2
55		4012 delka	25.09393	25.09126	9.9	20.3
56		4012 smer	244.212000	244.211876	36.4	74.2
57	4012	4011 delka	25.10640	25.09126	9.9	20.3
58		4011 smer	162.833200	162.833036	36.3	74.0
59		4007 delka	39.23949	39.23688	11.3	23.0
60		4007 smer	286.204600	286.204764	36.3	74.0
61	4007	4012 delka	39.22365	39.23688	11.3	23.0
62		4012 smer	57.278800	57.278065	36.3	74.0
63		4001 delka	50.58278	50.59825	11.2	22.9
64		4001 smer	259.889600	259.890335	36.3	74.0

## Prehled neznamych - orientacni posuny

\*\*\*\*\*

i	stanovisko	priblizna hodn. [g]	korekce [g]	vyrovnana hodn. [g]	str.ch.	konf.i. +-[cc]
3	4001	314.501200	0.000194	314.501394	31.2	63.6
10	4002	234.616447	0.006500	234.622948	39.0	79.6
15	4003	293.980407	0.001851	293.982258	48.7	99.4
16	4004	349.469329	-0.005712	349.463617	39.1	79.7
19	4008	265.671043	-0.006698	265.664346	46.7	95.2
20	4006	266.688174	-0.002619	266.685554	47.3	96.4
23	4005	93.498000	-0.001574	93.496426	32.2	65.7
24	4009	26.836700	-0.000656	26.836044	59.0	120.4
27	4010	122.116600	-0.000376	122.116224	78.0	159.0
30	4011	244.136034	0.030601	244.166635	91.4	186.4
33	4012	125.541931	0.003544	125.545475	76.2	155.4
34	4007	154.474200	-0.002026	154.472174	57.8	118.0

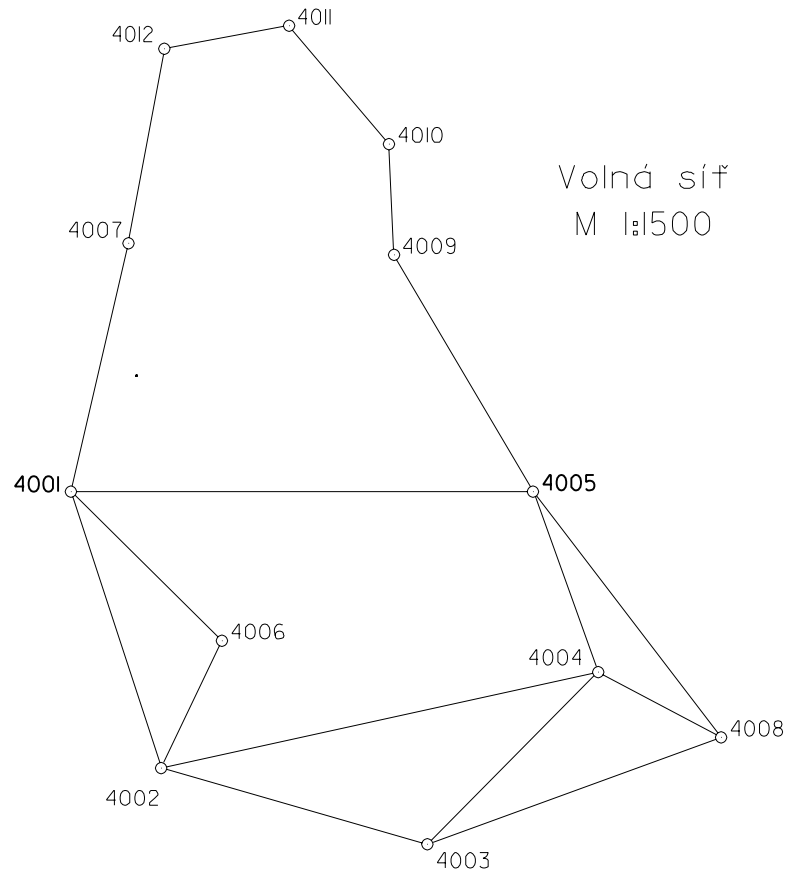
## Prehled neznamych - souradnice

\*\*\*\*\*

i	bod	priblizna hodnota	korekce [m]	vyrovnana hodnota	str.ch.	konf.i. +-[mm]
	4001	typ: 0		mp =	7.3	
1	y	1091.66600	-0.00168	1091.66432	7.3	14.9
2	x	5000.00000	0.00000	5000.00000	0.0	0.0

	4006	typ: U		mp =	9.2	
4	y	1061.72797	0.00147	1061.72943	8.0	16.4
5	x	5029.55509	-0.00555	5029.54954	4.5	9.2
	4002	typ: U		mp =	10.4	
6	y	1073.79216	0.00213	1073.79429	7.9	16.0
7	x	5054.78738	-0.00152	5054.78587	6.8	13.8
	4007	typ: U		mp =	13.7	
8	y	1080.34741	-0.00176	1080.34566	8.3	16.9
9	x	4950.68891	-0.00494	4950.68397	10.9	22.3
	4003	typ: U		mp =	9.4	
11	y	1020.97698	-0.00826	1020.96872	6.5	13.2
12	x	5069.95849	0.00841	5069.96690	6.8	13.9
	4004	typ: U		mp =	7.0	
13	y	987.17825	-0.00474	987.17351	3.1	6.3
14	x	5035.75117	0.00764	5035.75881	6.3	12.8
	4008	typ: U		mp =	9.4	
17	y	962.69069	-0.00384	962.68685	5.2	10.6
18	x	5048.77654	0.00362	5048.78015	7.9	16.0
	4009	typ: U		mp =	12.0	
21	y	1027.66884	-0.00521	1027.66363	6.6	13.5
22	x	4953.05951	0.01042	4953.06993	10.1	20.5
	4010	typ: U		mp =	15.4	
25	y	1028.61853	-0.00542	1028.61312	7.6	15.6
26	x	4931.02467	0.01821	4931.04288	13.4	27.2
	4011	typ: U		mp =	15.9	
28	y	1048.48513	-0.01481	1048.47032	10.2	20.8
29	x	4907.56759	-0.00907	4907.55852	12.2	24.9
	4012	typ: U		mp =	16.3	
31	y	1073.14572	-0.00106	1073.14466	10.2	20.8
32	x	4912.12401	-0.01047	4912.11354	12.7	26.0





Obrázek A.2: Volná síť

## A.3 Vázaná síť

Davkove vyrovnani mistni trigonometricke site verze: 0.6.3 / 0.8.0 / GNU C++  
 \*\*\*\*\*

Vypocet pribliznych souradnic bodu  
 \*\*\*\*\*

Pocet bodu se znamymi souradnicemi: 3  
 Pocet urcovanych bodu : 9  
 Pocet mereni : 69

-----  
 Uspesne vypoctenych bodu : 9  
 Zustalo nevypocteno bodu : 0

Popis site  
 \*\*\*\*\*

Ukazkovy priklad pro program GaMa

Zakladni parametry reseni ulohy  
 \*\*\*\*\*

Pocet urcovanych bodu : 9  
 Pocet pevnych bodu : 3  
 Bodu celkem : 12

Pocet smeru : 46                      Pocet osnov : 12  
 Pocet delek : 23  
 Celkem pozorovani : 69

Pocet rovnic oprav : 69                      Pocet neznamych: 30  
 Pocet nadbyt. pozorovani: 39                      Defekt site : 0  
 m0' [mm/cc] : 9.65                      [pvv] : 3.63033e+03

Pri statisticke analyze se pracuje

- s empirickou jednotkovou stredni chybou 9.65 [mm/cc]  
 - s konfidencni pravdepodobnosti 95 %

Pomer m0' empiricka / m0 apriorni: 0.965  
 95 % interval (0.779, 1.221) obsahuje hodnotu m0'/m0  
 m0'/m0 (delky): 1.000    m0'/m0 (smery): 0.943

Maximalni pokles m0''/m0 pri vyloucení jednoho pozorovani: 0.911

Maximalni studentizovana oprava 2.26  
 presahuje kritickou hodnotu 1.95 na hladine vyznamnosti 5 %  
 35. pozorovani - stanovisko 407 d = 422 346.415 5.0

## Přehled vyrovnaných pozorování

\*\*\*\*\*

i	stanovisko	cíl	merena hodnota	vyrovnana [m/g]	str.ch. +-[mm/cc]	konf.i.
1	1	2 smer	0.000000	0.001194	4.6	9.3
2		422 smer	28.205700	28.205776	5.0	10.2
3		424 smer	60.490600	60.491147	6.1	12.4
4		403 smer	324.366200	324.365051	6.2	12.5
5		407 smer	382.818200	382.817533	5.1	10.2
6		2 delka	845.77700	845.77891	2.7	5.4
7		422 delka	493.79300	493.79988	2.8	5.8
8		424 delka	288.30100	288.30333	2.9	6.0
9		403 delka	388.53600	388.54002	2.8	5.7
10		407 delka	498.75000	498.74642	2.7	5.6
11	2	1 smer	0.000000	399.999127	4.8	9.8
12		407 smer	22.237600	22.238405	5.2	10.6
13		409 smer	73.898400	73.898284	6.5	13.1
14		411 smer	134.209000	134.209163	6.1	12.2
15		416 smer	203.070600	203.070420	6.3	12.7
16		418 smer	287.295100	287.296191	6.8	13.7
17		420 smer	345.692800	345.693402	6.0	12.2
18		422 smer	368.990800	368.989307	5.0	10.1
19		407 delka	388.56200	388.55966	2.5	5.1
20		409 delka	257.49800	257.49660	2.7	5.4
21		411 delka	360.28200	360.28280	2.6	5.3
22		416 delka	338.91900	338.91550	2.6	5.2
23		418 delka	292.09400	292.09952	2.8	5.6
24		420 delka	261.40800	261.40169	2.7	5.4
25		422 delka	452.24900	452.25342	2.6	5.2
26	403	1 smer	0.000000	399.999990	7.4	14.9
27		407 smer	313.554200	313.554210	7.4	14.9
28		407 delka	405.40300	405.40357	2.9	5.8
29	407	1 smer	0.000000	399.999078	5.9	11.9
30		403 smer	55.101300	55.100817	6.2	12.6
31		409 smer	193.341000	193.339946	6.7	13.5
32		2 smer	239.420400	239.422017	5.9	12.0
33		422 smer	323.544300	323.545142	6.1	12.3
34		409 delka	281.99700	282.00007	3.1	6.2
35		422 delka	346.41500	346.40608	2.8	5.6
36	409	2 smer	0.000000	399.999805	7.1	14.3
37		407 smer	102.257500	102.257855	7.8	15.7
38		411 smer	310.175100	310.174940	7.6	15.4
39		411 delka	296.28100	296.28188	3.2	6.4
40	411	2 smer	0.000000	0.000117	6.0	12.2
41		409 smer	49.864700	49.864373	7.0	14.3
42		413 smer	291.495300	291.494823	7.4	15.0
43		416 smer	337.666700	337.667387	6.3	12.6
44		413 delka	252.26600	252.26660	3.1	6.2

45		416 delka	360.44900	360.44942	2.9	5.8
46	413	411 smer	0.000000	399.999835	8.0	16.2
47		416 smer	295.358200	295.358365	8.0	16.2
48		416 delka	239.74500	239.73839	3.3	6.6
49	416	2 smer	0.000000	399.999962	6.1	12.4
50		411 smer	68.806500	68.805974	6.3	12.7
51		413 smer	117.992200	117.991939	7.4	14.9
52		418 smer	348.160600	348.161425	6.9	14.0
53		418 delka	389.39700	389.39329	3.4	6.8
54	418	2 smer	0.000000	399.999535	6.6	13.3
55		416 smer	63.934700	63.935227	7.1	14.4
56		420 smer	336.319000	336.318938	7.5	15.2
57		420 delka	246.59400	246.59734	3.3	6.7
58	420	2 smer	0.000000	399.999845	7.4	14.9
59		418 smer	77.922100	77.922038	7.7	15.5
60		422 smer	250.180400	250.180616	8.1	16.4
61		422 delka	228.20700	228.20785	3.2	6.5
62	422	2 smer	0.000000	399.999334	5.6	11.3
63		420 smer	26.883400	26.884199	6.8	13.7
64		424 smer	225.796400	225.795774	6.8	13.7
65		1 smer	259.212400	259.213736	5.6	11.4
66		407 smer	337.372400	337.371556	5.9	11.9
67		424 delka	279.40500	279.40150	2.9	5.8
68	424	1 smer	0.000000	399.999416	7.9	15.9
69		422 smer	134.295500	134.296084	7.9	15.9

## Prehled neznamych - orientacni posuny

\*\*\*\*\*

i	stanovisko	priblizna hodn. [g]	korekce [g]	vyrovnana hodn. [g]	str.ch. +-[cc]	konf.i. [g]
1	1	296.483127	0.000061	296.483188	5.4	11.0
10	2	96.483567	0.001687	96.485254	4.4	8.9
21	403	20.848300	-0.000051	20.848249	7.6	15.3
22	407	79.301008	0.000634	79.301642	5.0	10.1
23	409	370.381654	0.002079	370.383733	6.7	13.6
24	411	30.694545	-0.000245	30.694300	6.5	13.1
25	413	122.188700	0.000588	122.189288	8.9	18.1
26	416	99.555965	-0.000252	99.555713	6.4	13.0
27	418	183.780849	0.001062	183.781911	7.6	15.3
28	420	242.177749	0.001061	242.178811	6.8	13.8
29	422	265.474138	0.001090	265.475227	5.3	10.8
30	424	156.975200	-0.000282	156.974918	8.0	16.3

## Prehled neznamych - souradnice

\*\*\*\*\*

i	bod	priblizna	korekce	vyrovnana	str.ch.	konf.i.
---	-----	-----------	---------	-----------	---------	---------

```

===== hodnota ===== [m] ===== hodnota ===== +/-[mm] =====
      1          typ: U                      mp =      3.9
2      y 644498.59274 -0.00050 644498.59224 2.5 5.0
3      x 1054980.48229 0.00371 1054980.48600 3.0 6.0

      2          typ: U                      mp =      4.1
4      y 643654.10589 -0.00325 643654.10264 2.6 5.2
5      x 1054933.78432 0.01879 1054933.80311 3.2 6.4

     422          typ: U                      mp =      3.9
6      y 644041.46181 0.00101 644041.46283 2.9 5.8
7      x 1055167.21962 0.00424 1055167.22385 2.6 5.3

     407          typ: U                      mp =      4.0
8      y 644025.97814 -0.00045 644025.97769 2.8 5.6
9      x 1054821.16271 0.00135 1054821.16405 2.9 5.8

     409          typ: U                      mp =      4.8
11     y 643769.62669 -0.00738 643769.61930 3.3 6.6
12     x 1054703.66181 0.01003 1054703.67184 3.5 7.1

     411          typ: U                      mp =      4.2
13     y 643487.05410 -0.00821 643487.04589 3.0 6.1
14     x 1054614.58480 0.00720 1054614.59200 2.9 5.9

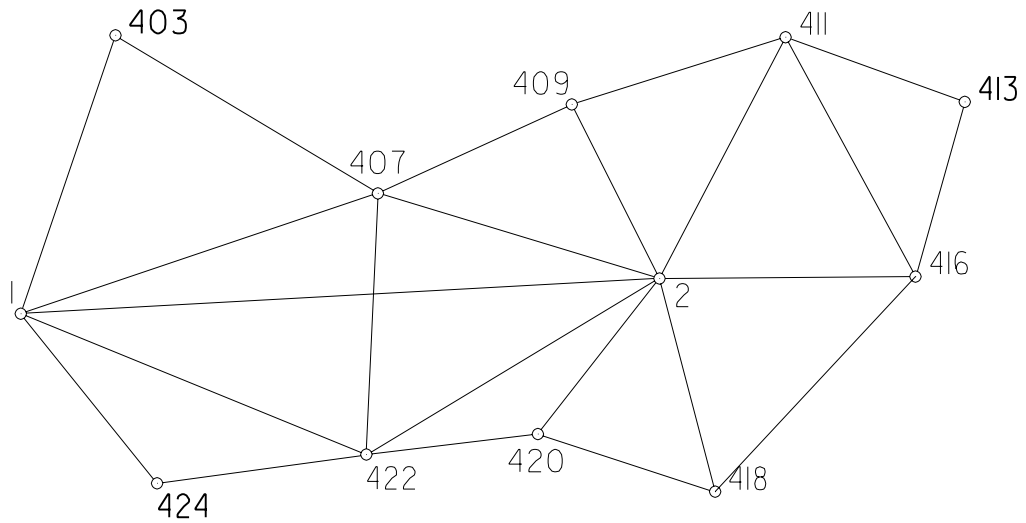
     416          typ: U                      mp =      4.2
15     y 643315.19521 0.00018 643315.19539 2.7 5.4
16     x 1054931.42905 0.00864 1054931.43769 3.3 6.6

     418          typ: U                      mp =      5.6
17     y 643580.48293 0.00666 643580.48959 4.1 8.3
18     x 1055216.46488 0.00986 1055216.47474 3.9 7.9

     420          typ: U                      mp =      4.9
19     y 643814.89687 -0.00028 643814.89659 3.6 7.2
20     x 1055139.89249 0.00806 1055139.90054 3.3 6.8

```

Vázaná síť  
M 1:10000



Obrázek A.3: Vázaná síť

## A.4 Vázaná síť – zhoršená konfigurace

Davkove vyrovnani mistni trigonometricke site verze: 0.6.3 / 0.8.0 / GNU C++  
 \*\*\*\*\*

Vypocet pribliznych souradnic bodu  
 \*\*\*\*\*

Pocet bodu se znamymi souradnicemi: 5  
 Pocet urcovanych bodu : 34  
 Pocet mereni : 181

-----  
 Uspesne vypoctenych bodu : 34  
 Zustalo nevypocteno bodu : 0

Popis site  
 \*\*\*\*\*

Drazni teleso byvale vlecky - Libusin

včetně výpočtu přibližných souřadnic

Zakladni parametry reseni ulohy  
 \*\*\*\*\*

Pocet urcovanych bodu : 34  
 Pocet pevnych bodu : 5  
 Bodu celkem : 39

Pocet smeru : 92 Pocet osnov : 34  
 Pocet delek : 89  
 Celkem pozorovani : 181

Pocet rovnic oprav : 181 Pocet neznamych: 102  
 Pocet nadbyt. pozorovani: 79 Defekt site : 0  
 m0' [mm/cc] : 17.69 [pvv] : 2.47168e+04

Pri statisticke analyze se pracuje

- s empirickou jednotkovou střední chybou 17.69 [mm/cc]
- s konfidencní pravděpodobností 95 %

Pomer m0' empiricka / m0 apriorni: 1.769  
 95 % interval (0.844, 1.155) neobsahuje hodnotu m0'/m0  
 m0'/m0 (delky): 1.322 m0'/m0 (smery): 2.729

Maximalni pokles m0''/m0 pri vyloucení jednoho pozorovani: 1.529

Maximalni studentizovana oprava 4.55  
 presahuje kritickou hodnotu 1.95 na hladine vyznamnosti 5 %

136. pozorovani - stanovisko 4027 s = 4026 105.7272 20.0

Prehled vyrovnanych pozorovani

\*\*\*\*\*

i	stanovisko	cil	merena hodnota	vyrovnana [m/g]	str.ch. +-[mm/cc]	konf.i.
1	4001	4002 delka	115.31970	115.32793	10.8	21.6
2		4002 smer	320.032800	320.032082	32.5	64.7
3		4003 delka	110.23020	110.22986	8.5	17.0
4		4003 smer	175.013000	175.017282	24.6	49.0
5		4004 delka	196.77160	196.76913	8.3	16.4
6		4004 smer	156.042500	156.043749	24.9	49.6
7		4005 delka	28.48399	28.48846	8.2	16.3
8		4005 smer	163.881700	163.880840	34.2	68.0
9		4007 delka	227.39840	227.39713	10.8	21.4
10		4007 smer	237.535500	237.531546	27.8	55.3
11	4005	4003 delka	82.32893	82.32561	8.6	17.2
12		4003 smer	257.010000	257.009037	29.3	58.4
13		4001 delka	28.48762	28.48846	8.2	16.3
14		4008 delka	116.62890	116.62760	12.5	24.9
15		4008 smer	241.826500	241.826500	35.4	70.4
16		4004 delka	168.54160	168.53257	8.7	17.3
17		4004 smer	232.876600	232.877563	29.3	58.4
18	4008	4005 delka	116.62630	116.62760	12.5	24.9
19		4005 smer	382.542200	382.542200	35.4	70.4
20	4002	4009 delka	90.46238	90.46818	11.2	22.3
21		4009 smer	180.815400	180.814682	34.5	68.7
22		4001 delka	115.32190	115.32793	10.8	21.6
23		4001 smer	398.712900	398.713618	34.5	68.7
24	4009	4002 delka	90.46028	90.46818	11.2	22.3
25		4002 smer	353.976900	353.977845	34.5	68.7
26		3960032 delka	110.73280	110.73718	13.3	26.5
27		3960032 smer	252.155000	252.154055	34.5	68.7
28	4010	4003 delka	53.62352	53.62472	12.5	24.9
29		4003 smer	275.566700	275.566700	35.4	70.4
30	4003	4001 delka	110.22710	110.22986	8.5	17.0
31		4001 smer	350.625300	350.623982	24.7	49.2
32		4005 delka	82.32125	82.32561	8.6	17.2
33		4005 smer	354.462000	354.460402	25.7	51.1
34		4010 delka	53.62592	53.62472	12.5	24.9
35		4010 smer	347.401800	347.401800	35.4	70.4
36		4011 delka	50.69680	50.70384	8.6	17.1
37		4011 smer	165.323500	165.326416	31.3	62.3
38	4011	4003 delka	50.70611	50.70384	8.6	17.1
39		4003 smer	251.359200	251.356033	31.7	63.0
40		10600 delka	66.15172	66.17219	8.0	15.9
41		10600 smer	47.807200	47.805676	22.3	44.4
42		10601 delka	82.08559	82.04758	8.0	15.9



43		10601 smer	50.915600	50.919187	18.7	37.2
44		4012 delka	65.99001	65.99126	10.9	21.7
45		4012 smer	82.609400	82.611301	31.3	62.3
46		4013 delka	70.94774	70.96019	8.0	16.0
47		4013 smer	51.322000	51.323121	19.5	38.7
48		4014 delka	31.32846	31.32523	9.5	18.9
49		4014 smer	377.202000	377.200083	31.7	63.2
50	4014	4011 delka	31.31405	31.32523	9.5	18.9
51		4011 smer	35.211200	35.213282	31.8	63.3
52		4004 delka	45.29522	45.29242	8.9	17.8
53		4004 smer	208.665200	208.663118	31.8	63.3
54	4004	13090032 smer	34.456100	34.453786	24.3	48.3
55		4001 delka	196.76700	196.76913	8.3	16.4
56		4001 smer	203.307100	203.307936	21.5	42.9
57		4005 delka	168.53070	168.53257	8.7	17.3
58		4005 smer	201.987500	201.986416	22.7	45.2
59		4014 delka	45.28574	45.29242	8.9	17.8
60		4014 smer	136.275400	136.277790	30.8	61.4
61		4015 delka	150.39700	150.39728	11.2	22.3
62		4015 smer	38.918100	38.918271	27.8	55.4
63	4015	4004 delka	150.38470	150.39728	11.2	22.3
64		4004 smer	184.421200	184.422451	29.7	59.0
65		13090032 smer	379.582000	379.578736	25.2	50.2
66		4016 delka	135.35630	135.35961	11.2	22.2
67		4016 smer	374.474000	374.476013	29.2	58.1
68	4016	4015 delka	135.35280	135.35961	11.2	22.2
69		4015 smer	130.108600	130.108270	29.2	58.2
70		13090032 smer	335.634600	335.633192	27.5	54.8
71		13090035 delka	91.07367	91.04179	9.4	18.7
72		4017 delka	79.89514	79.89428	11.1	22.0
73		4017 smer	18.181100	18.182838	31.4	62.5
74	4017	4016 delka	79.90071	79.89428	11.1	22.0
75		4016 smer	8.937600	8.935018	31.8	63.3
76		4018 delka	60.15739	60.15168	11.1	22.2
77		4018 smer	201.376400	201.378982	31.8	63.3
78	4007	4001 delka	227.41090	227.39713	10.8	21.4
79		4001 smer	64.566400	64.567998	31.6	62.9
80		4019 delka	177.76350	177.76681	10.5	20.9
81		4019 smer	182.267200	182.265602	31.6	62.9
82	4012	4011 delka	66.00607	65.99126	10.9	21.7
83		4011 smer	314.141100	314.139288	33.0	65.7
84		4020 delka	26.98248	26.97816	10.9	21.7
85		4020 smer	110.753800	110.755612	33.0	65.7
86	4020	4012 delka	26.98749	26.97816	10.9	21.7
87		4012 smer	334.718000	334.716212	33.2	66.0
88		4021 delka	53.19481	53.19213	9.2	18.4
89		4021 smer	165.456100	165.459119	31.9	63.5
90		4031 delka	30.43800	30.44128	11.0	22.0
91		4031 smer	146.538200	146.536969	32.4	64.4
92	4021	4020 delka	53.19654	53.19213	9.2	18.4
93		4020 smer	213.580600	213.577875	32.7	65.0

94		4019 delka	45.86587	45.86372	10.2	20.4
95		4019 smer	360.303500	360.306225	32.7	65.0
96	4019	4007 delka	177.75840	177.76681	10.5	20.9
97		4007 smer	103.654400	103.653876	30.9	61.5
98		4021 delka	45.87286	45.86372	10.2	20.4
99		4021 smer	143.490000	143.487293	31.9	63.6
100		4022 delka	67.22816	67.21650	10.3	20.4
101		4022 smer	346.683500	346.686731	32.4	64.4
102	4022	4019 delka	67.21881	67.21650	10.3	20.4
103		4019 smer	40.269700	40.265584	30.5	60.6
104		4023 delka	46.63353	46.64653	10.4	20.7
105		4023 smer	112.619300	112.623941	32.6	64.9
106		4024 delka	92.24250	92.24236	8.8	17.6
107		4024 smer	173.700800	173.699365	31.4	62.5
108		4033 delka	40.80000	40.80132	10.7	21.2
109		4033 smer	81.518000	81.518910	31.6	62.9
110	4024	4022 delka	92.24095	92.24236	8.8	17.6
111		4022 smer	241.070600	241.072229	31.4	62.4
112		4025 delka	30.92414	30.91681	10.0	19.8
113		4025 smer	328.118400	328.115851	32.6	64.8
114		4018 delka	65.54380	65.54182	11.3	22.6
115		4018 smer	118.148000	118.148920	32.4	64.5
116	4018	4024 delka	65.54804	65.54182	11.3	22.6
117		4024 smer	286.993400	286.993251	31.3	62.3
118		4017 delka	60.15122	60.15168	11.1	22.2
119		4017 smer	340.450200	340.446954	31.3	62.4
120		13090032 smer	60.162300	60.159711	26.8	53.4
121		4034 delka	20.39300	20.40968	5.0	10.0
122		4034 smer	379.749000	379.748986	35.3	70.3
123		4036 delka	49.27700	49.27955	9.8	19.6
124		4036 smer	61.089500	61.095498	30.0	59.7
125	4025	4024 delka	30.91819	30.91681	10.0	19.8
126		4024 smer	179.706400	179.709345	33.0	65.7
127		4026 delka	25.68594	25.67723	9.9	19.8
128		4026 smer	377.300000	377.297055	33.0	65.7
129	4026	4025 delka	25.67799	25.67723	9.9	19.8
130		4025 smer	84.441600	84.444867	33.1	65.9
131		4027 delka	15.28554	15.28772	10.5	20.8
132		4027 smer	152.741200	152.735483	33.0	65.7
133		4028 delka	24.93589	24.93355	11.3	22.6
134		4028 smer	250.084000	250.086450	33.3	66.3
135	4027	4026 delka	15.26083	15.28772	10.5	20.8
136		4026 smer	105.727200	105.732784	33.2	66.0
137		4023 delka	42.31678	42.31928	9.9	19.7
138		4023 smer	342.674900	342.669316	33.2	66.0
139	4023	4027 delka	42.30505	42.31928	9.9	19.7
140		4027 smer	43.111000	43.115852	33.2	66.2
141		4022 delka	46.62813	46.64653	10.4	20.7
142		4022 smer	192.186800	192.181948	33.2	66.2
143	4028	4026 delka	24.93700	24.93355	11.3	22.6
144		4026 smer	208.287000	208.285208	33.7	67.1

145		4029 delka	34.55282	34.55598	11.4	22.7
146		4029 smer	17.975700	17.977492	33.7	67.1
147	4029	4028 delka	34.55849	34.55598	11.4	22.7
148		4028 smer	180.463600	180.462727	34.0	67.6
149		4030 delka	64.24385	64.23882	11.4	22.7
150		4030 smer	375.000500	375.001373	34.0	67.6
151	4030	4029 delka	64.23676	64.23882	11.4	22.7
152		4029 smer	219.261200	219.262030	33.5	66.8
153		4013 delka	38.55667	38.55819	11.4	22.7
154		4013 smer	33.319000	33.318170	33.5	66.8
155	4013	4030 delka	38.55336	38.55819	11.4	22.7
156		4030 smer	183.424400	183.426244	32.7	65.0
157		10601 delka	11.15467	11.09795	1.7	3.4
158		10601 smer	169.869700	169.869878	35.0	69.7
159		10600 delka	6.10854	6.10375	1.9	3.7
160		10600 smer	13.719400	13.719901	35.3	70.3
161		4011 delka	70.94656	70.96019	8.0	16.0
162		4011 smer	372.859800	372.857278	33.1	65.8
163	4031	4020 delka	30.44834	30.44128	11.0	22.0
164		4020 smer	208.024000	208.025185	33.3	66.2
165		4032 delka	42.09038	42.08860	10.8	21.5
166		4032 smer	371.996400	371.995215	33.3	66.2
167	4032	4031 delka	42.08873	42.08860	10.8	21.5
168		4031 smer	286.177600	286.178677	33.7	67.1
169		4033 delka	49.22905	49.22870	11.0	21.8
170		4033 smer	90.592700	90.591623	33.7	67.1
171	4033	4032 delka	49.23052	49.22870	11.0	21.8
172		4032 smer	95.263700	95.264656	33.1	65.8
173		4022 delka	40.80677	40.80132	10.7	21.2
174		4022 smer	336.985900	336.984944	33.1	65.8
175	4034	4018 delka	20.40200	20.40968	5.0	10.0
176		4018 smer	277.335200	277.335200	35.4	70.4
177	4036	4018 delka	49.28800	49.27955	9.8	19.6
178		4018 smer	59.645400	59.640318	30.0	59.7
179		4034 delka	47.58600	47.58176	9.5	18.9
180		4034 smer	86.581200	86.580482	34.7	69.1
181		13090032 smer	258.668900	258.674700	30.4	60.4

Prehled neznamych - orientacni posuny

\*\*\*\*\*

i	stanovisko	priblizna hodn. [g]	korekce [g]	vyrovnana hodn. [g]	str.ch.	konf.i. +-[cc]
5	4001	81.283231	-0.035802	81.247428	29.6	59.0
14	4005	3.126034	-0.033939	3.092094	37.9	75.4
17	4008	62.410334	-0.033939	62.376394	62.8	124.9
20	4002	202.559246	0.006646	202.565892	42.0	83.6
21	4009	229.337543	0.065186	229.402729	53.7	106.9
24	4010	377.515902	-0.040073	377.475829	61.6	122.6

25	4003	105.675876	-0.035147	105.640729	36.0	71.6
28	4011	219.654954	-0.043842	219.611112	35.0	69.7
35	4014	361.632390	-0.034477	361.597913	42.6	84.7
36	4004	234.008826	-0.025585	233.983241	24.3	48.4
39	4015	288.486669	-0.007608	288.479061	24.8	49.4
42	4016	332.848021	-0.001217	332.846804	26.8	53.4
45	4017	142.092665	0.001959	142.094624	40.0	79.5
48	4007	54.248207	-0.037231	54.210976	39.2	78.0
51	4012	188.123254	-0.040129	188.083125	46.7	93.0
54	4020	164.159054	-0.036529	164.122525	47.2	93.9
59	4021	316.034554	-0.030785	316.003769	49.5	98.6
60	4019	332.859600	-0.036898	332.822702	40.0	79.6
63	4022	39.275774	-0.031926	39.243849	41.4	82.4
70	4024	171.870296	0.000689	171.870984	39.1	77.9
73	4018	203.020009	0.006644	203.026653	25.6	51.0
78	4025	120.282296	-0.004805	120.277490	51.2	101.9
81	4026	213.140696	-0.011018	213.129678	51.4	102.3
86	4027	60.155136	-0.022759	60.132377	58.4	116.3
87	4023	159.711575	-0.025733	159.685842	55.6	110.6
88	4028	54.935459	-0.004538	54.930921	53.7	106.9
91	4029	92.469424	-0.023737	92.445686	51.3	102.0
94	4030	48.222554	-0.037525	48.185029	48.2	96.0
95	4013	98.117154	-0.040199	98.076955	41.5	82.5
96	4031	302.673254	-0.038945	302.634309	53.7	106.9
99	4032	188.492054	-0.041207	188.450847	53.6	106.8
100	4033	383.812903	-0.035089	383.777814	51.5	102.5
101	4034	105.433809	0.006630	105.440439	56.2	111.8
102	4036	4.485318	-0.003484	4.481833	29.7	59.1

## Přehled neznámých - souradnice

\*\*\*\*\*

i	bod	priblizna	korekce	vyrovnana	str.ch.	konf.i.
=====	=====	hodnota	==== [m]	====	hodnota	==== +- [mm] ===
	4001	typ: U		mp =	14.2	
1	y	766777.34277	-0.06314	766777.27963	9.9	19.7
2	x	1030029.09921	0.12888	1030029.22809	10.2	20.3
	4002	typ: U		mp =	16.3	
3	y	766779.72653	-0.12913	766779.59739	11.2	22.4
4	x	1030144.39537	0.13738	1030144.53275	11.8	23.6
	4003	typ: U		mp =	13.0	
6	y	766692.09148	-0.03686	766692.05462	10.3	20.5
7	x	1029959.23883	0.07986	1029959.31869	7.9	15.6
	4004	typ: U		mp =	12.3	
8	y	766668.48202	0.01576	766668.49778	8.5	16.8
9	x	1029865.21480	0.04802	1029865.26281	8.9	17.7

	4005	typ: U		mp =	14.4	
10	y	766758.78588	-0.05164	766758.73424	10.2	20.4
11	x	1030007.48266	0.11995	1030007.60261	10.2	20.3
	4007	typ: U		mp =	17.6	
12	y	766559.79811	-0.09394	766559.70417	14.6	29.1
13	x	1030095.33707	-0.00005	1030095.33703	9.9	19.7
	4008	typ: U		mp =	19.6	
15	y	766683.10846	-0.00433	766683.10413	13.9	27.7
16	x	1029918.74189	0.07960	1029918.82149	13.8	27.5
	4009	typ: U		mp =	16.8	
18	y	766756.30125	-0.05388	766756.24737	12.1	24.1
19	x	1030231.80556	0.13012	1030231.93568	11.6	23.2
	4010	typ: U		mp =	19.0	
22	y	766731.80078	-0.05954	766731.74124	14.2	28.3
23	x	1029995.27744	0.10486	1029995.38230	12.5	25.0
	4011	typ: U		mp =	9.1	
26	y	766646.55522	-0.02254	766646.53268	7.5	15.0
27	x	1029936.94339	0.04519	1029936.98858	5.2	10.3
	4012	typ: U		mp =	13.8	
29	y	766580.59892	-0.01731	766580.58161	12.9	25.7
30	x	1029939.29034	0.00150	1029939.29184	5.1	10.1
	4013	typ: U		mp =	1.9	
31	y	766582.85402	-0.01328	766582.84074	1.9	3.8
32	x	1029905.70790	-0.00330	1029905.70460	0.3	0.7
	4014	typ: U		mp =	12.9	
33	y	766648.10095	0.00014	766648.10109	7.4	14.8
34	x	1029905.66030	0.04233	1029905.70263	10.6	21.1
	4015	typ: U		mp =	16.4	
37	y	766531.48644	0.03467	766531.52110	13.1	26.1
38	x	1029803.16951	-0.00933	1029803.16019	9.9	19.7
	4016	typ: U		mp =	17.9	
40	y	766418.39778	0.04123	766418.43901	14.7	29.3
41	x	1029728.78410	-0.01925	1029728.76485	10.2	20.3
	4017	typ: U		mp =	19.5	
43	y	766362.82204	0.04424	766362.86628	15.6	31.1
44	x	1029786.18633	-0.02141	1029786.16492	11.7	23.2
	4018	typ: U		mp =	19.7	
46	y	766316.15487	0.04830	766316.20317	15.7	31.3

47	x	1029824.14308	-0.02057	1029824.12250	11.9	23.8
	4019	typ: U		mp =	15.9	
49	y	766463.33799	-0.00699	766463.33100	12.9	25.7
50	x	1029946.01881	-0.05803	1029945.96078	9.4	18.7
	4020	typ: U		mp =	15.3	
52	y	766553.61813	-0.01020	766553.60793	14.2	28.2
53	x	1029938.81437	-0.01462	1029938.79975	5.7	11.4
	4021	typ: U		mp =	15.9	
55	y	766506.07531	-0.01959	766506.05572	13.9	27.6
56	x	1029962.67774	-0.04124	1029962.63650	7.9	15.6
	4031	typ: U		mp =	17.7	
57	y	766523.60373	-0.01136	766523.59238	16.4	32.6
58	x	1029943.90577	-0.03274	1029943.87302	6.8	13.6
	4022	typ: U		mp =	17.4	
61	y	766399.56062	0.00563	766399.56625	13.1	26.0
62	x	1029924.80210	-0.10436	1029924.69775	11.4	22.7
	4023	typ: U		mp =	18.6	
64	y	766431.53594	0.03260	766431.56854	13.9	27.7
65	x	1029890.86078	-0.10051	1029890.76027	12.3	24.5
	4024	typ: U		mp =	17.8	
66	y	766380.89007	0.05120	766380.94127	14.2	28.2
67	x	1029834.40338	-0.04812	1029834.35526	10.8	21.4
	4033	typ: U		mp =	18.8	
68	y	766438.20539	0.01138	766438.21677	15.6	31.2
69	x	1029911.71286	-0.08743	1029911.62543	10.4	20.6
	4025	typ: U		mp =	19.0	
71	y	766411.81123	0.04685	766411.85808	16.1	32.1
72	x	1029834.40887	-0.04722	1029834.36166	10.0	19.9
	4034	typ: U		mp =	20.5	
74	y	766321.60857	0.04951	766321.65808	15.8	31.5
75	x	1029804.48818	-0.03288	1029804.45530	13.0	25.9
	4036	typ: U		mp =	22.2	
76	y	766274.49424	0.05024	766274.54447	17.8	35.4
77	x	1029797.83141	-0.03433	1029797.79708	13.3	26.5
	4026	typ: U		mp =	17.1	
79	y	766437.47468	0.04200	766437.51668	14.3	28.5
80	x	1029835.38397	-0.04427	1029835.33970	9.4	18.7
	4027	typ: U		mp =	19.4	

---

82	y	766429.65858	0.04815	766429.70673	14.6	29.0
83	x	1029848.54812	-0.06616	1029848.48197	12.8	25.5
	4028	typ: U		mp =	17.4	
84	y	766458.38465	0.01802	766458.40268	14.7	29.2
85	x	1029849.01866	-0.06070	1029848.95796	9.4	18.8
	4029	typ: U		mp =	15.9	
89	y	766489.86691	0.00971	766489.87662	13.6	27.1
90	x	1029863.28158	-0.05804	1029863.22353	8.2	16.3
	4030	typ: U		mp =	11.8	
92	y	766545.90831	-0.00969	766545.89862	11.0	21.9
93	x	1029894.68506	-0.02646	1029894.65859	4.3	8.5
	4032	typ: U		mp =	19.1	
97	y	766484.80216	-0.00020	766484.80196	16.9	33.5
98	x	1029927.59723	-0.05683	1029927.54039	8.9	17.8

Davkove vyrovnani mistni trigonometricke site verze: 0.6.3 / 0.8.0 / GNU C++  
 \*\*\*\*\*

Popis site

\*\*\*\*\*

Drazni teleso byvale vlecky - Libusin

priblizne souradnice - vystup z predchoziho vyrovnani

Zakladni parametry reseni ulohy

\*\*\*\*\*

Pocet urcovanych bodu : 34  
 Pocet pevnych bodu : 5  
 Bodu celkem : 39

Pocet smeru : 92                      Pocet osnov : 34  
 Pocet delek : 89  
 Celkem pozorovani : 181

Pocet rovnic oprav : 181                      Pocet neznamych: 102  
 Pocet nadbyt. pozorovani: 79                      Defekt site : 0  
 m0' [mm/cc] : 17.69                      [pvv] : 2.47218e+04

Pri statisticke analyze se pracuje

- s empirickou jednotkovou stredni chybou 17.69 [mm/cc]  
 - s konfidencni pravdepodobnosti 95 %

Pomer m0' empiricka / m0 apriorni: 1.769  
 95 % interval (0.844, 1.155) neobsahuje hodnotu m0'/m0  
 m0'/m0 (delky): 1.323    m0'/m0 (smery): 2.730

Maximalni pokles m0''/m0 pri vyloucení jednoho pozorovani: Inf

Maximalni studentizovana oprava Inf

presahuje kritickou hodnotu 1.95 na hladine vyznamnosti 5 %

176. pozorovani - stanovisko 4034 s = 4018 277.3352 20.0

Prehled vyrovnanych pozorovani

\*\*\*\*\*

i	stanovisko	cil	merena hodnota	vyrovnana [m/g]	str.ch. [mm/cc]	konf.i. +- [mm/cc]
1	4001	4002 delka	115.31970	115.32797	10.9	21.6
2		4002 smer	320.032800	320.032088	32.5	64.8
3		4003 delka	110.23020	110.22987	8.5	17.0



4		4003 smer	175.013000	175.017280	24.6	49.0
5		4004 delka	196.77160	196.76914	8.3	16.4
6		4004 smer	156.042500	156.043749	24.9	49.6
7		4005 delka	28.48399	28.48847	8.2	16.3
8		4005 smer	163.881700	163.880841	34.2	68.0
9		4007 delka	227.39840	227.39713	10.8	21.4
10		4007 smer	237.535500	237.531542	27.8	55.3
11	4005	4003 delka	82.32893	82.32562	8.6	17.2
12		4003 smer	257.010000	257.009036	29.3	58.4
13		4001 delka	28.48762	28.48847	8.2	16.3
14		4008 delka	116.62890	116.62760	12.5	24.9
15		4008 smer	241.826500	241.826500	35.4	70.4
16		4004 delka	168.54160	168.53258	8.7	17.3
17		4004 smer	232.876600	232.877564	29.3	58.4
18	4008	4005 delka	116.62630	116.62760	12.5	24.9
19		4005 smer	382.542200	382.542200	35.4	70.4
20	4002	4009 delka	90.46238	90.46821	11.2	22.3
21		4009 smer	180.815400	180.814682	34.5	68.7
22		4001 delka	115.32190	115.32797	10.9	21.6
23		4001 smer	398.712900	398.713618	34.5	68.7
24	4009	4002 delka	90.46028	90.46821	11.2	22.3
25		4002 smer	353.976900	353.977849	34.5	68.7
26		3960032 delka	110.73280	110.73725	13.3	26.5
27		3960032 smer	252.155000	252.154051	34.5	68.7
28	4010	4003 delka	53.62352	53.62472	12.5	24.9
29		4003 smer	275.566700	275.566700	35.4	70.4
30	4003	4001 delka	110.22710	110.22987	8.5	17.0
31		4001 smer	350.625300	350.623982	24.7	49.2
32		4005 delka	82.32125	82.32562	8.6	17.2
33		4005 smer	354.462000	354.460402	25.7	51.1
34		4010 delka	53.62592	53.62472	12.5	24.9
35		4010 smer	347.401800	347.401800	35.4	70.4
36		4011 delka	50.69680	50.70385	8.6	17.1
37		4011 smer	165.323500	165.326415	31.3	62.3
38	4011	4003 delka	50.70611	50.70385	8.6	17.1
39		4003 smer	251.359200	251.356033	31.7	63.0
40		10600 delka	66.15172	66.17222	8.0	15.9
41		10600 smer	47.807200	47.805677	22.3	44.4
42		10601 delka	82.08559	82.04760	8.0	15.9
43		10601 smer	50.915600	50.919186	18.7	37.2
44		4012 delka	65.99001	65.99126	10.9	21.7
45		4012 smer	82.609400	82.611301	31.3	62.3
46		4013 delka	70.94774	70.96022	8.0	16.0
47		4013 smer	51.322000	51.323122	19.5	38.7
48		4014 delka	31.32846	31.32522	9.5	18.9
49		4014 smer	377.202000	377.200081	31.7	63.2
50	4014	4011 delka	31.31405	31.32522	9.5	18.9
51		4011 smer	35.211200	35.213282	31.8	63.3
52		4004 delka	45.29522	45.29243	8.9	17.8
53		4004 smer	208.665200	208.663118	31.8	63.3
54	4004	13090032 smer	34.456100	34.453784	24.3	48.3

55		4001 delka	196.76700	196.76914	8.3	16.4
56		4001 smer	203.307100	203.307939	21.6	42.9
57		4005 delka	168.53070	168.53258	8.7	17.3
58		4005 smer	201.987500	201.986418	22.7	45.2
59		4014 delka	45.28574	45.29243	8.9	17.8
60		4014 smer	136.275400	136.277788	30.8	61.4
61		4015 delka	150.39700	150.39727	11.2	22.3
62		4015 smer	38.918100	38.918270	27.8	55.4
63	4015	4004 delka	150.38470	150.39727	11.2	22.3
64		4004 smer	184.421200	184.422451	29.7	59.0
65		13090032 smer	379.582000	379.578736	25.2	50.2
66		4016 delka	135.35630	135.35961	11.2	22.2
67		4016 smer	374.474000	374.476012	29.2	58.1
68	4016	4015 delka	135.35280	135.35961	11.2	22.2
69		4015 smer	130.108600	130.108270	29.2	58.2
70		13090032 smer	335.634600	335.633192	27.5	54.8
71		13090035 delka	91.07367	91.04181	9.4	18.7
72		4017 delka	79.89514	79.89428	11.1	22.0
73		4017 smer	18.181100	18.182838	31.4	62.6
74	4017	4016 delka	79.90071	79.89428	11.1	22.0
75		4016 smer	8.937600	8.935017	31.8	63.3
76		4018 delka	60.15739	60.15168	11.1	22.2
77		4018 smer	201.376400	201.378983	31.8	63.3
78	4007	4001 delka	227.41090	227.39713	10.8	21.4
79		4001 smer	64.566400	64.567998	31.6	62.9
80		4019 delka	177.76350	177.76683	10.5	20.9
81		4019 smer	182.267200	182.265602	31.6	62.9
82	4012	4011 delka	66.00607	65.99126	10.9	21.7
83		4011 smer	314.141100	314.139288	33.0	65.7
84		4020 delka	26.98248	26.97816	10.9	21.7
85		4020 smer	110.753800	110.755612	33.0	65.7
86	4020	4012 delka	26.98749	26.97816	10.9	21.7
87		4012 smer	334.718000	334.716212	33.2	66.1
88		4021 delka	53.19481	53.19213	9.2	18.4
89		4021 smer	165.456100	165.459119	31.9	63.5
90		4031 delka	30.43800	30.44127	11.0	22.0
91		4031 smer	146.538200	146.536968	32.4	64.4
92	4021	4020 delka	53.19654	53.19213	9.2	18.4
93		4020 smer	213.580600	213.577875	32.7	65.0
94		4019 delka	45.86587	45.86372	10.2	20.4
95		4019 smer	360.303500	360.306225	32.7	65.0
96	4019	4007 delka	177.75840	177.76683	10.5	20.9
97		4007 smer	103.654400	103.653876	30.9	61.5
98		4021 delka	45.87286	45.86372	10.2	20.4
99		4021 smer	143.490000	143.487293	31.9	63.6
100		4022 delka	67.22816	67.21650	10.3	20.4
101		4022 smer	346.683500	346.686731	32.4	64.4
102	4022	4019 delka	67.21881	67.21650	10.3	20.4
103		4019 smer	40.269700	40.265583	30.5	60.7
104		4023 delka	46.63353	46.64653	10.4	20.7
105		4023 smer	112.619300	112.623942	32.6	64.9

106		4024 delka	92.24250	92.24237	8.8	17.6
107		4024 smer	173.700800	173.699365	31.4	62.5
108		4033 delka	40.80000	40.80132	10.7	21.2
109		4033 smer	81.518000	81.518909	31.6	62.9
110	4024	4022 delka	92.24095	92.24237	8.8	17.6
111		4022 smer	241.070600	241.072229	31.4	62.4
112		4025 delka	30.92414	30.91681	10.0	19.8
113		4025 smer	328.118400	328.115851	32.6	64.8
114		4018 delka	65.54380	65.54183	11.3	22.6
115		4018 smer	118.148000	118.148920	32.4	64.5
116	4018	4024 delka	65.54804	65.54183	11.3	22.6
117		4024 smer	286.993400	286.993252	31.3	62.3
118		4017 delka	60.15122	60.15168	11.1	22.2
119		4017 smer	340.450200	340.446953	31.3	62.4
120		13090032 smer	60.162300	60.159711	26.8	53.4
121		4034 delka	20.39300	20.40968	5.0	10.0
122		4034 smer	379.749000	379.748986	35.3	70.3
123		4036 delka	49.27700	49.27955	9.8	19.6
124		4036 smer	61.089500	61.095498	30.0	59.7
125	4025	4024 delka	30.91819	30.91681	10.0	19.8
126		4024 smer	179.706400	179.709344	33.0	65.7
127		4026 delka	25.68594	25.67723	9.9	19.8
128		4026 smer	377.300000	377.297056	33.0	65.7
129	4026	4025 delka	25.67799	25.67723	9.9	19.8
130		4025 smer	84.441600	84.444867	33.1	65.9
131		4027 delka	15.28554	15.28772	10.5	20.8
132		4027 smer	152.741200	152.735483	33.0	65.7
133		4028 delka	24.93589	24.93355	11.3	22.6
134		4028 smer	250.084000	250.086450	33.3	66.4
135	4027	4026 delka	15.26083	15.28772	10.5	20.8
136		4026 smer	105.727200	105.732784	33.2	66.0
137		4023 delka	42.31678	42.31928	9.9	19.7
138		4023 smer	342.674900	342.669316	33.2	66.0
139	4023	4027 delka	42.30505	42.31928	9.9	19.7
140		4027 smer	43.111000	43.115852	33.2	66.2
141		4022 delka	46.62813	46.64653	10.4	20.7
142		4022 smer	192.186800	192.181948	33.2	66.2
143	4028	4026 delka	24.93700	24.93355	11.3	22.6
144		4026 smer	208.287000	208.285207	33.7	67.2
145		4029 delka	34.55282	34.55597	11.4	22.7
146		4029 smer	17.975700	17.977493	33.7	67.2
147	4029	4028 delka	34.55849	34.55597	11.4	22.7
148		4028 smer	180.463600	180.462726	34.0	67.7
149		4030 delka	64.24385	64.23881	11.4	22.7
150		4030 smer	375.000500	375.001374	34.0	67.7
151	4030	4029 delka	64.23676	64.23881	11.4	22.7
152		4029 smer	219.261200	219.262030	33.5	66.8
153		4013 delka	38.55667	38.55818	11.4	22.7
154		4013 smer	33.319000	33.318170	33.5	66.8
155	4013	4030 delka	38.55336	38.55818	11.4	22.7
156		4030 smer	183.424400	183.426244	32.7	65.0

157		10601 delka	11.15467	11.09794	1.7	3.4
158		10601 smer	169.869700	169.869879	35.0	69.7
159		10600 delka	6.10854	6.10376	1.9	3.7
160		10600 smer	13.719400	13.719902	35.3	70.3
161		4011 delka	70.94656	70.96022	8.0	16.0
162		4011 smer	372.859800	372.857275	33.1	65.8
163	4031	4020 delka	30.44834	30.44127	11.0	22.0
164		4020 smer	208.024000	208.025185	33.3	66.2
165		4032 delka	42.09038	42.08861	10.8	21.5
166		4032 smer	371.996400	371.995215	33.3	66.2
167	4032	4031 delka	42.08873	42.08861	10.8	21.5
168		4031 smer	286.177600	286.178677	33.7	67.1
169		4033 delka	49.22905	49.22870	11.0	21.8
170		4033 smer	90.592700	90.591623	33.7	67.1
171	4033	4032 delka	49.23052	49.22870	11.0	21.8
172		4032 smer	95.263700	95.264655	33.1	65.8
173		4022 delka	40.80677	40.80132	10.7	21.2
174		4022 smer	336.985900	336.984945	33.1	65.8
175	4034	4018 delka	20.40200	20.40968	5.0	10.0
176		4018 smer	277.335200	277.335200	35.4	70.4
177	4036	4018 delka	49.28800	49.27955	9.8	19.6
178		4018 smer	59.645400	59.640318	30.0	59.7
179		4034 delka	47.58600	47.58175	9.5	18.9
180		4034 smer	86.581200	86.580482	34.7	69.1
181		13090032 smer	258.668900	258.674699	30.4	60.4

Prehled neznamych - orientacni posuny

\*\*\*\*\*

i	stanovisko	priblizna hodn. [g]	korekce ==== [g]	vyrovnana === hodn. [g]	str.ch.	konf.i. ==== +-[cc] ===
5	4001	81.248301	-0.000868	81.247433	29.6	59.0
14	4005	3.091735	0.000362	3.092097	37.9	75.4
17	4008	62.376035	0.000362	62.376397	62.8	124.9
20	4002	202.565701	0.000201	202.565902	42.0	83.6
21	4009	229.402863	-0.000128	229.402735	53.7	106.9
24	4010	377.475871	-0.000040	377.475831	61.6	122.6
25	4003	105.640180	0.000551	105.640731	36.0	71.6
28	4011	219.611144	-0.000031	219.611113	35.0	69.7
35	4014	361.598198	-0.000285	361.597913	42.6	84.8
36	4004	233.983326	-0.000083	233.983242	24.3	48.4
39	4015	288.480226	-0.001165	288.479061	24.8	49.4
42	4016	332.846583	0.000221	332.846804	26.8	53.4
45	4017	142.094736	-0.000111	142.094624	40.0	79.5
48	4007	54.210960	0.000017	54.210977	39.2	78.0
51	4012	188.083106	0.000020	188.083126	46.7	93.0
54	4020	164.120948	0.001578	164.122526	47.2	93.9
59	4021	316.003848	-0.000077	316.003770	49.5	98.6
60	4019	332.822175	0.000528	332.822702	40.0	79.6

63	4022	39.243934	-0.000084	39.243850	41.4	82.4
70	4024	171.872540	-0.001554	171.870985	39.2	77.9
73	4018	203.027032	-0.000378	203.026653	25.6	51.0
78	4025	120.276933	0.000559	120.277492	51.2	101.9
81	4026	213.132687	-0.003006	213.129681	51.4	102.3
86	4027	60.132117	0.000263	60.132380	58.4	116.3
87	4023	159.685883	-0.000039	159.685844	55.6	110.6
88	4028	54.930866	0.000058	54.930924	53.7	107.0
91	4029	92.445376	0.000315	92.445691	51.3	102.0
94	4030	48.185036	-0.000001	48.185035	48.2	96.0
95	4013	98.077301	-0.000340	98.076961	41.5	82.5
96	4031	302.633568	0.000741	302.634309	53.7	106.9
99	4032	188.450402	0.000445	188.450847	53.7	106.8
100	4033	383.778137	-0.000322	383.777814	51.5	102.5
101	4034	105.440832	-0.000392	105.440439	56.2	111.8
102	4036	4.481473	0.000360	4.481833	29.7	59.1

## Prehled neznamych - souradnice

\*\*\*\*\*

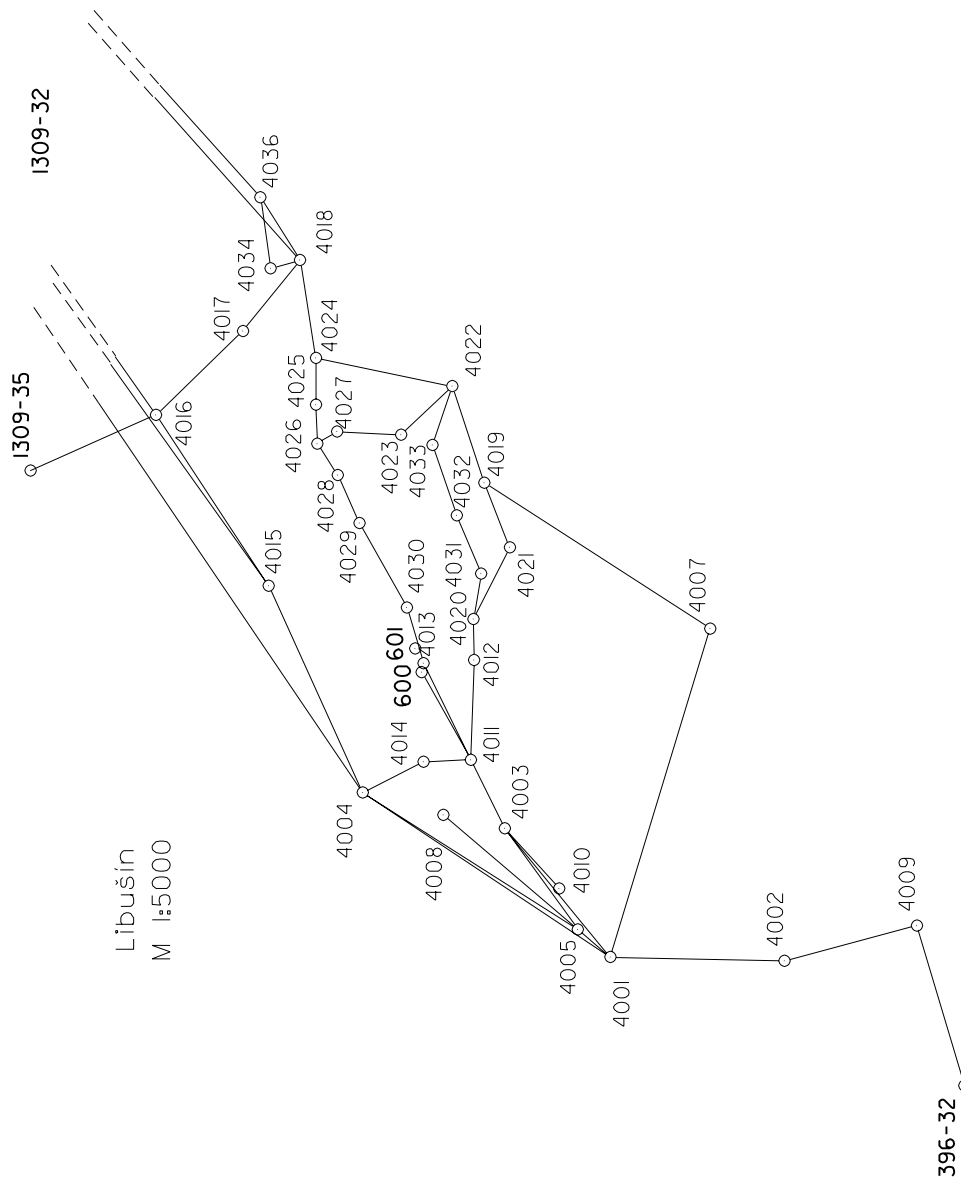
i	bod	priblizna	korekce	vyrovnana	str.ch.	konf.i.
=====	=====	hodnota	=====[m]	=====[m]	=====	+-[mm]====
	4001	typ: U		mp =	14.2	
1	y	766777.28000	-0.00037	766777.27963	9.9	19.7
2	x	1030029.22800	0.00009	1030029.22809	10.2	20.3
	4002	typ: U		mp =	16.3	
3	y	766779.59700	0.00041	766779.59741	11.2	22.4
4	x	1030144.53300	-0.00023	1030144.53277	11.8	23.6
	4003	typ: U		mp =	13.0	
6	y	766692.05500	-0.00037	766692.05463	10.3	20.5
7	x	1029959.31900	-0.00031	1029959.31869	7.9	15.6
	4004	typ: U		mp =	12.3	
8	y	766668.49800	-0.00021	766668.49779	8.5	16.8
9	x	1029865.26300	-0.00017	1029865.26283	8.9	17.7
	4005	typ: U		mp =	14.4	
10	y	766758.73400	0.00024	766758.73424	10.2	20.4
11	x	1030007.60300	-0.00039	1030007.60261	10.2	20.3
	4007	typ: U		mp =	17.6	
12	y	766559.70400	0.00021	766559.70421	14.6	29.1
13	x	1030095.33700	0.00003	1030095.33703	9.9	19.7
	4008	typ: U		mp =	19.6	
15	y	766683.10400	0.00013	766683.10413	13.9	27.7
16	x	1029918.82100	0.00051	1029918.82151	13.8	27.5

	4009	typ: U		mp =	16.8	
18	y	766756.24700	0.00038	766756.24738	12.1	24.1
19	x	1030231.93600	-0.00029	1030231.93571	11.6	23.2
	4010	typ: U		mp =	19.0	
22	y	766731.74100	0.00024	766731.74124	14.2	28.3
23	x	1029995.38200	0.00029	1029995.38229	12.5	25.0
	4011	typ: U		mp =	9.1	
26	y	766646.53300	-0.00032	766646.53268	7.5	15.0
27	x	1029936.98900	-0.00041	1029936.98859	5.2	10.3
	4012	typ: U		mp =	13.9	
29	y	766580.58200	-0.00037	766580.58163	12.9	25.7
30	x	1029939.29200	-0.00015	1029939.29185	5.1	10.1
	4013	typ: U		mp =	1.9	
31	y	766582.84100	-0.00027	766582.84073	1.9	3.8
32	x	1029905.70500	-0.00041	1029905.70459	0.3	0.7
	4014	typ: U		mp =	12.9	
33	y	766648.10100	0.00010	766648.10110	7.4	14.8
34	x	1029905.70300	-0.00035	1029905.70265	10.6	21.1
	4015	typ: U		mp =	16.4	
37	y	766531.52100	0.00013	766531.52113	13.1	26.1
38	x	1029803.16000	0.00021	1029803.16021	9.9	19.7
	4016	typ: U		mp =	17.9	
40	y	766418.43900	0.00005	766418.43905	14.7	29.3
41	x	1029728.76500	-0.00013	1029728.76487	10.2	20.3
	4017	typ: U		mp =	19.5	
43	y	766362.86600	0.00032	766362.86632	15.6	31.1
44	x	1029786.16500	-0.00006	1029786.16494	11.7	23.2
	4018	typ: U		mp =	19.7	
46	y	766316.20300	0.00020	766316.20320	15.7	31.3
47	x	1029824.12300	-0.00047	1029824.12253	11.9	23.8
	4019	typ: U		mp =	16.0	
49	y	766463.33100	0.00005	766463.33105	12.9	25.7
50	x	1029945.96100	-0.00021	1029945.96079	9.4	18.7
	4020	typ: U		mp =	15.3	
52	y	766553.60800	-0.00004	766553.60796	14.2	28.2
53	x	1029938.80000	-0.00024	1029938.79976	5.7	11.4
	4021	typ: U		mp =	15.9	
55	y	766506.05600	-0.00024	766506.05576	13.9	27.6

56	x	1029962.63700	-0.00049	1029962.63651	7.9	15.7
	4031	typ: U		mp =	17.7	
57	y	766523.59200	0.00041	766523.59241	16.4	32.6
58	x	1029943.87300	0.00003	1029943.87303	6.8	13.6
	4022	typ: U		mp =	17.4	
61	y	766399.56600	0.00031	766399.56631	13.1	26.0
62	x	1029924.69800	-0.00023	1029924.69777	11.4	22.7
	4023	typ: U		mp =	18.6	
64	y	766431.56900	-0.00040	766431.56860	13.9	27.7
65	x	1029890.76000	0.00030	1029890.76030	12.3	24.5
	4024	typ: U		mp =	17.8	
66	y	766380.94100	0.00030	766380.94130	14.2	28.2
67	x	1029834.35500	0.00029	1029834.35529	10.8	21.4
	4033	typ: U		mp =	18.8	
68	y	766438.21700	-0.00018	766438.21682	15.7	31.2
69	x	1029911.62500	0.00045	1029911.62545	10.4	20.6
	4025	typ: U		mp =	19.0	
71	y	766411.85800	0.00011	766411.85811	16.1	32.1
72	x	1029834.36200	-0.00032	1029834.36168	10.0	19.9
	4034	typ: U		mp =	20.5	
74	y	766321.65800	0.00011	766321.65811	15.8	31.5
75	x	1029804.45500	0.00032	1029804.45532	13.0	25.9
	4036	typ: U		mp =	22.2	
76	y	766274.54400	0.00051	766274.54451	17.8	35.4
77	x	1029797.79700	0.00011	1029797.79711	13.3	26.5
	4026	typ: U		mp =	17.1	
79	y	766437.51700	-0.00029	766437.51671	14.3	28.5
80	x	1029835.34000	-0.00028	1029835.33972	9.4	18.7
	4027	typ: U		mp =	19.4	
82	y	766429.70700	-0.00023	766429.70677	14.6	29.0
83	x	1029848.48200	-0.00001	1029848.48199	12.8	25.5
	4028	typ: U		mp =	17.4	
84	y	766458.40300	-0.00029	766458.40271	14.7	29.2
85	x	1029848.95800	-0.00002	1029848.95798	9.4	18.8
	4029	typ: U		mp =	15.9	
89	y	766489.87700	-0.00035	766489.87665	13.6	27.1
90	x	1029863.22400	-0.00045	1029863.22355	8.2	16.3
	4030	typ: U		mp =	11.8	

92	y	766545.89900	-0.00037	766545.89863	11.0	21.9
93	x	1029894.65900	-0.00041	1029894.65859	4.3	8.5
	4032	typ: U		mp =	19.1	
97	y	766484.80200	-0.00000	766484.80200	16.9	33.5
98	x	1029927.54000	0.00041	1029927.54041	8.9	17.8





Obrázek A.4: Vázaná síť – zhoršená konfigurace

# Příloha B

## Hlavičkové soubory knihovny

Median

---

## B.1 zmeny.h

```
/*
 * zmeny.h - sprava verzi
 * =====
 * Copyright (C) 1999 Jiri Vesely <vesely@gama.fsv.cvut.cz>
 * This file is part of the Median library.
 *
 */

#ifndef GaMaLib_Median_Zmeny_H
#define GaMaLib_Median_Zmeny_H

#ifndef NonStdCpp
namespace GaMaLib_Median
{
#endif

    extern const char* GaMaLib_Median_version;

#ifndef NonStdCpp
}
#endif
#endif // GaMaLib_Median
//EOF zmeny.h
```

## B.2 vyjimky.h

```

/*****
 * vyjimky.h - odvozeni vlatni vyjimky pro GaMaLib_Median      *
 * ===== *
 * Copyright (C) 1999 Jiri Vesely <vesely@gama.fsv.cvut.cz>    *
 * This file is part of the Median library.                    *
 *****/

#ifndef GaMaLib_Median_Vyjimky_H
#define GaMaLib_Median_Vyjimky_H

#include <gamalib/bodmer/vyjimky.h>

#ifndef NonStdCpp
namespace GaMaLib_Median
{
using GaMaLib_BodMer::Vyjimka;
using GaMaLib_BodMer::Double;
using GaMaLib_BodMer::Bod;
using GaMaLib_BodMer::SezBod;
using GaMaLib_BodMer::SezMer;
using GaMaLib_BodMer::SezCB;
using GaMaLib_BodMer::Mereni;
using GaMaLib_BodMer::Smer;
using GaMaLib_BodMer::Delka;
using GaMaLib_BodMer::Uhel;

#endif

class ps_chyba : public Vyjimka
{
public:
    ps_chyba(const std::string& popis) : Vyjimka("Chyba: "+popis) {}
};

#ifndef NonStdCpp
} // GaMaLib_Median
#endif

#endif // GaMaLib_Median_Vyjimky_H
// EOF vyjimky.h

```

## B.3 g\_fce.h

```

/*****
 * g_fce.h - pomocne funkce a tridy:
 * -----
 * - typedef std::vector<Bod> Pom_seznam;
 * - inline int signum(const Double& d1)
 * - inline Double sqr(const Double& d)
 * - enum Navesti_stavu
 * - enum Typy_Mereni
 * - inline Typy_Mereni TypMereni(Mereni* m)
 * - inline Double delka(const Bod& b1, const Bod& b2)
 * - inline bool even(std::vector<Double>::size_type& x)
 * - class Vyber_reseni
 * - class Statistika
 * - class Transformace
 * - class PodobnostniTr2D : public Transformace
 * =====
 * Copyright (C) 1999 Jiri Vesely <vesely@gama.fsv.cvut.cz>
 * This file is part of the Median library.
 *****/

#ifndef GaMaLib_Median_G_fce_H
#define GaMaLib_Median_G_fce_H

#include <algorithm>
#include <gamalib/bodmer/seznam.h>
#include <gamalib/median/vyjimky.h>

#ifndef NonStdCpp
namespace GaMaLib_Median
{
using namespace GaMaLib_BodMer;
}
#endif

// -----
typedef std::vector<Bod> Pom_seznam;

// -----
inline int signum(const Double& d1)
{
    if(d1 < 0)
        return -1;
    if(d1 > 0)
        return 1;
    return 0;
}; // int signum(const Double& d1)

// -----
inline Double sqr(const Double& d)
{

```

```

    return (d*d);
}; // Double sqr(Double& d)

// -----
enum Navesti_stavu
{
    neni_init = -2,
    neproveden_vypocet = -1,
    neni_reseni = 0,
    jednoznacne_reseni = 1,
    dvojnacne_reseni = 2,
    probehl_vypocet = 3
};

// -----
enum Typy_Mereni
{
    je_Delka = 1,
    je_Smer = 10,
    je_Uhel = 100
};

// -----
inline Typy_Mereni TypMereni(Mereni* m)
{
    Delka* d = m->PtrDelka();
    Smer* s = m->PtrSmer();
    if(d)
        return je_Delka;
    if(s)
        return je_Smer;
    return je_Uhel;
}; // Typy_Mereni TypMereni(Mereni*)

// -----
inline Double delka(const Bod& b1, const Bod& b2)
{
    return sqrt(sqr(b1.x()-b2.x())+sqr(b1.y()-b2.y()));
}; // Double delka(Bod& b1, Bod& b2)

// -----
inline bool even(std::vector<Double>::size_type& x)
{
    return (fmod(x,2) == 0);
}; // bool even(Double& x)

// -----
// pri nejednoznacnem reseni geodeticke ulohy se vybira jedno
// podle souhlasu s ostatnimi merenimi
class Vyber_reseni
{

```

```

private:
    Navesti_stavu stav;
    Bod B1, B2;
    SezBod* SB;
    SezMer* SM;
public:
    Vyber_reseni(SezBod* sb, SezMer* sm)
        : stav(neproveden_vypocet), SB(sb), SM(sm) {}
    Vyber_reseni(Bod& b1, Bod& b2, SezBod* sb, SezMer* sm) :
        stav(neproveden_vypocet), B1(b1), B2(b2), SB(sb), SM(sm) {}
    void Vypocet();
    void Vypocet(Bod b1, Bod b2)
    {
        stav = neproveden_vypocet;
        B1 = b1; B2 = b2;
        Vypocet();
    }
    Bod Reseni()
    {
        if(stav == neproveden_vypocet)
            throw ps_chyba("Vyber_reseni::nebyl proveden vypocet");
        if(stav == neni_reseni)
            throw ps_chyba("Vyber_reseni::nejednoznacne reseni");
        return B1;
    }
    int Stav() const
        { return (stav > neni_reseni ? jednoznacne_reseni : stav); }
}; // class Vyber_reseni

// -----
// Statistika - vypocet medianu ze seznamu souradnic
class Statistika
{
private:
    Pom_seznam* PS;
    Bod median;
    Navesti_stavu stav;
public:
    Statistika() : stav(neni_init) {}
    Statistika(Pom_seznam* ps) : PS(ps), stav(neproveden_vypocet) {}
    void Vypocet();
    void Vypocet(Pom_seznam* ps)
    {
        stav = neproveden_vypocet;
        PS = ps;
        Vypocet();
    }
    Navesti_stavu Stav() const { return stav; }
// vysledna souradnice
    Bod Median()
    {

```

```

        if(stav < neni_reseni)
            throw ps_chyba("Statistika: nebyl proveden vypocet");
        return median;
    }
}; // class Statistika

// -----
// bazova trida pro transformace
class Transformace
{
protected:
    // seznam bodu v cilove soustave
    SezBod& SB;
    // seznam bodu v mistni soustave
    SezBod& mistni;
    // seznam urcovanych bodu
    SezCB& urcovane;
    Navesti_stavu stav;
    // body transformovane z mistni do cilove soustavy
    SezBod transf_body;
    // v kazdem potomkovi se musi definovat - bude se tam minimalne
    // cistit transformacni klic
    virtual void Reset() = 0;
public:
    Transformace(SezBod& sb, SezBod& mist, SezCB& urc)
        : SB(sb), mistni(mist), urcovane(urc), stav(neproveden_vypocet) {}
    virtual ~Transformace()
    {
        transf_body.erase(transf_body.begin(), transf_body.end());
    }
    void Reset(SezBod& sb, SezBod& mist, SezCB& urc)
    {
        SB = sb;
        mistni = mist;
        urcovane = urc;
        stav = neproveden_vypocet;
        transf_body.erase(transf_body.begin(), transf_body.end());
        Reset();
    }
    virtual void Vypocet() = 0;
    Navesti_stavu Stav() const { return stav; }
    SezBod Transf_body() const
    {
        if(stav == neproveden_vypocet)
            throw ps_chyba("Transformace: nebyl proveden vypocet");
        if(stav == neni_reseni)
            throw ps_chyba("Transformace: nedostatecny pocet identickych bodu");
        return transf_body;
    }
}; // class Transformace

```



```

// -----
// Transformace v rovine
// Transformace z nutneho poctu bodu - vybira nejvhodnejssi dvojici
// - podobnostni transformace
/*
 * transformacni klic pro podobnostni transformaci:
 *
 * | x |   | tx | | a1  -a2 | | x' |
 * |   | = |   | | + |   | | * | |
 * | y |   | ty | | a2   a1 | | y' |
 *
 * transf_klic[0] = a2
 * transf_klic[1] = a1
 * transf_klic[2] = ty
 * transf_klic[3] = tx
 *
 */
class PodobnostniTr2D : public Transformace
{
private:
    std::vector<Double> transf_klic;
    void Reset();
    bool Dany_bod(const CisBod& cb)
    {
        return (std::find(urcovane.begin(), urcovane.end(), cb) ==
                urcovane.end());
    }
    void Identicke_body(SezBod::iterator& b1, SezBod::iterator& b2);
    void Transformacni_klic(SezBod::iterator& b1, SezBod::iterator& b2);
public:
    PodobnostniTr2D(SezBod& sb, SezBod& mist, SezCB& urc)
    : Transformace(sb, mist, urc) { Reset(); }
    void Vypocet();
    std::vector<Double> Transf_klic() const
    {
        if(stav == neproveden_vypocet)
            throw ps_chyba("PodobnostniTr2D::nebyl proveden vypocet");
        if(stav == neni_reseni)
            throw ps_chyba("PodobnostniTr2D::nedostatecny pocet identickych bodu");
        return transf_klic;
    }
}; // class PodobnostniTr2D

#ifdef NonStdCpp
} // namespace GaMaLib_Median
#endif

#endif // GaMaLib_Median_G_fce_H
// EOF g_fce.h

```

## B.4 g\_ulohy.h

```

/*****
 * g_ulohy.h - vypocty geodetickych uloh                               *
 * ----- *
 * - class GeodetickaUloha                                           *
 * - class Delka_delka : public GeodetickaUloha                       *
 * - class Smer_smer : public GeodetickaUloha                       *
 * - class Smer_delka : public GeodetickaUloha                       *
 * - class Smer_uhel : public GeodetickaUloha                       *
 * - class Delka_uhel : public GeodetickaUloha                       *
 * - class Uhel_uhel : public GeodetickaUloha                       *
 * - class Kruznice : public GeodetickaUloha                       *
 * ===== *
 * Copyright (C) 1999 Jiri Vesely <vesely@gama.fsv.cvut.cz> *
 * This file is part of the Median library. *
 *****/

#ifndef GaMaLib_Median_G_ulohy_H
#define GaMaLib_Median_G_ulohy_H

#include <gamalib/bodmer/seznam.h>
#include <gamalib/median/vyjimky.h>

#ifndef NonStdCpp
namespace GaMaLib_Median
{
// using namespace GaMaLib_BodMer; //VC++
#endif

class GeodetickaUloha
{
protected:
    int pocet_reseni;
    Bod* bod1;
    Bod* bod2;
    SezBod* SB;
    virtual void Prevod_mereni(Mereni*, Mereni*) = 0; // {}
public:
    GeodetickaUloha(SezBod* sb) : pocet_reseni(-1), SB(sb)
        { bod1 = new Bod; bod2 = new Bod; }
    virtual ~GeodetickaUloha() { delete bod1; delete bod2; }
    virtual void Vypocet() = 0; // {}
    int Pocet_reseni() const { return pocet_reseni; }
    Bod Reseni_1() const
    {
        if(pocet_reseni == -1)
            throw ps_chyba("GeodetickaUloha:nebyl proveden vypocet");
        if(pocet_reseni < 1)
            throw ps_chyba("GeodetickaUloha:uloha nema reseni");
        return *bod1;
    }
}

```

```

    }
    Bod Reseni_2() const
    {
        if(pocet_reseni == -1)
            throw ps_chyba("GeodetickaUloha:nebyl proveden vypocet");
        if(pocet_reseni < 2)
            throw ps_chyba("GeodetickaUloha:uloha nema dve reseni");
        return *bod2;
    }
};

//-----
class Delka_delka : public GeodetickaUloha
{
private:
    Delka* h1;
    Delka* h2;
    CisBod CB;          // aby se to lepe hledalo v delkach (urcovany)
    Double r1, r2;
    Bod B1;
    Bod B2;
    void Prevod_mereni(Mereni*, Mereni*);
public:
    Delka_delka() : GeodetickaUloha(NULL), h1(NULL), h2(NULL), r1(-1) {}
    Delka_delka(Mereni* m1, Mereni* m2, SezBod* sb, CisBod cb) :
        GeodetickaUloha(sb), CB(cb), r1(-1)
    {
        Prevod_mereni(m1, m2);
    }
    Delka_delka(Double& m1, Double& m2, Bod b1, Bod b2, SezBod* sb) :
        GeodetickaUloha(sb), r1(m1), r2(m2), B1(b1), B2(b2) {}
    ~Delka_delka() {}
    void Vypocet();
    void Novy_vypocet(Mereni* m1, Mereni* m2, SezBod* sb, CisBod cb)
    {
        bod1 = bod2 = NULL;
        SB = sb;
        CB = cb;
        r1 = -1;
        Prevod_mereni(m1, m2);
        Vypocet();
    }
};

//-----
class Smer_smer : public GeodetickaUloha
{
private:
    Smer* h1;
    Smer* h2;
    void Prevod_mereni(Mereni*, Mereni*);
};

```

```

public:
    Smer_smer() : GeodetickaUloha(NULL), h1(NULL), h2(NULL) {}
    Smer_smer(Mereni* m1, Mereni* m2, SezBod* sb) :
    GeodetickaUloha(sb)
    {
        Prevod_mereni(m1, m2);
    }
    ~Smer_smer() {}
    void Vypocet();
    void Novy_vypocet(Mereni* m1, Mereni* m2, SezBod* sb)
    {
        bod1 = bod2 = NULL;
        SB = sb;
        Prevod_mereni(m1, m2);
        Vypocet();
    }
};

//-----
class Smer_delka : public GeodetickaUloha
{
private:
    Smer* h1;
    Delka* h2;
    Bod B;
    Double r;
    void Prevod_mereni(Mereni*, Mereni*);
public:
    Smer_delka() : GeodetickaUloha(NULL), h1(NULL), h2(NULL), r(-1) {}
    Smer_delka(Mereni* m1, Mereni* m2, SezBod* sb) : GeodetickaUloha(sb), r(-1)
    {
        Prevod_mereni(m1, m2);
    }
    Smer_delka(Smer* m1, Double m2, Bod b, SezBod* sb) :
        GeodetickaUloha(sb), h1(m1), B(b), r(m2) {}
    ~Smer_delka() {}
    void Vypocet();
    void Novy_vypocet(Mereni* m1, Mereni* m2, SezBod* sb)
    {
        bod1 = bod2 = NULL;
        SB = sb;
        r = -1;
        Prevod_mereni(m1, m2);
        Vypocet();
    }
};

//-----
class Smer_uhel : public GeodetickaUloha
{
private:

```

```

    Smer* h1;
    Uhel* h2;
    void Prevod_mereni(Mereni*, Mereni*);
public:
    Smer_uhel() : GeodetickaUloha(NULL), h1(NULL), h2(NULL) {}
    Smer_uhel(Mereni* m1, Mereni* m2, SezBod* sb) :
    GeodetickaUloha(sb)
    {
        Prevod_mereni(m1, m2);
    }
    ~Smer_uhel() {}
    void Vypocet();
    void Novy_vypocet(Mereni* m1, Mereni* m2, SezBod* sb)
    {
        bod1 = bod2 = NULL;
        SB = sb;
        Prevod_mereni(m1, m2);
        Vypocet();
    }
};

//-----
class Delka_uhel : public GeodetickaUloha
{
private:
    Delka* h1;
    Uhel* h2;
    void Prevod_mereni(Mereni*, Mereni*);
public:
    Delka_uhel() : GeodetickaUloha(NULL), h1(NULL), h2(NULL) {}
    Delka_uhel(Mereni* m1, Mereni* m2, SezBod* sb) :
    GeodetickaUloha(sb)
    {
        Prevod_mereni(m1, m2);
    }
    ~Delka_uhel() {}
    void Vypocet();
    void Novy_vypocet(Mereni* m1, Mereni* m2, SezBod* sb)
    {
        bod1 = bod2 = NULL;
        SB = sb;
        Prevod_mereni(m1, m2);
        Vypocet();
    }
};

//-----
class Uhel_uhel : public GeodetickaUloha
{
private:
    Uhel* h1;

```

```

    Uhel* h2;
    void Prevod_mereni(Mereni*, Mereni*);
public:
    Uhel_uhel() : GeodetickaUloha(NULL), h1(NULL), h2(NULL) {}
    Uhel_uhel(Mereni* m1, Mereni* m2, SezBod* sb) :
    GeodetickaUloha(sb)
    {
        Prevod_mereni(m1, m2);
    }
    ~Uhel_uhel() {}
    void Vypocet();
    void Novy_vypocet(Mereni* m1, Mereni* m2, SezBod* sb)
    {
        bod1 = bod2 = NULL;
        SB = sb;
        Prevod_mereni(m1, m2);
        Vypocet();
    }
};

//-----
// ** vypocet parametru kruznice z dvou bodu a obvodoveho uhlu **
class Kruznice : public GeodetickaUloha
{
private:
    Uhel* h1;
    Bod B1, B2;
    Double R;
    void Prevod_mereni(Mereni*, Mereni*) {}
public:
    Kruznice() : GeodetickaUloha(NULL), h1(NULL) {}
    Kruznice(Uhel* u,SezBod* sb) : GeodetickaUloha(sb), h1(u) {}
    ~Kruznice() {}
    void Vypocet();
    void Novy_vypocet(Uhel* u,SezBod* sb)
    {
        bod1 = bod2 = NULL;
        SB = sb;
        h1 = u;
        Vypocet();
    }
    Double r() const
    {
        if(pocet_reseni == -1)
            throw ps_chyba("Kruznice:nebyl proveden vypocet");
        if(pocet_reseni < 1)
            throw ps_chyba("Kruznice:uloha nema reseni");
        return R;
    }
};

```

```
#ifndef NonStdCpp
} // namespace GaMaLib_Median
#endif

#endif // GaMaLib_Median_G_ulohy_H
// EOF g_ulohy.h
```

## B.5 pribl\_b.h

```

/*****
 * pribl_b.h - vypocet pribl. souradnic konkretniho bodu      *
 * ===== *
 * Copyright (C) 1999 Jiri Vesely <vesely@gama.fsv.cvut.cz> *
 * This file is part of the Median library.                  *
 *****/

#ifndef GaMaLib_Median_Pribl_b_H
#define GaMaLib_Median_Pribl_b_H

#include <gamalib/bodmer/seznam.h>
#include <gamalib/bodmer/bmfce.h>
#include <gamalib/median/vyjimky.h>
#include <gamalib/median/g_fce.h>

#ifndef NonStdCpp
namespace GaMaLib_Median
{
using namespace GaMaLib_BodMer;
using GaMaLib_BodMer::smernik;
#endif

class PriblBod
{
private:
    Pom_seznam Vyp_body;
    SezBod SB;
    CisBod CB;
    SezMer SM;
// Navesti_stavu -> viz g_fce.h
    SezMer SM_U; // pro vnitri uhly
    SezMer SM_S; // pro vnejsi smery
    Bod v_bod; // konecny bod - vystup
    Bod v_bod2; // --"-- - pro dvojznacne reseni
    SezBod* SB_puv; // opakovani vypoctu s jinym CB
    SezMer* SM_puv;
    Navesti_stavu stav;
    void CistiSeznamy(); // vyprazdneni pomocnych seznamu
    Uhel* VyrobUhel(const SezMer::iterator i, const SezMer::iterator j)
    {
        Smer* s1 = (*i)->PtrSmer();
        Smer* s2 = (*j)->PtrSmer();
        if(!(s1 && s2))
            throw ps_chyba("PriblBod::VyrobUhel : neni smer");
        Double uhel = s2->mer() - s1->mer();
        return new Uhel(s1->stan(), s1->cil(), s2->cil(),
            (uhel < 0 ? uhel+2*M_PI : uhel), s1->str());
    }
    Smer* VyrobSmernik(const Uhel* u, const CisBod& cb)

```



```

{
    CisBod bod = (u->cil() == cb ? u->cil2() : u->cil());
    Double sm = smernik(SB[u->stan()],SB[bod]);
    sm += (u->cil() == cb ? -u->mer() : u->mer());
    sm += (sm < 0 ? 2*M_PI : 0);
    sm -= (sm >= 2*M_PI ? 2*M_PI : 0);
    return new Smer(u->stan(),cb,sm,u->str());
}
Smer* VyrobSmernik(const Smer* s, const CisBod& cb)
{
    Double sm = s->mer() + SB[s->stan()].z().orpos;
    sm -= (sm >= 2*M_PI ? 2*M_PI : 0);
    return new Smer(s->stan(),cb,sm,s->str());
}
bool ZnamyCil(SezMer::const_iterator i)
{
    // true: znam sour. cile
    Uhel* u = (*i)->PtrUhel();
    return ZnamyCil1(i) && (u ? ZnamyCil2(u) : true);
}
bool ZnamyCil1(SezMer::const_iterator i)
{
    return SB[(*i)->cil()].test_yx();
}
bool ZnamyCil2(const Uhel* u)
{
    // true: znam sour. cile2 u uhlu
    return SB[u->cil2()].test_yx();
}
bool ZnameStan(SezMer::const_iterator i)
{
    // true: znam sour. stanoviska a je pro nej znam or. posun
    Smer* s = (*i)->PtrSmer();
    return SB[(*i)->stan()].test_yx() &&
        (s ? SB[s->stan()].test_z() : true);
}
void SrovnejMereni(SezMer&);
void Reset(SezBod*, SezMer*, const CisBod&);
public:
    PriblBod(SezBod* sb, SezMer* sm, const CisBod& cb) :
        SB_puv(sb), SM_puv(sm), stav(neni_init)
    {
        Reset(sb,sm,cb);
    }
    PriblBod(SezBod* sb, SezMer* sm)
        : SB_puv(sb), SM_puv(sm), stav(neni_init) {};
    ~PriblBod()
    {
        CistiSeznamy();
    }
    void Vypocet(SezBod* sb, SezMer* sm, const CisBod& cb)

```

```
{
    SB_puv = sb;
    SM_puv = sm;
    Reset(sb, sm, cb);
    Vypocet();
}
void Vypocet(const CisBod& cb)
{
    Reset(SB_puv, SM_puv, cb);
    Vypocet();
}
void Vypocet();
Navesti_stavu Stav() const { return stav; }
Bod Reseni()
{
    if(stav == neproveden_vypocet)
        throw ps_chyba("PriblBod:nebyl proveden vypocet");
    if(stav == neni_reseni)
        throw ps_chyba("PriblBod:neni reseni");
    return v_bod;
}
Bod Reseni_2()
{
    if(stav == neproveden_vypocet)
        throw ps_chyba("PriblBod:nebyl proveden vypocet");
    if(stav == neni_reseni)
        throw ps_chyba("PriblBod:neni reseni");
    if(stav == jednoznacne_reseni)
        throw ps_chyba("PriblBod:pouze jednoznacne reseni");
    return v_bod2;
}
};

#ifdef NonStdCpp
} // namespace GaMaLib_Median
#endif

#endif // GaMaLib_Median_Pribl_b_H
// EOF pribl_b.h
```

## B.6 pribl\_s.h

```

/*****
 * pribl_s.h - vypocet pribliznych souradnic          *
 * - jednoho bodu                                  *
 * - seznamu bodu                                   *
 * - vseh neznamych bodu                           *
 * ===== *
 * Copyright (C) 1999 Jiri Vesely <vesely@gama.fsv.cvut.cz> *
 * This file is part of the Median library.          *
 *****/

#ifndef GaMaLib_Median_Pribl_s_H
#define GaMaLib_Median_Pribl_s_H

#include <algorithm>
#include <gamalib/bodmer/seznam.h>
#include <gamalib/median/g_fce.h>

#ifndef NonStdCpp
namespace GaMaLib_Median
{
using namespace GaMaLib_BodMer;
}
#endif

class PriblSour
{
private:
// seznam bodu, v nem se predavaji vysledky
  SezBod& SB;
  SezMer& SM;
// Navesti_stavu -> viz g_fce.h
  Navesti_stavu stav;
// seznam pozadovanych bodu (pocitanych)
  SezCB urcovane;
// seznam vypoctenych bodu
  SezBod vypoctene;
  int vnoreni;
  bool Neobsahuje(CisBod cb)
  {
    return (std::find(urcovane.begin(),urcovane.end(),cb) == urcovane.end());
  }
  bool Mistni_mereni(SezMer::iterator sm, SezCB sb)
  {
    bool pom = false;
    pom = (std::find(sb.begin(), sb.end(), (*sm)->stan()) != sb.end()) &&
          (std::find(sb.begin(), sb.end(), (*sm)->cil()) != sb.end());
    if((*sm)->PtrUhel())
    {
      Uhel* u = static_cast<Uhel*>(*sm);
      pom = pom && (std::find(sb.begin(), sb.end(), u->cil2()) != sb.end());
    }
  }
};

```

```

    }
    return pom;
}
void Reset()
{
    stav = neproveden_vypocet;
    urcovane.erase(urcovane.begin(), urcovane.end());
    vypoctene.erase(vypoctene.begin(), vypoctene.end());
}
bool Mereni_obsahujeCB(SezMer::iterator m, SezCB::iterator cb)
{
    bool pom = (((*m)->stan() == (*cb)) || ((*m)->cil() == (*cb)));
    if((*m)->PtrUhel())
    {
        Uhel* u = static_cast<Uhel*>(*m);
        pom = pom || (u->cil2() == (*cb));
    };
    return pom;
}
// true - existuji alespon dva body s danymi souradnicemi
bool Resitelna_data(SezBod& b);
// true - existuji minimalne 2 mereni s bodem cb
bool Nutna_mereni(CisBod cb);
// v SM a SB najde body bez souradnic a jejich cisla da do urcovane
void Najdi_nevypoctene();
// presun bodu cb z seznamu odkud do kam
void Presun(SezBod& odkud, SezBod& kam, CisBod& co);
// vypocet klasickym protinanim; true - vypocetlo alespon jeden bod
bool Pocitej_protinani(SezBod& body, SezCB& co);
// vypocet bodu, ktere nejdu beznym protinanim - napr. vetknuty PGP
// true - urceny souradnice alespon jednoho bodu
bool Pocitej_vetknuti();
// kombinace predeslych dvou metod
void Vypocetni_cyklus();
public:
    PriblSour(SezBod& b, SezMer& m) : SB(b), SM(m), vnoreni(0)
    {
        Reset();
    }
    PriblSour(SezBod& b, SezMer& m, int vn)
        : SB(b), SM(m), vnoreni(vn)
    {
        Reset();
    }
    void Reset(SezBod& b, SezMer& m)
    {
        SB = b;
        SM = m;
        vnoreni = 0;
        Reset();
    }
}

```

---

```

// jeden bod - i uz vypocteny; true - podarilo se urcit souradnice
bool Vypocet(CisBod cb);
// body v SezCB - i uz vypoctene; true - podarilo se urcit vsechny souradnice
bool Vypocet(SezCB cb);
// vsechny nezname body; true - podarilo se urcit vsechny souradnice
bool Vypocet()
{
// hledam vsechny body, ktere nejsou v seznamu bodu (jen u mereni)
// a body nemajici souradnice
stav = probehl_vypocet;
Najdi_nevypoctene();
if(!Resitelna_data(SB))
return false;
Vypocetni_cyklus();
return Vypocteno_vse();
}
bool Vypocteno_vse() const
{
if(stav <= neproveden_vypocet)
throw ps_chyba("PriblSour::Vypocteno_vse: neproveden vypocet");
return urcovane.empty();
}
SezCB Nevypoctene() const
{
if(stav <= neproveden_vypocet)
throw ps_chyba("PriblSour::Nevypoctene: neproveden vypocet");
return urcovane;
}
SezBod Vypoctene() const
{
if(stav <= neproveden_vypocet)
throw ps_chyba("PriblSour::Vypoctene: neproveden vypocet");
return vypoctene;
}
int Celkem_bodu () const
{
return SB.size();
}
int Celkem_mereni () const
{
return SM.size();
}
};

#ifdef NonStdCpp
} // namespace GaMaLib_Median
#endif

#endif // GaMaLib_Median_Pribl_s_H
// EOF pribl_s.h

```