

STAVEBNÁ FAKULTA
SLOVENSKEJ TECHNICKEJ UNIVERZITY V BRATISLAVE
Katedra geodézie

**VYUŽITIE PROGRAMOVACÍCH PROSTRIEDKOV
PRI AUTOMATIZÁCII GEODETICKÝCH
VÝPOČTOV**

(Študentská vedecká práca)

Autor:

Tomáš Bacigál, 4.GaK

Marián Rehuš, 3.GaK

Dušan Danko, 4.GaK

Vedúci práce:

Doc. Ing. Štefan Sokol, PhD

Ing. Andrej Villim

Študentská vedecká konferencia

Bratislava 10. apríla 2002

VYUŽITIE PROGRAMOVACÍCH PROSTRIEDKOV PRI AUTOMATIZÁCII GEODETICKÝCH VÝPOČTOV

Abstrakt

Využitie programovacieho jazyka C++ a Delphi pri riešení špecifických geodetických úloh. Súradnicové riešenie trasy komunikácie v tvare jednoduchého kružnicového oblúka, výpočet rajónu, pretínania napred, pretínanie nazad, riešenie Hansenovej úlohy a redukcia niektorých meraných veličín.

Summary

Making use of programming language C++ and Delphi to solve some specific geodetic problems. Computation of parameters and points coordinates of communication track given as simple circular arc. Computation of polar bar, forward intersection, resection, Hansen's problem, reduction of some survey values.

Úvod

Geodézia ako fyzikálno-matematická veda prešla od svojich počiatkov bohatou históriou. Vývoj presnejších meracích prístrojov spolu s teoretickými objavmi fyzikálnych, geometrických a iných zákonitostí zvýšili efektivitu geodetických prác a výrazne prispeli k zvýšeniu presnosti meraných veličín. Elektronika umožnila vznik nových metód zberu údajov, napr. využitím umelých družíc Zeme na presné určenie polohy, a takisto priniesla fenomén počítačov. Tie znamenajú skutočný prevrat, svojou rýchlosťou a programovateľnosťou sú vhodné pre riešenie takmer akejkoľvek geodetickej úlohy. Podmienkou úspešnosti však naďalej zostáva človek a jeho programátorská, prípadne používateľská zručnosť.

Na platformu dnešných osobných počítačov existuje pestrá paleta programovacích nástrojov, t.j. jazyk + prekladač. V našej práci prezentujeme programy, vytvorené v jazyku C, jeho objektovej modifikácii C++ a v jazyku Delphi. Tieto jazyky sú pomerne ľahko zvládnuteľné, základy jazyka C sú dokonca zahrnuté aj v učebných osnovách odboru Geodézia a kartografia na Stavebnej fakulte STU.

Implementácia teoretického modelu a aplikačné prostredie

Vytvorenie akéhokoľvek programu je proces skladajúci sa z niekoľkých krokov, nedodržanie ktorých vedie k neprehľadnosti, chybovosti a často aj nefunkčnosti diela. Výkupnou cenou tak býva neúmerne množstvo času, strávené testovaním funkčnosti softvérového produktu. Postupnosť týchto krokov možno zhrnúť do osnovy:

1. Definícia účelu. Rozhodujem o cieľovej skupine užívateľov, skúmam ich požiadavky a predstavy.
2. Navrhnutie matematického modelu. Skúmam možnosti implementácie teoretických podkladov do logiky počítača a navrhujem najvhodnejší algoritmus.
3. Navrhnutie grafického používateľského rozhrania. Snažím sa o prehľadný a intuitívny systém komunikácie počítača s užívateľom v podmienkach prekladačom a operačným systémom.
4. Zápis v zdrojovom kóde a preklad do strojového kódu. Premieňam ľudské predstavy na strohé vety v jazyku počítača. Tento krok má vplyv na stabilitu a efektívnosť programu a pri väčších projektoch vyžaduje pokročilé programátorské znalosti.
5. Testovanie a distribúcia. Zisťujem správnosť vstupno-výstupných a výpočtových funkcií programu na konkrétnych úlohách. Fáza testovania je nevyhnutná pre zaručenie kvality a často začína už pri návrhu algoritmu.

Načrtli sme postup, ku ktorému sme dospeli po niekoľkých amatérskych pokusoch. Tie poskytli cennú teoretickú, ale hlavne empirickú základňu, z ktorej sme vytiahli do boja proti komplexnejšiemu problému. Vytvoriť ucelený a široko použiteľný program.

Súradnicové riešenie kružnicových oblúkov

Nápad sa zrodil počas štúdia predmetu Inžinierska geodézia a ako inak, s cieľom uľahčiť zdĺhavé a pritom elementárne výpočty. Rozhodli sme sa pre implementáciu pod operačným systémom MS DOS, ktorá hoci neponúka ľahkosť grafického prostredia, jednako bola bližšie práve absolvovanej výučbe programovania v jazyku C v strohom prostredí UNIX. Mohli sme sa tak viac sústrediť na samotný výpočtový potenciál vznikajúceho softvéru. Pozrime sa naň bližšie.

Obr. 1 Úvodná obrazovka

```
Programmed by Tomas Bacigal

uvod -> [u]
Obluk dany prvkami:   t1, t2, r -> [1]
                      t1, t2, TK [2]
                      t1, t2, t3 [3]
                      t1, t2, A  [4]
                      TK, K1, r  [5]
                      A, B, C    [6]
                      TK, K1, A   [7]
                      TK, 2, A    [8]
                      TK, A, t2   [9]
                      t1, K1, r   [0]
Drobne vypocty -> [d]
Schema kruznicoveho obluka -> [s]
koniec -> [k]
```

Program je vytvorený v editovacom a kompilačnom systéme Borland C, obsahuje 2 súbory, jeden spúšťací a druhý nevyhnutný pre intepretáciu grafických elementov. Po spustení sa objaví úvodná obrazovka (Obr.1) s ponukou možností. Výber možno uskutočniť zadáním zodpovedajúceho písmena a potvrdením klávesou <Enter>. Začínajúcemu používateľovi odporúčame pozrieť úvod a najmä schému kružnicového oblúka, čo mu pomôže lepšie pochopiť systém komunikácie s programom a vyhnúť sa nedorozumeniu či azda sklamaniu. Menu ďalej ponúka samotný výpočet kružnicových oblúkov podľa zadanej kombinácie určujúcich prvkov. Po výpočte hlavných prvkov a súradníc charakteristických bodov program sprístupní výpočet súradníc podrobných bodov. Posledná voľba úvodného menu nás zavedie k drobným geodetickým výpočtom ako určenie smerníka a dĺžky spojnice dvoch bodov, výpočet rajónu, smerové a uhlové pretínanie napred.

Fungovanie programu najlepšie ukážeme na príklade. Predpokladajme kružnicový oblúk zadany dvoma dotyčnicami a polomerom. Potvrdením voľby "1" program postupne požiada o všetky vstupné údaje, odoslaním posledného rozbehne výpočet a zobrazí výsledky (Obr. 2).

```
1
zadaj suradnice Y,X bodov 1,2 - t1:
576524.11 1120575.99 576126.41 1120872.77
zadaj suradnice Y,X bodov 3,4 - t2:
576126.41 1120872.77 576675.68 1121037.23
zadaj polomer[m]:
165

VYSLEDKY:
      uhol alfa: 140.6664g      VB [ 576126.409, 1120872.769 ]
      uhol tau : 59.3336g      TK [ 576389.334, 1120676.564 ]
      polomer r: 165.000m      KT [ 576440.688, 1120966.869 ]
      dotycnica t: 328.064m    S [ 576488.016, 1120808.803 ]
                                K [ 576325.538, 1120837.544 ]
```

Obr.2 Príklad riešenia kružnicového oblúka

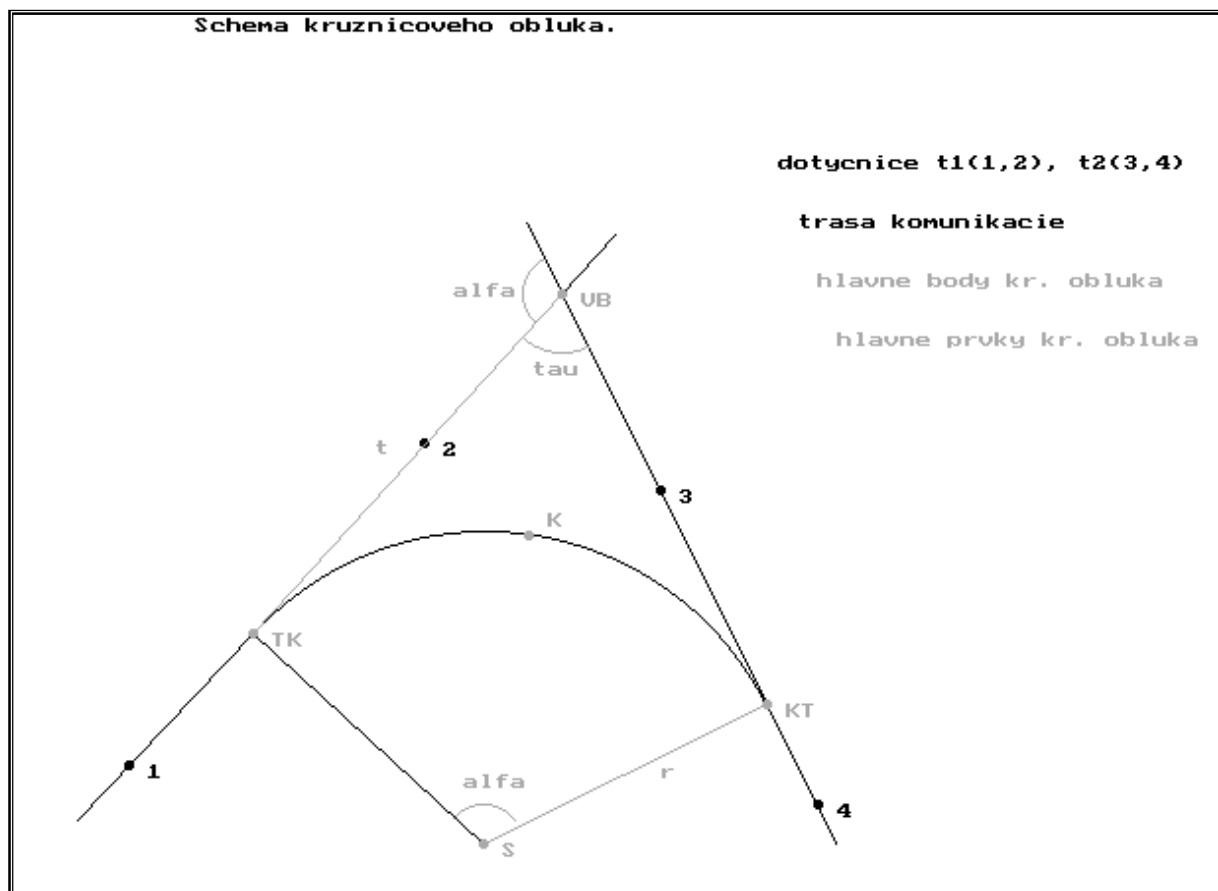
```
p
zadaj suradnice Y,X zaciatočneho a končoveho bodu trasy (A,B):
576524.11 1120575.99 576675.68 1121037.23
zadaj stanícenie podrobných bodov st[m]:
50.00

Suradnice podrobných bodov:
      1 [ 576484.038, 1120605.893 ]
      2 [ 576443.965, 1120635.796 ]
      3 [ 576403.893, 1120665.700 ]
      4 [ 576365.810, 1120697.939 ]
      5 [ 576338.295, 1120739.458 ]
      6 [ 576324.423, 1120787.297 ]
      7 [ 576325.459, 1120837.095 ]
      8 [ 576341.309, 1120884.315 ]
      9 [ 576370.527, 1120924.653 ]
     10 [ 576410.452, 1120954.435 ]
     11 [ 576457.216, 1120971.818 ]
     12 [ 576505.115, 1120986.160 ]
     13 [ 576553.014, 1121000.502 ]
     14 [ 576600.913, 1121014.843 ]
     15 [ 576648.812, 1121029.185 ]
```

Obr.3 Výpočet podrobných bodov trasy komunikácie

V prípade záujmu o určenie súradníc podrobných bodov trasy komunikácie je potrebné prejsť klávesou “Medzerník” do hlavného menu a zadať parameter “p”. Program nás opäť svojou interaktivitou vedie (Obr. 3). Podotýkame, že program otestuje polohu začiatočného a koncového bodu trasy vzhľadom k oblúku a dotyčniciam.

Ako sme už spomínali, pred zadávaním vstupných parametrov je dobré preštudovať si schému kružnicového oblúka (Obr. 4), dostupnú z hlavného menu pod písmenom “s”. Je to jediná grafická obrazovka programu a vyžaduje prítomnosť súboru “egavga.bgi”. Jej vykresľovanie je rozvrhnuté do krokov a zprehľadňuje tak výslednú kresbu spolu s farebným odlíšením jednotlivých elementov.



Obr. 4 Schéma kružnicového oblúka

Ukázali sme si “interface” programu, základy ovládania aj príklad výpočtu. Tým však zďaleka nie sú vyčerpané všetky možnosti a prekvapenia programu, skutočná kvalita je skrytá v algoritmoch a prejaví sa až pri riešení konkrétnych úloh. Je výsledkom mnohých hodín strávených testovaním a ošetrovaním možných situácií, týkajúcich sa nejednoznačnosti riešenia, kritických hodnôt vstupných parametrov a podobne.

Na záver ešte ukážka zdrojového kódu tvoreného v špecializovanom editore fy Borland, ktorý umožňuje ľahkú kontrolu zvýrazňovaním syntaxe a ďalšími funkciami (Obr. 5). Editor je prepojený na prekladač (tzv. kompilátor) a ďalšie nástroje ako odstraňovač chýb (debuger) a pod.

Programovanie v jazyku C pod operačným systémom MS DOS nám do určitej miery pomohlo zvládnuť programovací jazyk až k rutinej ľahkosti, dalo nám priestor venovať čas zdokonaľovaniu návrhu matematického modelu na úkor strohej grafiky a pohodlnosti používateľského rozhrania. No hoci v jednoduchosti spočíva krása, nároky na grafickú stránku

softvéru všeobecne vzrástli a stali sa dôležitým parametrom kvality. Pôvodné programátorské prostredie narazilo na hranicu svojich možností a preto sme nový projekt automatizácie geodetických výpočtov začali realizovať v prostredí objektovo-orientovaného a používateľsky príjemného operačného systému Windows 9x.



```
pripad1(void)
{
    double y1,x1,y2,x2,y3,x3,y4,x4,smernik1,smernik2,smernik2_dop;

    printf("zadaj suradnice Y,X bodov 1,2 - t1:\n");
    scanf("%lf %lf %lf %lf",&y1,&x1,&y2,&x2);
    printf("zadaj suradnice Y,X bodov 3,4 - t2:\n");
    scanf("%lf %lf %lf %lf",&y3,&x3,&y4,&x4);

    if ( identita_2b(y1,x1,y2,x2) || identita_2b(y3,x3,y4,x4) )
        return(0);

    printf("zadaj polomer[m]:\n");
    scanf("%lf",&r);
    smernik1 = smernik(y1,x1,y2,x2);
    smernik2 = smernik(y3,x3,y4,x4);
    alfa = uhol(smernik2 - smernik1);
    smernik2_dop = doplnok(smernik2);

    if (smernik1 == smernik2 || smernik1 == smernik2_dop)
    {
        printf("\nERROR: Dotyčnice su rovnobezne!\n");
        return(0);
    }

    pretinanie(y1, x1, smernik1, y4, x4, smernik2_dop, &yVB, &xVB);
    t = fabs( r*tan( alfa/(2*R0) ) );
    rajon(yVB, xVB, doplnok(smernik1), t, &yTK, &xTK);
    rajon(yVB, xVB, smernik2, t, &yKT, &xKT);

    if (alfa < 200)
    {
        Q = 1;
    }
    else
    {
        Q = -1;
        alfa = 400 - alfa;
    }

    smernikSTK = smer(smernik1 - Q*100);
    rajon(yTK, xTK, doplnok(smernikSTK), r, &yS, &xS);
    rajon(yS, xS, smer(smernikSTK + Q*alfa/2), r, &yK, &xK);
    tau = 200 - alfa;

    return(1);
}
```

Obr. 5 Ukážka zdrojového kódu v editore fy Borland

Úlohy rovinnej geodézie

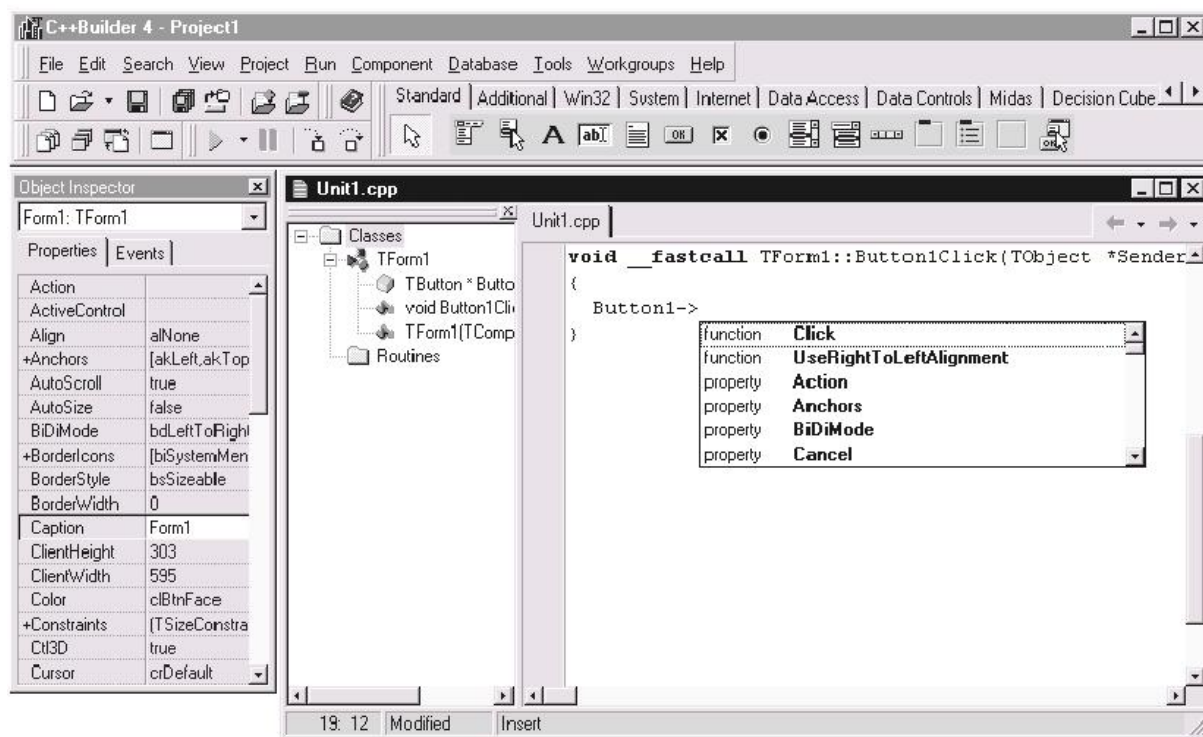
Nápad opäť prišiel s potrebou v geodetickej praxi študentov, tentokrát eliminovať manuálne výpočty najzákladnejších príkladov v rámci riešenia zložitejších úloh. Ide najmä o výpočet rajónu, pretínania napred, nazad a riešenie Hansenovej úlohy. Návrh programu sme realizovali v jednom z najmodernejších vývojových systémov C++ Builder v5.0 fy Borland (Obr. 6), ktorý integruje všetky výhody objektovo-orientovaného grafického operačného systému Windows 9x.

Ako študenti geodézie s poznatkami nanajvýš v programovaní v jazyku C sme tak museli vyriešiť hneď niekoľko problémov:

- prechod na novú filozofiu objektového programovania,
- zvládnutie množstva nástrojov vizuálneho programovacieho prostredia,

- návrh novej, grafickej koncepcie komunikácie programu s užívateľom vrátane výstupov v inej ako číselnej/textovej forme.

Po ich aspoň čiastočnom prekonaní, a to najmä vďaka [1], sme však ocenili ľahkosť, s akou sme odrazu mohli pristupovať k návrhu programového prostredia popri nezmenených



prioritách bezchybnosti matematického modelu.

Obr.7 Vizúálne programovacie prostredie Borland C++ Builder

Na zostavenie a zápis funkčných vzťahov sme po prvýkrát s výhodou použil maticový prepis známych riešení a vzťahov podľa [2]. Matice a operácie s nimi totiž predstavujú v súčasnosti najefektívnejší spôsob zápisu dát a matematických operácií do logiky jazyka výpočtovej techniky.

Samozrejmosťou pri zostavovaní programu bola snaha o dodržanie všetkých vyššie uvedených zásad aj napriek obmedzujúcim časovým limitom, ktoré ho sprevádzali.

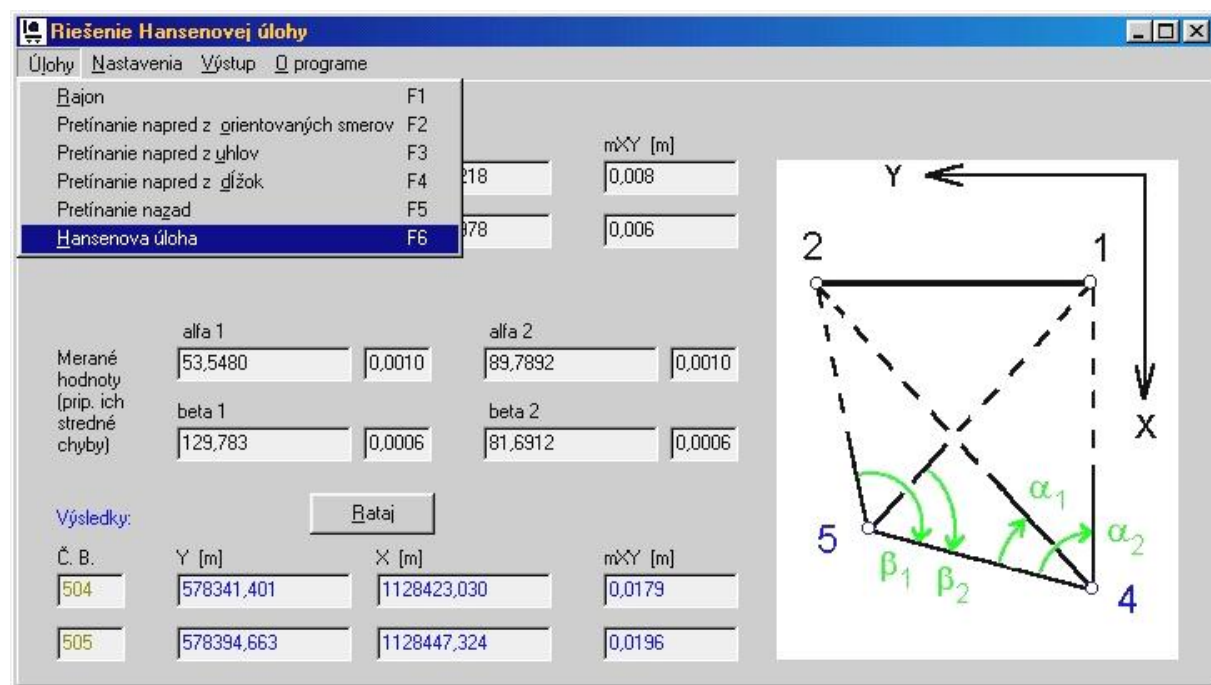
Aby sme príliš neteoretizovali, pozrime sa teraz bližšie na výsledok niekoľko týždenného snaženia. Program, ako obvyčajne, obsahuje spúšťací súbor, ďalej skupinu obrázkových a jeden textový súbor so všetkými potrebnými informáciami. V okamihu spustenia programu sa objaví klasické "windowsoidné" okno s ponukou v hornej časti. Tá obsahuje príkazy alebo zaškrtnuté položky s uvedením klávesovej skratky:

▼ Úlohy

- | | | |
|---|--|----|
| Ø | Rajón | F1 |
| Ø | Pretínanie napred z orientovaných smerov | F2 |
| Ø | Pretínanie napred z uhlov | F3 |
| Ø | Pretínanie napred z dĺžok | F4 |
| Ø | Pretínanie nazad | F5 |

- Ø Hansenova úloha F6
- ▼ Nastavenia
 - Ø Počítaj presnosť F9
 - Ø Zobraz pozadie
 - Ø Zobraz schému
- ▼ Výstup
 - Ø Nakresli situáciu F12
- ▼ O programe

Štandardne na začiatku nám program ponúkne vstupno-výstupnú masku pre úlohu Rajón bez počítania presnosti. To môžeme aktivovať napr. klávesou F9 a zvolením konkrétnej úlohy na začiatku alebo kedykoľvek počas práce. Po zadaní vstupných údajov (znak oddeľujúci desatinné miesta musí byť totožný s tým, ktorý je v nastaveniach operačného systému, štandardne čiarka, príp. bodka) stačí kliknúť na tlačítko “Rátaj” a v dolnej časti sa zobrazia výsledky. V prípade nejasnosti ohľadne zadávania vstupných hodnôt pre jednotlivé úlohy si




môžeme zapnúť pomocnú schému v ponuke Nastavenia (Obr. 8).

Obr. 8 Ukážka základného okna programu

Ak nás zaujíma grafické znázornenie situácie bodov z konkrétnej úlohy, program cez príkaz Výstup / Nakresli situáciu otvorí ďalšie okno a v ňom danú situáciu vykreslí, a to v premenlivej mierke, závislej na veľkosti okna, vzdialenosti bodov a rozlíšení pracovnej plochy. V prípade, že bola počítaná aj presnosť, tj. stredná súradnicová chyba m_{xy} , vykreslia sa aj stredné elipsy chýb, ale vo fixnej mierke 5:1. Obe mierkové čísla sú zobrazené v dolnej časti, okrem nich sa v okne nachádza aj editovateľné textové pole s predpísanými výsledkami

výpočtu, parametrami stredných elíps chýb a voľným miestom na vlastné poznámky. Obsah takéhoto okna odošleme na tlač stlačením kombinácie kláves <CTRL> + <T> alebo výberom z miestneho menu po kliknutí pravým tlačítkom myši na plochu okna.

Keďže je lepšie raz vidieť ako stokrát počuť, pozrime sa na “interface“ programu prostredníctvom menšieho príkladu. Majme za úlohu zistiť polohu bodu 504, na ktorý bola meraná vzdialenosť z dvoch susedných bodov 501 a 502 so známymi súradnicami Y, X v systéme S-JTSK aj strednými súř.chybami m_{XY} . Vstupné polia vyplníme podľa schémy a necháme vypočítať (Obr. 9). Výsledky sú farebne odlišené. Situáciu si dajme vykresliť stlačením F12, a v prípade potreby dopíšme do textového poľa vlastné poznámky. Obrázok je pripravený na tlač (Obr. 10).


Pretínanie napred z dĺžok

Úlohy

Nastavenia

Výstup

o programe

Nakresli situáciu F12

Č. B.	Y [m]	X [m]	mXY [m]
501	578413,326	1128315,218	0,008
502	578526,268	1128341,978	0,006

Merané hodnoty (prip. ich stredné chyby)

dĺžka 1

83,548

0,003

dĺžka 2

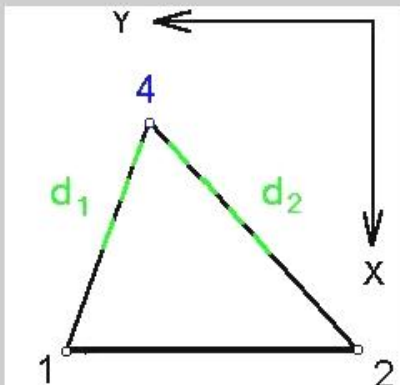
91,783

0,001

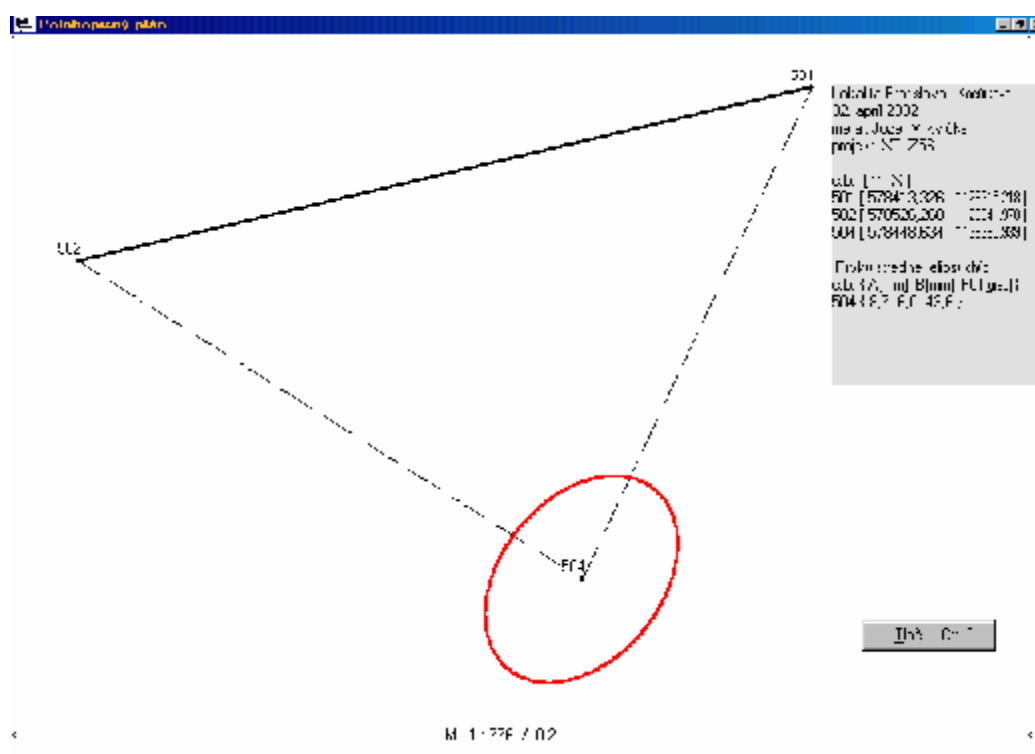
Výsledky:

Rataj

Č. B.	Y [m]	X [m]	mXY [m]
504	578448,634	1128390,939	0,0075



Obr.9 Pretínanie napred z dĺžok



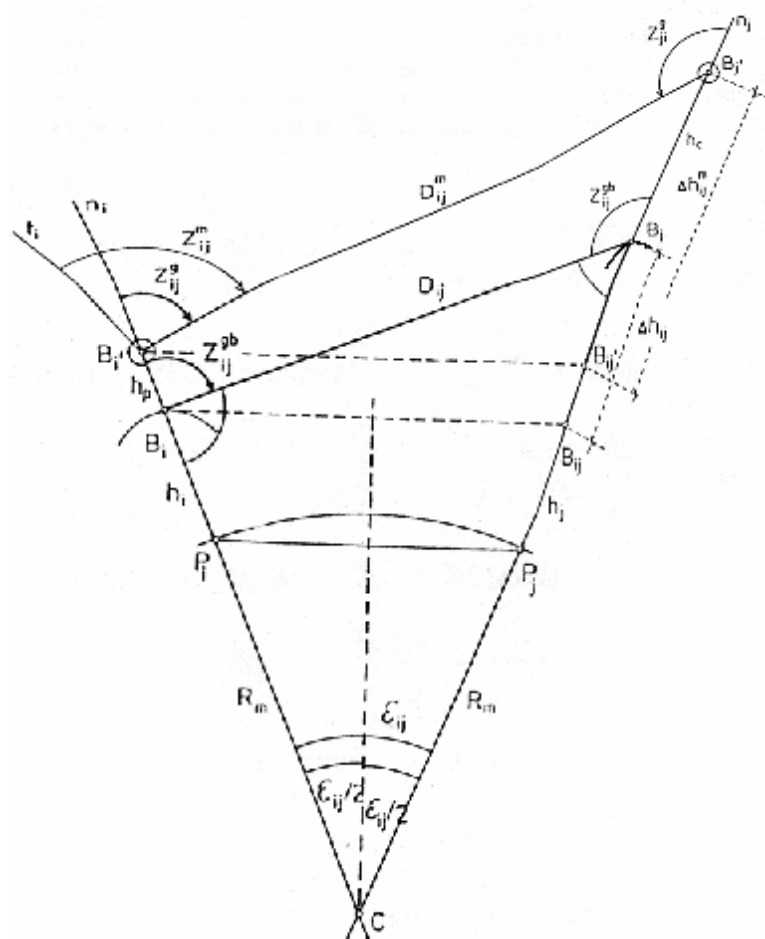
Obr. 10 Grafické znázornenie situácie bodov

Ako vidíme, prostredie programu je dnešnému užívateľovi známe a zrozumiteľné, využíva všetky výhody ovládania myšou, ale zároveň zachováva princíp plného prístupu z klávesnice.

Podobne i náš ďalší program využíva toto známe užívateľské rozhranie, avšak je už vytvorený v jazyku Delphi a vizuálnom vývojovom prostredí Borland Delphi 6.0, ktoré ponúka veľmi jednoduché riešenie nášho ďalšieho geodetického problému.

Redukcie niektorých meraných veličín

Pri meraní priestorových dĺžok vodorovných a zenitových uhlov, trigonometrických prevýšení v polohových bodových poliach, zmerané hodnoty D^m , w^m , z^m , Dh^m sa vzťahujú na stredy prístrojov a cieľových značiek (terčov), resp. na ich spojnice. V procesoch spracovania sietí treba však použiť hodnoty meraných veličín vzťahujúcich sa na meračské značky fyzických geodetických bodov, resp. na ich spojnice. Pritom sa predpokladá, že referenčná plocha Besselovho elipsoidu pre takto meraný priestor sa môže nahradiť sférickou plochou so stredným polomerom R_m , a že bodové pole pozostáva okrem určených bodov aj s bodov daných nielen súradnicami v S-JTSK, ale i elipsoidickými výškami h , ktoré boli získané napr. z meraní metódou GPS (Obr.11).



Obr. 11 Schéma meraných a redukovaných parametrov

[3] ukazuje, že daný problém je jednoznačne matematicky definovateľný a riešiteľný, a teda je vhodný pre automatizáciu na platforme osobných počítačov prostredníctvom nášho programu.. Navyše, aplikovaním zákona o hromadení chýb sme model rozšírili o charakteristiky presnosti redukovaných veličín. Geodét okrem týchto dostáva okamžite a s vysokou presnosťou i hodnotu, ktorá ho pri trigonometrickom meraní prevýšenia zaujíma najviac – výsledné prevýšenie. Na základe charakteristík presnosti tak môže promptne posúdiť potrebu zvýšenia presnosti meraných parametrov.

Prostredie nášho softvéru s pracovným názvom Geo 4 pozostáva zo základných prvkov aplikačného okna, hlavného programového menu a palety, ktorá obsahuje tri stránky: Vstup, Výstup a Schéma.

Vstupné hodnoty zadávame na stránke Vstup do polí zoradených v tabuľke. Riadky predstavujú jednotlivé merania na stanovisku, stĺpce zas konkrétne parametre vstupujúce do výpočtu. Niektoré už majú prednastavenú hodnotu 0, čo má uľahčiť zadávanie hodnôt v prípade, že ich vo výpočte neuvažujeme. Výpočet presnosti môžeme navyše deaktivovať prepínačom v pravom hornom rohu okna. Na stránke sa nachádzajú i tlačítka pre uloženie vpísaných a načítanie skôr uložených údajov, a vyčistenie vstupných polí. Tieto príkazy sú prístupné aj z hlavného menu.

Verní názorným príkladom z predošlých tematických častí, namodelujme zopár konkrétnych meraní (Obr. 12).

Geo 4 v.0.896

Súbor Pomoc

Vstup Výstup Schéma ☒ Počítaj presnosť

Stanovisko	Výška bodu nad elipsoid.		Výška Prístroja		Cieľ	Výška cieľu		Šikmá dĺžka		Zenitový uhol	
	hi [m]	m h i	hp [m]	m h p		hc [m]	m h c	D ij [m]	m D ij m	Z ij [g]	m Z ij m
501	110,5	0	1,432	0,001	506	1,806	0,001	805,631	0,002	88,3047	0,0006
502	110,5	0	1,432	0,001	504	0,789	0,001	698,345	0,002	106,5731	0,0006
		0		0			0		0		0

Zložky zvislicovej odchýlky a jej azimut

Ksi i		Eta i		Alfa ij alfa [g]
ksi i ["]	m ksi	eta i ["]	m eta	
1	0	3	0	30
2	0	5	0	160
0	0	0	0	0

Nové Údaje
Načítaj Údaje
Ulož Údaje

Vypočítaj

Obr. 12 Vstupná stránka palety programu

Po stlačení tlačítka pre výpočet (v pravom dolnom rohu) program aktivuje stránku Výstup. Zobrazia sa výsledné hodnoty, ktorých číselný formát možno priamo meniť, konkrétne počet

zobrazených desatinných miest, a tvar – či už fixný (Fixed, napr.602,03284) alebo exponenciálny (Scientific, napr.6,0232E+2) (Obr.13).

Tretia stránka obsahue schému umiestnenia vstupných i výstupných veličín v teréne (Obr. 11).

Geo 4 v.0.896

Súbor Pomoc

Vstup Výstup Schéma

☒ Počítat presnosť

Stanovisko	Cieľ	Stredný uhol normál n1, n2 epsilon ij ["]		Elipsoidická výška cieľového bodu h ij [m]	
			M epsilon ij		M h ij
501	506	25,6064	0,0000	257,3374	0,0014
502	504	22,4568	0,0000	39,2003	0,0014

Presnosť: 4

Tvar: Fixed Fixed Scientific

Redukovaná šikmá dĺžka D ij [m]		Redukovaný geodetický zenitový uhol z ij gb [g]		Elipsoidické prevýšenie delta h ij [m]	
	M D ij		M z ij gb		M delta h ij
805,5626	0,0000	88,3344	0,0000	146,8374	0,0015
698,2789	0,0000	106,6318	0,0000	-71,2997	0,0014

Obr.13 Výstupná stránka palety programu

Záver

Ukázali sme si možnosti automatizácie riešenia typických geodetických problémov vo vývojových prostrediach , či už staršieho Borland C, alebo modernejších Borland C++ Builder a Borland Delphi. Všetky však, v rámci svojich možností, predstavujú pomerne jednoduchý a silný nástroj pre bežného geodeta, ktorý so základnými znalosťami programovania môže zefektívniť svoju každodennú prax automatizáciou najmonotónnejších a tak často i najúnavnejších výpočtových procesov. V dnešných časoch agresívnej infiltrácie počítačov do rôznych oblastí záujmu geodeta pozorujeme čoraz väčší dopyt po špecializovaných softvéroch, ktoré by dokonale uspokojili potrebu najjednoduchšieho riešenia vzniknutých problémov (úloh). Ako príklad spomeňme rôzne požiadavky na spracovanie údajov z automatického zberu meraných údajov totálnou stanicou a ich vstup pri návrhu grafickej prezentácie v ktoromkoľvek CAD systéme. Na mnohé požiadavky existujúce softvérové produkty nedokážu adekvátne odpovedať, tak prečo si nevytvoriť vlastný?

Zoznam použitej literatúry

- [1] VITÁSEK, J.: Geodézie. – Souřadnicové výpočty, VUT, Brno, 1988
- [2] VIRIUS, M.: C++ Builder verze 5.0 – podrobný průvodce, GRADA, Praha, 2000
- [3] GAŠINEC, J.: Redukcie niektorých meraných veličín, GaKO ročník 44/86, 1998,
číslo 9, str.189-192