

Dokumentácia k projektu pre predmety IZP a IUS

Iteračné výpočty

Projekt č.2

1.decembra 2014

Autor: Maroš Cocul'a, xcocul00@stud.fit.vutbr.cz
Fakulta Informačných Technológií
Vysoké Učení Technické v Brne

Obsah

1.	Úvod	1
2.	Analýza problému a princíp riešenia	1
2.1	Zadanie problému	1
2.2	Tangens	2
2.3	Taylorov polynóm	2
2.4	Tangens pomocou zreťazených zlomkov	3
3.	Riešenie problému.....	3
3.1	Voľba rozsahov	3
3.2	Analýza vstupných dát.....	3
3.3	Výpočet presnosti iterácií.....	4
4.	Špecifikácia testov	4
5.	Popis riešenia.....	5
5.1	Ovládanie programu	5
5.2	Implementácia programu	6
6.	Záver	6
A	METRIKY KÓDU	6

1. Úvod

Počítanie vzdialenosti objektu pomocou goniometrickej funkcie tangens, má viacero možných spôsobov implementácie. Spôsoby ako sa to dá vypočítať aj bez matematických knižníc sú zaujímavé. V riešení tohto problému sa vyskytlo viacero problémov, ktoré sme museli riešiť.

Tento dokument popisuje návrh a riešenie problémov, ktoré sa vyskytli v projekte. Konečná podoba programu funguje ako konzolová aplikácia. Porovnáva presnosť všetkých implementovaných spôsobov riešenia funkcie tangens. Počíta vzdialenosť meracieho prístroja od predmetu, ale aj výšku daného predmetu. Výšku v ktorej sa merací prístroj nachádza, si taktiež môžeme zvoliť.

V tomto dokumente nájdeme viacero kapitol. Kapitola číslo [2](#) sa zaoberá všeobecnou analýzou programu. Vysvetľujú sa v nej základné pojmy, ktoré sú potrebné k pochopeniu projektu. Kapitola [3](#) sa zaoberá priamo problémami pri riešení projektu. Ako napríklad riešenie počtu iterácií, ktoré sú potrebné pre presnosť výsledku na 10 desatinných miest. Ďalej analyzujeme rozsah dát s ktorými vie program pracovať, ako napríklad uhli. V kapitole [4](#) sú popísané testovacie hodnoty a stavy, ktoré sú výsledkom týchto hodnôt. V testovacích hodnotách sú použité aj stavy s možnými komplikáciami. Táto kapitola zaručuje správnosť algoritmov a implementácií funkcií programu. V kapitole [5](#) sa popisuje konečná implementácia programu a ovládanie programu.

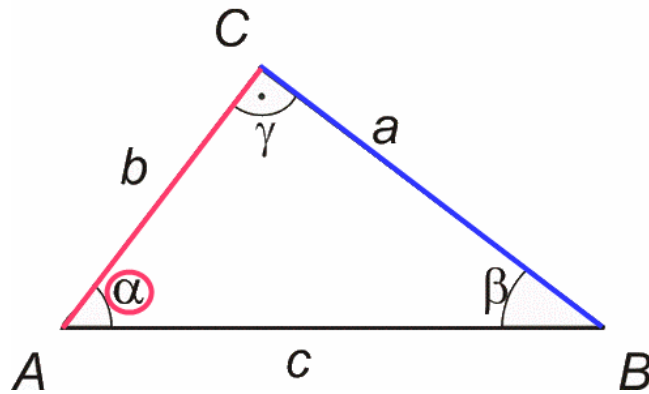
2. Analýza problému a princíp riešenia

2.1 Zadanie problému

Hlavnou úlohou je naprogramovať program v jazyku C. Tento program obsahuje implementovaný tangens uhla v radiánoch, pomocou Taylorovho polynómu a pomocou výpočtu zreťazených zlomkov. Program porovnáva presnosť rôznych metód výpočtu. Ďalej počíta vzdialenosť od predmetu a výšku predmetu. Program načítava údaje zo štandardného vstupu a výsledok vypisuje na štandardný výstup. Podľa zadaných argumentov program spracováva len žiadanú funkciu.

2.2 Tangens

Tangens je v matematike vyjadrený ako pomer protiľahlej a priľahlej odvesny v pravouhlom trojuholníku. Grafom takejto funkcie je tangentoida. Vyjadruje sa ako tg, zriedkavo aj ako tan, či tn.



Obr. 1 : Vyjadrený tangens v pravouhlom trojuholníku

$$\operatorname{tg} \alpha = \frac{a}{b}$$

2.3 Taylorov polynóm

Pomocou Taylorovho polynómu sa dajú približne vypočítať viaceré funkcie ako sin, cos, tg, arctg atď. Vo všeobecnosti vieme pomocou Taylorovho polynómu vyriešiť každú funkciu $f(x)$, ktorá má v okolí bodu a derivácie do $(n+1)$ -vého radu. Takáto funkcia sa dá vyjadriť aj ako :

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(n)}(a)}{n!}(x-a)^n + (R_n + 1)(x)$$

My sme potrebovali vyjadriť funkciu $\operatorname{tg}(x)$, z čoho nám vyplýva nasledujúca funkcia :

$$\tan(x) = x + \frac{x^3}{3} + \frac{2x^5}{15} + \frac{17x^7}{315} + \frac{62x^9}{2835} + \dots$$

Čím viac členov polynómu používame vo výpočte, tým presnejší výsledok získame.

2.4 Tangens pomocou zreťazených zlomkov

Funkcia tg sa dá riešiť aj pomocou rady zreťazených zlomkov. Oproti Taylorovmu polynómu je tento spôsob presnejší, a preto aj výhodnejší pre výpočet vzdialenosti v projekte. X udáva uhol a n rozvoj zreťazeného zlomku.

$$\tan(x) = \frac{x}{1 - \frac{x^2}{3 - \frac{x^2}{5 - \frac{x^2}{7 - \dots}}}}$$

Obr. 2 : Funkcia tangens vyjadrená pomocou zreťazených zlomkov

3. Riešenie problému

3.1 Voľba rozsahov

Program má pracovať s uhlami v rozsahu (0 ; 1,4>. Preto bolo potrebné ošetriť vstupné údaje. Keďže tg, ktorý používame na výpočet vzdialenosti a výšky, má byť presný na 10 desatinných miest, nemohli sme pracovať s premennými typu integer. Pre zabezpečenie proti pretečeniu sme používali dátový typ double.

3.2 Analýza vstupných dát

V zadaní je presne určené ako majú vyzeráť vstupné parametre, čiže sme to museli ošetriť. Na ošetrenie sme vytvorili funkciu s názvom zistiArgumenty, dátového typu integer. Rozlišujeme funkcie --tan, alebo výpočet výšky, ktorá má štyri rôzne spôsoby zadania argumentov. Taktiež sme museli ošetriť aby sa po zadaní parametra --help zobrazila nápoveda a program skončil.

Vo funkcii zistiArgumenty sme ošetrovali aj rozsah uhlov, ktorý bol v intervale (0 ; 1,4>. Ďalej sme museli ošetriť, aby sa nevyskytli čísla na mieste kde mali byť znaky a naopak. Napríklad pre výpočet vzdialenosti predmetu použijeme príkaz :
./proj2 -m A. Argument 1 musí byť znak -m, argument 2 musí byť číslo z intervalu, ktorý je vyššie uvedený.

3.3 Výpočet presnosti iterácií

Pre výpočet vzdialenosti a výšky, sme mali použiť tangens pomocou zreťazených zlomkov. Každou iteráciou je výsledok presnejší, pri zreťazených zlomkoch nie je potrebné vykonať 13 iterácií. Výsledok je presný na 10 desatinných miest skôr ako po 13-tich iteráciách, vid'. obrázok 3.

```
Maros@linux-klp6:~/Cecko/Projekt2> ./proj2 --tan 1.024 6 11
6 1.642829e+00 1.634327e+00 8.502803e-03 1.642829e+00 3.298801e-09
7 1.642829e+00 1.639216e+00 3.613451e-03 1.642829e+00 1.794542e-11
8 1.642829e+00 1.641294e+00 1.535615e-03 1.642829e+00 7.460699e-14
9 1.642829e+00 1.642177e+00 6.525932e-04 1.642829e+00 4.440892e-16
10 1.642829e+00 1.642552e+00 2.773337e-04 1.642829e+00 0.000000e+00
11 1.642829e+00 1.642712e+00 1.176550e-04 1.642829e+00 0.000000e+00
Maros@linux-klp6:~/Cecko/Projekt2> █
```

Obr. 3 : Výpis funkcie --tan 1,024 6 11

Preto sme do programu museli implementovať algoritmus ktorý ukončí výpočet hodnoty tangensu pri presnosti na 10 desatinných miest. Dokázali sme to odčítaním predchádzajúcej iterácie od iterácie aktuálnej. Rozdiel medzi iteráciami sme porovnali s konštantou presnosť. Ak bol rozdiel medzi iteráciami menší ako konštanta, výpočet sa skončil a hodnota aktuálnej iterácie bol hľadaný tangens.

Definícia konštanty : **const double presnosť = 1e-10;**

4. Špecifikácia testov

Pri riešení problému je potrebné overiť niekoľko oblastí. Chybnú syntax, hodnoty mimo rozsahu uhlov, iterácií či výšky meracieho prístroja.

TEST 1 : Chybná syntax parametrov = výpis chybovej hlášky.

```
-- help
-tan 1,1 2 5
--tan 1 N M
-t 1,2
-m A
-m A B
-c X -m A B
-c 20 -g 1 1,2
fsdfsdfs
```

TEST 2 : Parameter mimo povolený rozsah uhlov = výpis chybovej hlášky

```
--tan 1,5 2 3
--tan 0 2 3
-m 0
-m 1,5
-m 1,2 3
```

-c 20 -m 4,4
-c 20 -m 1,2 1,5

TEST 3 : Zvolenie počtu iterácií mimo rozsah programu = výpis chybovej hlášky

--tan 1,1 3 15

-- tan 1,1 0 2

TEST 4 : Výška meracieho prístroja mimo rozsah programu = výpis chybovej hlášky

-c 0 -m 1,1 1,2
-c 101 -m 1,1 1,2

TEST 5 : Správne zadané parametre = výpis výsledku na štandardný výstup

Vstup	Výstup
--help	Nápoveda
--tan 1 2 3	Výpis v podobe tabuľky
-m 1	9.6313892390e-01
-m 1 1.2	9.6313892390e-01 3.9773393454e+00
-c 50 -m 1	3.2104630797e+01
-c 50 -m 1 1.2	3.2104630797e+01 1.3257797818e+02

Tabuľka 1 : Výpis výsledkov na štandardný výstup

5. Popis riešenia

5.1 Ovládanie programu

Program ma len textové ovládanie. Po spustení programu, program očakáva správne zadané vstupné parametre. Pri použití parametra --help, program vypíše na konzolu stručnú nápovedu a skončí, neprevádza žiadne výpočty.

Po zadaní parametra - - tan, program očakáva na vstupe korektne zadaný reťazec parametrov. Ak parametre nie sú zadané korektne, program vypíše chybovú hlášku a skončí. Program vypíše výsledok len ak sú parametre zadané správne. Na obrazovke sa zobrazí tabuľka s výsledkami, v poradí zľava sa zobrazí počet iterácií, tangens z matematickej knižnice, tangens z funkcie `taylor_tan`, absolútna chyba medzi knižnicou a funkciou `taylor_tan`, tangens z funkcie `cfrac_tan` a absolútna chyba medzi knižnicou a funkciou `cfrac_tan`.

Pre výpočet vzdialenosti a výšky, máme viacero možností. Po zadaní parametra -m A, pričom A je uhol α , program vypíše do jediného riadka vzdialenosť od predmetu. Pri výpočte sa počíta s tým, že merací prístroj je implicitne vo výške 1,5m. Po zadaní -m A B, pričom A je uhol α a B uhol β , program vypíše vzdialenosť od predmetu aj výšku predmetu. Po zadaní -c X ... je zmenená výška v ktorej sa nachádza merací prístroj. Taktiež si môžeme vybrať, či

chceme len vzdialenosť od predmetu, alebo aj výšku pomocou argumentov A a B ako je uvedené vyššie.

5.2 Implementácia programu

Ako prvá sa spustí funkcia main, ktorá vyvolá ďalšie funkcie. Na začiatku sa ošetrí, či bol zadán aspoň jeden parameter. Ak nebol zadán parameter, program vypíše chybovú hlášku. Program pokračuje tým, že zistí požadovanú funkciu programu. Pre parameter `--help` sa spustí funkcia `zistiHelp`, ktorá overí správnosť všetkých argumentov. Ďalej sa spustí funkcia `help`, ktorá vypíše nápovedu. Ak parameter 1 nebol `--help` spustí sa funkcia `zistiArgumenty`, ktorá overuje správnosť všetkých parametrov na vstupe. Ak sú zadané správne, pre parameter `-tan` sa spustí funkcia `vypis`, pre parameter `-m` a `-c X` sa spustí funkcia `vzdialenost`. Funkcia `vypis` pracuje s ďalšími funkciami `taylor_tan`, `cfrac_tan` a `myAbs`. Na konzolu vypíše výsledok v podobe tabuľky. Funkcia `myAbs` spraví absolútnu hodnotu, čo potrebujeme pre porovnanie presnosti. Funkcia `vzdialenost` pracuje s funkciami `taylor_tan` a `cfrac_tan`. Ak parametre neboli zadané správne tak sa vypíše chybová hláška a program skončí.

6. Záver

Program počíta s iteráciami od 1 do 13. Viac iterácií by spresnilo výsledok, ale v našom prípade to nie je potrebné. Najmä u funkcie `cfrac_tan` kde je výsledok presný oveľa skôr ako na 13-tej iterácii. Do budúcnosti by sme však mohli pridať iterácie, čím by sme boli schopný vypočítať výsledok, aj na viac ako presnosť 10 desatinných miest. Toto rozšírenie by bolo vhodné najmä pre funkciu `taylor_tan`. Ako ďalšie rozšírenie by padlo vhod aj pri rozsahu uhlov. A to nie len zväčšením rozsahu ale aj možným počítaním so stupňami, nie len s radiánmi. Keďže nie celý svet používa rovnaké miery dĺžky, do programu by bolo vhodné implementovať aj tento výber.

Program bol úspešne testovaný a je tak pripravený na použitie v praxi. Je ho možné používať na operačných systémoch GNU/Linux ale aj MS Windows. Všetky požiadavky zo zadania boli splnené.

A METRIKY KÓDU

Počet súborov : 1 súbor

Počet riadkov zdrojového kódu : 176 riadkov

Veľkosť statických dát : 664B

Veľkosť spustiteľného súboru : 4461B