

Dokumentácia k projektu pre predmety IZP a IUS

Iteračné výpočty

projekt č. 2

12. apríla 2015

Autor: Viktor Jančík, xjanci09@stud.fit.vutbr.cz
Fakulta Informačních Technologii
Vysoké Učení Technické v Brně

Obsah

1	Úvod	1
2	Analýza problému a princíp jeho riešenia	1
2.1	Zadanie problému	1
2.2	Špecifiká zadania problému	2
2.3	Presnosť a počítanie s číslami s pohyblivou desatinnou čiarkou	2
2.4	Taylorov polynóm	3
2.5	Zreťazené zlomky	3
2.6	Princíp riešenia výpočtu výšky a vzdialenosti meraného objektu	3
3	Návrh riešenia problému	3
3.1	Výpočet Taylorového polynómu	4
3.2	Výpočet zreťazeného zlomku	4
3.3	Výpočet vzdialenosti a výšky objektu	4
4	Špecifikácia testov	4
5	Popis riešenia	6
5.1	Ovládanie programu	6
5.2	Voľba dátových typov	7
5.3	Špecifikácia výpočtu zreťazeného zlomku	7
5.4	Vlastná implementácia	7
5.5	Optimalizačné detaily	7
6	Záver	8
A	Metriky kódu	8

1 Úvod

Iteračné výpočty sú výpočty hodnôt pomocou iterácie, a teda opakovania nejakého výpočtu. S každou iteráciou sa výsledok tohto výpočtu čoraz viac približuje k presnému očakávanému výsledku výpočtu. V počítačoch je však málokedy niečo úplne presné, najmä keď pracujeme s číslami s desatinným rozvojom. Vždy hovoríme o akejsi úrovni presnosti, teda počet desatinných miest, v ktorom sa výpočet zhoduje s presnou hodnotou výsledku.

Tento dokument sa zaoberá návrhom a implementáciou programu pre výpočet funkcie tangens a výpočet veľkosti a vzdialenosti meraného objektu na základe uhla, ktorý zvierá s meracím prístrojom. Funkciu tangens program počíta iteračnými výpočtami s určitým počtom iterácií. Výsledný program je konzolová aplikácia, ktorá priama textový vstup zo štandardného vstupu, vykonáva tieto výpočty a vypisuje výsledok týchto výpočtov na štandardný výstup.

Dokument má viacero častí. Kapitola 2 sa zaoberá analýzou problému a možnými spôsobmi riešenia. Kapitola 3 sa zaoberá konkrétnym návrhom riešenia a v kapitole 5 je popísaná implementácia riešenia tohto problému. V kapitole 4 je niekoľko testovacích vstupov a výstupov, ktoré boli použité pre kontrolu správnosti programu.

2 Analýza problému a princíp jeho riešenia

V tejto kapitole sa budem hlbšie venovať problematike počítania s číslami s desatinným rozvojom reprezentovanými v počítači ako čísla s pohyblivou desatinnou čiarkou. Ďalej sa budem zaoberať počítaním funkcie tangens vrámci limitácií počítačov, spôsobmi počítania funkcie tangens a zadaním projektu.

2.1 Zadanie problému

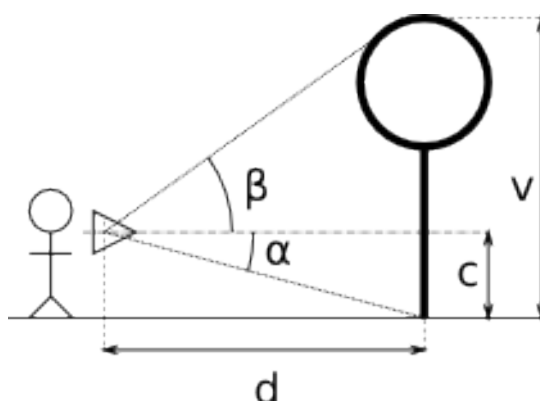
Úlohou projektu vytvoriť konzolovú aplikáciu v jazyku C, ktorá dokáže vypočítať vzdialenosť a výšku meraného objektu pomocou údajov zo senzoru, v podobe uhlov α , β a výšky meraného prístroja. Situácia je ilustrovaná obrázkom 1.

Uhol α reprezentuje uhol, ktorý zvierá merací prístroj s meraným objektom od úrovne výšky meracieho prístroja po spodnú hranicu objektu. Uhol β reprezentuje uhol, ktorý zvierá merací prístroj s meraným objektom od úrovne výšky meracieho prístroja po hornú hranicu objektu. Dĺžka c značí výšku meracieho prístroja od pomyslenej zeme. Dĺžka v značí výšku meraného objektu a dĺžka d značí vzdialenosť objektu od meracieho prístroja.

Pre výpočet vzdialenosti a výšky meraného objektu budeme potrebovať výpočet funkcie tangens uhlov α a β .

Ďalšia požiadavka pre program, je samostatný výpočet funkcie tangens na špecifikovaný počet iterácií, troma spôsobmi, a to funkciou `double tan(double x)`; z knižnice `math.h`, Taylorovým polynómom a metódou zreťazených zlomkov. V tomto prípade je výstupom programu výsledná hodnota tangens pre každú iteráciu v určenom rozsahu iterácií, pre každú z týchto troch metód¹. Súčasťou výstupu je zároveň porovnanie presnosti každej z týchto metód pre danú iteráciu, voči presnej hodnote.

¹Výsledok funkcie `tan` z knižnice `math.h` nie je ovplyvnený počtom iterácií a slúži ako referenčná hodnota.



Obr. 1: Ilustračný obrázok popisujúci vzťahy medzi meracím prístrojom a meraným objektom.

2.2 Špecifiká zadania problému

Zadanie špecifikuje nutnosť výpočtu funkcie tangens pre výpočet výšky a vzdialenosti meraného objektu metódou zretazovaných zlomkov s presnosťou aspoň na desať cifier za desatinnou čiarkou.

Pre uhol α , β a výšku c , zadanie udáva rozsah hodnôt, pre ktoré musí mať program správny výstup:

$$0 < \alpha \leq 1.4 < \frac{\pi}{2} \quad (1)$$

$$0 < \beta \leq 1.4 < \frac{\pi}{2} \quad (2)$$

$$0 < c \leq 100 \quad (3)$$

Pre výpočet Taylorového polynómu a zretazného zlomku, zadanie vyžaduje implementovanie funkcií s týmito prototypmi:

```
double taylor_tan(double x, unsigned int n);
double cfrac_tan(double x, unsigned int n);
```

Pre počet iterácií I výpočtu platí:

$$0 < I < 14 \quad (4)$$

2.3 Presnosť a počítanie s číslami s pohyblivou desatinnou čiarkou

V počítačoch sú všetky dáta uložené ako reťazce bitov (núl a jednotiek). Konečným počtom bitov sa dá reprezentovať konečný počet unikátnych hodnôt. Napríklad dátový typ `double`, ktorý používa 64 bitov pamäte, dokáže reprezentovať 15 až 17 platných decimálnych číslíc².

Z toho vyplýva, že pre akékoľvek iteratívne výpočty s týmto dátovým typom sa oplatí počítať len do presnosti (maximálne) 10^{-17} . Po dosiahnutí tejto presnosti, ďalšie iterácie nemajú vplyv na výsledok výpočtu, pretože cifry, ktoré sa menia nie sú zobraziteľné v 64 bitoch pamäte.

Na porovnanie, dátový typ `float`, ktorý používa 32 bitov pamäte, dokáže reprezentovať 6 až 9 platných decimálnych číslíc².

²Presný počet závisí od štandardu, podľa ktorého sa riadi implementácia v danom programovacom jazyku.

2.4 Taylorov polynóm

Taylorov polynóm je spôsob reprezentácie funkcie ako nekonečnú sumu členov, vypočítaných z hodnôt derivácií (rôznych stupňov) funkcie pre konkrétnu hodnotu. Všeobecný tvar Taylorového polynómu je:

$$\sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!} (x-a)^n \quad (5)$$

Kde $f^{(n)}(a)$ reprezentuje n -tú deriváciu funkcie f v bode a .

Špecifický tvar Taylorového polynómu pre funkciu $\tan(x)$ je:

$$\tan(x) = x + \frac{x^3}{3} + \frac{2x^5}{15} + \frac{17x^7}{315} + \frac{62x^9}{2935} + \dots \quad (6)$$

V kontexte tohto projektu uvažujeme pripočítanie ďalšieho člena postupnosti k výsledku ako jednu iteráciu výpočtu.

2.5 Zreťazené zlomky

Metóda zreťazených zlomkov je metóda výpočtu funkcie pomocou nekonečného reťazca zlomkov. Pre výpočet funkcie $\tan(x)$ má zreťazený zlomok tvar 7 alebo 8:

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

$$\tan(x) = \frac{1}{\frac{1}{x} - \frac{\frac{1}{3 - \frac{1}{5 - \frac{1}{7 - \dots}}}}{x}} \quad (8)$$

V kontexte tohto projektu uvažujeme pridanie ďalšej úrovne zlomku ako jednu iteráciu výpočtu.

2.6 Princíp riešenia výpočtu výšky a vzdialenosti meraného objektu

Je viacero možností ako vypočítať výšku a vzdialenosť meraného objektu. Zadanie projektu si vyžaduje pre tento výpočet použitie funkcie pre výpočet tangens pomocou zreťazených zlomkov. Vzhľadom na tento fakt a vstupné hodnoty (uhol α , β ; výška c) je intuitívne vypočítať výšku a vzdialenosť objektu pomocou trigonometrických vzťahov, na trojuholníkoch, ktoré zadané uhly a veľkosti špecifikujú.

3 Návrh riešenia problému

Požiadavkami v zadaní problému je riešenie takmer presne definované. Presnejšie, aby sme splnili tieto požiadavky, dá sa použiť akurát jedno riešenie. V tejto kapitole opíšem spôsob implementácie výpočtov potrebných pre podproblémy môjho programu, zatiaľ čo v kapitole 5 sa budem venovať implementačným detailom.

3.1 Výpočet Taylorového polynómu

Taylorov polynóm pre funkciu tangens som vypočítal podľa vzorca 6. Pre tento vzorec však potrebujem vypočítať hodnoty koeficientov pre každý člen postupnosti. Vzorec pre výpočet týchto koeficientov je netriviálny, preto, keďže poznám maximálny počet iterácií, ktoré budú požadované, je efektívnejšie uložiť hodnoty týchto koeficientov pre prvých 13 iterácií do poľa.

Sú to tieto koeficienty v čitateli[1]:

1, 1, 2, 17, 62, 1382, 21844, 929569, 6404582, 443861162, 18888466084,
113927491862, 58870668456604, 8374643517010684, 689005380505609448,
129848163681107301953, 1736640792209901647222, 418781231495293038913922

A tieto hodnoty pre koeficienty v menovateli[2]:

1, 3, 15, 315, 2835, 155925, 6081075, 638512875, 10854718875, 1856156927625,
194896477400625, 49308808782358125, 3698160658676859375,
1298054391195577640625, 263505041412702261046875

3.2 Výpočet zreťazeného zlomku

Výpočet zreťazeného zlomku je trochu zložitejší (aspoň na pohľad), pretože narozdiel od Taylorového rozvoju kde jasne akumulujeme členy postupnosti, pri zreťazenom zlomku, nie je na prvý pohľad jasné, čo vlastne máme akumulovať.

Pri zreťazenom zlomku akumulujeme menovateľ zlomku odspodu, pričom uvažujeme, že tento menovateľ v nekonečne nabera hodnotu 0.

Pre výpočet som si zvolil vzorec 7. Po akumulácii zlomku na niekoľko úrovní (dané počtom iterácií), predelím čitateľ týmto menovateľom a dostanem správnu hodnotu.

3.3 Výpočet vzdialenosti a výšky objektu

Z trigonometrie sa dajú ľahko odvodiť nasledujúce vzorce pre výpočet vzdialenosti a výšky objektu (symboly podľa obrázku 1):

$$d = \frac{c}{\tan(\alpha)} \quad (9)$$

$$v = \tan(\beta) * d + c \quad (10)$$

Presnosť výpočtu funkcie tangens na desať cifier po desatinnej čiarky zaručujem opakovaným počítaním funkcie tangens, až kým rozdiel výsledkov dvoch po sebe idúcich iterácií výpočtu nie je menší ako 10^{-10} .

4 Špecifikácia testov

Na základe hraníc vstupných hodnôt a návrhu riešenia vznikajú krajné situácie, pre ktoré treba otestovať správnosť algoritmu.

Test 1: Prvý vzorový príklad vstupu a výstupu zo zadania projektu.

vstup					
./proj2 --tan 1.024 6 10					
očakávaný výstup					
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	1.794520e-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

Test 2: Druhý vzorový príklad vstupu a výstupu zo zadania projektu.

vstup	
./proj2 -m 0.3 0.9	
očakávaný výstup	
4.8490922156e+00	
7.6106234032e+00	

Test 3: Tretí vzorový príklad vstupu a výstupu zo zadania projektu.

vstup	
./proj2 -c 1.7 -m 0.15 1.3	
očakávaný výstup	
1.1248205560e+01	
4.2217188781e+01	

Test 4: Tangens uhla 0.

vstup					
./proj2 --tan 0 1 3					
očakávaný výstup					
1	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
2	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
3	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00

Test 5: Tangens uhla väčšieho ako $\frac{\pi}{2}$ (využitie periódy funkcie tangens).

vstup	
./proj2 --tan 2 11 13	

očekávaný výstup

```
11 4.576576e-01 2.132589e+00 -1.674932e+00 4.576576e-01 0.000000e+00
12 4.576576e-01 2.132589e+00 -1.674932e+00 4.576576e-01 0.000000e+00
13 4.576576e-01 2.132589e+00 -1.674932e+00 4.576576e-01 0.000000e+00
```

Test 6: Tangens záporného uhla (kontrola symetrie).

vstup

```
./proj2 --tan -1.024 8 10
```

očekávaný výstup

```
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
```

5 Popis riešenia

Implementácia programu vychádza zo záverov predošlých kapitol. V tejto kapitole sa budem zaoberať detailami implementácie v jazyku C.

5.1 Ovládanie programu

Program je implementovaný ako konzolová aplikácia. Spustiť ju je možné s rôznymi vstupnými parametrami³:

```
./proj2 --help
./proj2 --tan A N M
./proj2 [-c X] -m A [B]
```

V prípade, že sa vstupné parametre nebudú zhodovať ani s jedným z týchto vzorov, program sa ukončí, bez akéhokoľvek výstupu. Ak sa nedodrží rozsah hodnôt špecifikovaný v zadaní projektu, správanie programu nie je definované.

Pre výpočet funkcie tangens (volanie s parametrom `--tan`) má výstup formát:

I M T TE C CE

Vypísaný formátom `"%d %e %e %e %e %e\n"`, funkciou `printf`, kde:

- I značí počet iterácií iteračného výpočtu,
- M výsledok z funkcie `tan` matematickej knižnice,
- T výsledok z funkcie `taylor_tan`,
- TE absolútnu chybu medzi výpočtom pomocou Taylorového polynómu a matematickou knižnicou,
- C výsledok z funkcie `cfrac_tan` a

³Presný popis vstupných parametrov je rozsiahly a vypíše sa v prípade spustenia programu s parametrom `--help`.

- CE absolútnu chybu medzi výpočtom pomocou zreťazených zlomkov a matematickou knižnicou.

Pre výpočet vzdialenosti a výšky (volanie s parametrom `--m`) ma výstup formát⁴:

vzdialenosť výška

5.2 Voľba dátových typov

Pre hodnoty výpočtov funkcie tangens (všetky metódy) som použil dátový typ `double`, pretože by dátový typ `float` nedokázal zobraziť požadovanú presnosť hodnôt (sekcia 2.2), z dôvodov opísaných v sekcii 2.3.

5.3 Špecifikácia výpočtu zreťazeného zlomku

V predchádzajúcej kapitole (sekcia 3.2) som načrtnol problematiku akumulovania menovateľa pre výpočet zreťazeného zlomku. Presná implementácia tohto procesu je:

```
for(int i=(n+1); i>0; --i) {
    denominator = (2*i - 1) - xx/denominator;
}
```

Kde `n` je počet iterácií, `xx` je $u\text{hol}^2$ a `denominator` je menovateľ zreťazeného zlomku. Počiatočná hodnota `denominator` je 0 (menovateľ v nekonečne).

5.4 Vlastná implementácia

Parametre zo štandardného vstupu sú spracovávané vo funkcii `main`. Na základe tohto spracovania sa spustí buď funkcia `comp_tan`, ktorá je zodpovedná, za spúšťanie `taylor_tan`, `cfrac_tan` a správne vypisovanie a formátovanie výstupu pre výpočet funkcie tangens, alebo sa spustí funkcia `measures`, ktorá pomocou funkcie `cfrac_tan` a vzťahov zo sekcie 3.3 vypočíta a vypíše vzdialenosť a výšku meraného objektu, tiež v správnom formáte. V prípade spustenia programu so vstupným argumentom `--help`, sa spustí funkcia `print_desc` a vypíše popis funkcií programu.

Program taktiež zahŕňa pomocnú funkciu `power` ktorá vráti n -tú mocninu daného čísla (používa sa pre `taylor_tan`).

Koeficienty pre výpočet Taylorového polynómu funkcie tangens sú uložené v poliach `taylorNum` a `taylorDenom`.

5.5 Optimalizačné detaily

Pre optimalizáciu výpočtu som použil globálnu štruktúru `partialResult`, ktorá obsahuje číslo iterácie výpočtu (alebo stupeň mocniny - v prípade `prPower`) a čiastočný výsledok. Konkrétne, `prTaylor`, sa používa na optimalizáciu výpočtu Taylorového polynómu. Samozrejme by globálna premenná nemusela byť použitá v prípade, že by funkcia `taylor_tan` nemala vopred stanovený prototyp, ktorý som musel dodržať.

⁴Obidve hodnoty sú vypísané formátom `%.10e`

V prípade výpočtu zretazeneho zlomku, sa týmto spôsobom výpočet optimalizovať nedal, pretože čiastočné výpočty, a teda čiastočná akumulácia menovateľa zlomku, nemá pre konkrétny počet iterácií význam.

Funkcia **power** je z dôvodu jednoduchosti použiteľná len pre jednu hodnotu počas celého behu programu. To však správnosti tohto programu neprekáža.

6 Záver

Program spĺňa všetky požiadavky zo zadania problému. Navyše program optimalizuje opakovaný výpočet Taylorového polynómu alebo mocniny konkrétneho čísla. Program by sa dal ešte vylepšiť sprísnením kontroly vstupných argumentov z príkazového riadku, ale pre korektný (podľa špecifikácie zadania problému) vstup, program vždy vráti korektný výstup.

Program bol vyvíjaný a testovaný na operačnom systéme Ubuntu (Linux), ale mal by fungovať pre všetky platformy, ktoré podporujú jazyk C a štandard C99, keďže program nepoužíva žiadne systémovo závislé funkcionality.

Literatúra

[1] A002430 - OEIS. [online]. [cit. 2014-12-05]. Dostupné z: <https://oeis.org/A002430>

[2] A156769 - OEIS. [online]. [cit. 2014-12-05]. Dostupné z: <https://oeis.org/A156769>

A Metriky kódu

Počet súborov: 1 súbor

Počet riadkov zdrojového kódu: 137 riadkov

Veľkosť statických dat: 936B

Veľkosť spustiteľného súboru: 5396B (systém Linux, 32 bitová architektúra, pri preklade bez ladiacich informácií)