Dokumentácia k projektu pre predmety IZP a IUS

Iteračné výpočty

Projekt č.2

29. októbra 2014

Autor: Peter Tisovčík, xtisov00@stud.fit.vutbr.cz

Fakulta Informačních Technologií

Vysoké Učení Technické v Brně

Obsah

1 Úvod

Tangens patrí medzi goniometrické funkcie tak isto ako aj sínus, cosínus. Hlavným rozdielom oproti ostatným goniometrickým funkciám je ten, že tangens pracuje s odvesnami, nepracuje s preponami. Tangens nám slúži na výpočet odvesny, ak poznáme uhol a odvesnu alebo na výpočet uhla ak poznáme odvesny v pravouhlom trojuholníku, preto ho využijeme na výpočet vzdialenosti a výšky meraného objektu.

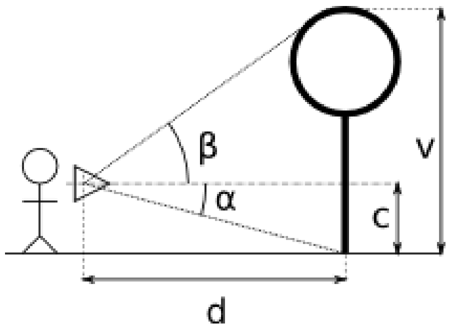
Dokument pojednáva o návrhu a implementácií výpočtu vzdialenosti a výšky meraného objektu pomocou zadaných údajov. Program je vytvorený ako konzolová aplikácia. Dáta pre program sa načítavajú zo štandardného vstupu a vypisujú sa na štandardný výstup. V prípade chybných údajov zo štandardného vstupu je vypísaná chyba. Program má niekoľko prepínačov, na výpočet vzdialenosti meraného objektu a na porovnanie tangensu z matematickej knižnice a výpočtu tangensu pomocou Taylorovej rady a zreťazených zlomkov v zadaných iteráciách.

Dokument sa skladá z niekoľkých častí. V kapitole 2 analyzujem jednotlivé problémy spojené s výpočtom tangensu pomocou zreťazených zlomkov a Taylorovej rady a výpočtu vzdialenosti a výšky meraného objektu. V 3. Tretej kapitole navrhujem riešenie výpočtu tangensu pomocou spomenutých metód, odvodenie presnosti pri výpočte tangensu pomocou zreťazených zlomkov a v závere tejto kapitoly sa nachádzajú testy, ktoré demonštrujú funkčnosť programu a ošetrenie chybových stavov. V 4. Kapitole popisujem ovládanie programu a vlastnú implementáciu riešenia problému.

2 Analýza problému a princíp jeho riešenia

2.1 Zadanie problému

Cieľom projektu bolo implementovanie výpočtu vzdialenosti a výšky objektu pomocou matematický operácii: +, -, *, / . Program bolo treba vytvoriť v programovacom jazyku C podľa štandardu C99 (ISO/IEC 9899:1999). Program načíta uhol A , slúži na výpočet vzdialenosti meraného objektu. Pri výpočte vzdialenosti a výšky môže načítať aj nepovinné údaje a to výšku meracieho prístroja, ktorá je implicitne nastavená na 100 alebo uhol B pomocou, ktorého vypočítame výšku meraného objektu. Uhly do programu sa zadávajú v radiánoch. Na výpočet vzdialenosti a výšky treba použiť funkciu tangens, ktorú vytvoríme pomocou zreťazených zlomkov a odvodíme jej presnosť na 10 desatinných miest. V programe by mala byť implementovaná funkcia na porovnanie presnosti tangensu z matematickej knižnice a výpočtu pomocou Taylorovej rady a zreťazených zlomkov v zdaných iteráciách zo štandardného vstupu. V programe netreba zabudnúť ošetriť chybové stavy ale aj výnimočné stavy ako zadanie textu namiesto uhla, zadanie neznámeho argumentu alebo zlej syntaxi argumentov.



Obrázok 1 Označenie uhlov a neznámych

2.2 Taylorov rozvoj - tangens

Tieto rozvoje boli dôležité hlavne v dobe, keď nebola k dispozícií výpočtová technika a všetky výpočty bolo potrebné robiť manuálne. Tieto rozvoje museli byť vytvorené, tak aby sa s nimi dobre pracovalo a zároveň musel tento rozvoj definovanú funkciu pokiaľ, čo najpresnejšie nahradzovať. Najviac sa používali Taylorove rozvoje funkcií do mocninových rád.

Mnoho zložitých funkcií je veľmi náročné zobraziť, poprípade predstaviť si ich ako napríklad základné funkcie tangens, sínus alebo cosínus. Tieto elementárne funkcie nadobúdajú najmä iracionálne hodnoty, ktoré sa nedajú presne vyčísliť ako napríklad π. Taylorov rad, pre ktorý platí, že súčet všetkých členov rady je výsledná funkcia. Konvergencia je veľmi rýchla pre *x* blízke nule. Pre funkciu tangens vyjadrenú pomocou Taylorovho rádu platí:

$$tan(x) = x + \frac{x^3}{3} + \frac{2x^5}{15} + \frac{17x^7}{315} + \frac{62x^9}{2835} + \cdots \qquad (2.1)$$

Tabuľka 1 Hodnoty pre výpočet Taylorovho rádu pre prvých 13 hodnôt

Číslo rádu	Hodnota čitateľa	Hodnota menovateľa
1	1	1
2	1	3
3	2	15
4	17	315
5	62	2835
6	1382	155925
7	21844	6081075
8	929569	638512875
9	6404582	10854718875
10	443861162	1856156927625
11	18888466084	194896477400625
12	113927491862	49308808782358125
13	58870668456604	3698160658676859375

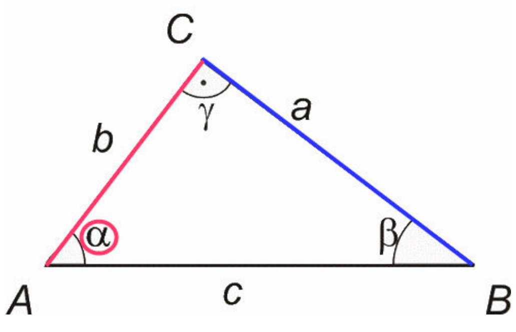
2.3 Zreťazené zlomky – tangens

Okrem Taylorovho rozvoja sa na výpočet tangensu dajú použiť zreťazené zlomky. Zreťazené zlomky sú oveľa jednoduchšie ako Taylorove rady a konvergujú rýchlejšie ako Taylorové rady. Zreťazený zlomok pre výpočet tangensu sa dá napísať v tvare:

$$tan(x) = \cfrac{x}{1 - \cfrac{x^2}{1 - \cfrac{x^2}{3 - \cfrac{x^2}{5 - \cfrac{x^2}{9 - \cfrac{x^2}{\dots}}}}}} \tag{2.2}$$

2.4 Výpočet vzdialenosti a výšky – Goniometrické funkcie

Na výpočet vzdialenosti a výšky (Obrázok 1) je možné využiť funkcie ako je sínus, cosínus a tangens. V našom prípade, budeme využívať funkciu tangens, ktorá nám vypočíta uhol. Tangens oproti sínusu a cosínusu pracuje len s odvesnami, nepracuje s preponou.



Obrázok 2 Pravouhlý trojuholník

Tangens Uhla α vypočítame ako:

$$tan(\alpha) = \frac{Dlžka\ protiľahlej\ odvesny}{Dĺžka\ priľahlej\ odvesny} \approx tan(\alpha) = \frac{|a|}{|b|} \tag{2.3}$$

V prípade výpočtu strany si predchádzajúcej rovnice vyjadríme neznáme nasledovne:

$$|a| = tan(\alpha) * |b| \quad \text{alebo} \quad /b/= \frac{|a|}{tan(\alpha)} \tag{2.4}$$

3 Návrh Riešenia problému

Po analýze problému som si uvedomil, že hlavným problémom bude vytvorenie efektívneho a správneho algoritmu na výpočet tangensu pomocou Taylorovej rady a zreťazených zlomkov. Ďalšou výzvou bude správne odvodenie počtu iterácií na požadovanú presnosť na výpočet vzdialenosti a výšky. Netreba však zabudnú ošetriť chybové stavy pri vstupných údajoch ale tak isto aj pri výpočtoch. Bude treba si vytvoriť štruktúry a funkcie, ktoré problém dekomponujú. Taktiež si budeme musieť zvoliť vhodné dátové štruktúry pre reprezentáciu dát.

3.1 Výpočet tangensu pomocou Taylorovej rady

Z definovaného vzťahu pre výpočet Taylorovej rady (Rovnica 2.1) si vieme vytvoriť podprogram na výpočet rady pre definovaný počet iterácii. Pre výpočet tangensu potrebujeme uhol a pre výpis v danej iterácií zase číslo iterácie, v ktorej sa má daný uhol vypočítať. Konštanty (Tabuľka 1) si uložíme do poľa. Pri konkrétnom implementovaní nesmieme zabudnúť, že prvý člen rádu je vlastne zadaný uhol a preto nemusíme nič počítať a výsledok je zadaný uhol v radiánoch. Ostatné rady vypočítame ako:

$$tan(x) = PredchadzajuceCleny + \frac{Uhol * Hodnota\ čitateľa}{Hodnota\ menovateľa} \tag{3.1}$$

Hodnota *PredchadzajuceCleny* je súčet predchádzajúcich členov v postupnosti. Hodnotu *Uhol* vypočítame ako:

$$Uhol = Uhol * DruhaMocninaUhla \tag{3.2}$$

Na výpočet druhej mocniny môžeme použiť matematickú funkciu alebo môžeme využiť vzťah:

$$DruhaMocninaUhla = Uhol * Uhol \tag{3.3}$$

Týmto pádom nemusíme vykonávať pri každom ráde výpočet určitej mocniny ale stačí nám hodnotu *Uhol* vynásobiť číslom *DruhaMocninaUhla.* Toto zjednodušenie výpočtu nám ušetrí inštrukcie, ktoré by počítač musel vykonávať navyše.

3.2 Výpočet tangensu pomocou zreťazených zlomkov

Vzťah pre výpočet (Rovnica 2.2) tangensu pomocou zreťazených zlomkov je definovaná opačne ako u Taylorovej rady, pretože pri tomto výpočte postupujeme od konca, čiže potrebujeme vedieť počet iterácií, ktoré nám treba na uvažovanú presnosť. Uhol pre výpočet sa udáva v radiánoch a v prípade, že je zadaná len jedna iterácia, tak použijeme vzťah:

$$tan(x) = \frac{Uhol}{1} \tag{3.4}$$

V prípade, že je zadaný väčší počet iterácií ako jedna musíme počítať odspodu. Vypočítame si hodnotu najnižšieho zlomku ako vzťah:

$$Zlomok = \frac{Uhol * Uhol}{Cislo} \tag{3.5}$$

Ďalej nesmieme zabudnú vypočítať hodnotu *Cislo*, od ktorého sa budú hodnoty jednotlivých zlomkov odpočítavať a následne pri každej iterácií sa odpočíta hodnota 2.

$$Cislo = (\ PocetIteracii * 2) - 1 \tag{3.6}$$

Po týchto jednotlivých výpočtoch môžeme začať počítať zreťazený zlomok, kde po vypočítaní nového zlomku ho priradíme ako novú hodnotu pre *Zlomok*:

$$Zlomok = \frac{Uhol * Uhol}{Cislo - Zlomok} \tag{3.7}$$

Ako posledný krok nesmieme zabudnúť urobiť posledný zreťazený zlomok, ktorý má v menovateli len prvú mocninu uhla:

$$\mathrm{tan(x)} = \frac{\mathrm{Uhol}}{1 - \mathrm{Zlomok}} \tag{3.8}$$

3.3 Odvodenie presnosti výpočtu

Pri výpočte vzdialenosti a výšky je treba zistiť koľko iterácii je potrebné na požadovanú presnosť. Presnosť je možné zistiť dvomi spôsobmi a to experimentálnou a analytickou metódou. V mojom prípade využijem experimentálnu metódu.

Presnosť som zistil pomocou funkcie, ktorá mala za úlohu vypočítavať tangens pomocou zreťazených zlomkov a porovnávať to z tangensom z matematickej knižnice. Absolútnu chybu som potom porovnal, či je menšia ako požadovaná presnosť ak bola považoval som toto číslo za minimálny počet iterácií, ak nebola zvýšil som počet iterácií. Túto presnosť som zisťoval v rozsahu od (0; 1.4> s prírastkami uhlu v jednotlivých iteráciách o $1e^{-1}$, presnosť mi vyšla na minimálne 8 iterácií. Avšak po zmenšení čísla, o ktoré pripočítavam na $1e^{-2}$ som prišiel na to, že minimálny počet iterácií je 9. Hodnotu prírastku som skúšal zmenšiť až na $1e^{-7}$ avšak minimálny počet iterácií sa mi už nezmenil. Z tohto zistenia som odvodil, že na zistenie presnosti na 10 desatinných miest potrebujeme aspoň 9 iterácii.

3.4 Výpočet vzdialenosti

Na výpočet vzdialenosti (Obrázok 1)som využil vzťah(Rovnica 2.3), z ktorého som si vyjadril vzdialenosť, čo je vlastne veľkosť priľahlej odvesny(Rovnica 2.4):

$$|Vzdialenosť| = \frac{|Výška\ meraného\ prístroja|}{\tan(\alpha)} \tag{3.9}$$

Na výpočet výšky meraného objektu som využil vypočítanú *Vzdialenosť,* z ktorej som vypočítal výšku od prístroja k meranému objektu (Rovnica 2.4) a k tejto výške som pripočítal výšku, v ktorej sa nachádzal merací prístroj.

$$|Výška\ objektu| = \tan(\beta) * |Vzdialenosť| + |Výška\ meracieho\ prístoja| \tag{3.10}$$

3.5 Analýza vstupných dát

Program očakáva len presne definované hodnoty, argumenty, prípadne hodnotu z definovaného intervalu. Uhly môžu byť iba v radiánoch v intervale (0; 1.4>. Pri zadaní väčšieho uhla ako je rozsah sú dve možnosti a to buď sa vypíše chyba alebo sa uhol snažiť prepočítať do daného kvadrantu a to odpočítaním $\frac{\pi}{2}$.

3.6 Špecifikácia testov

Program pracuje len s určitými rozsahom hodnôt a presne definovanými argumentmi, preto je treba otestovať program na: na neplatnú syntax argumentov, zle zadané hodnoty pre výpočet tangensu alebo neplatné hodnoty pre výpočet výšky a vzdialenosti objektu.

Test 1: Neúplná syntax → Detekcia chyby

--tan data

Test 2: Zadaný neplatný rozsah pre výpis iterácií → Detekcia chyby

--tan 1.024 1 100

Test 3: Správnosť výpočtu – porovnanie presnosti výpočtu → Predpokladaná správna hodnota

--tan 1.024 10 10
10 1.642829e+000 1.642552e+000 2.773337e-004 1.642829e+000 0.000000e+000

Test 4: Neplatná syntax → Detekcia chyby

X -m 10

Test 5: Zadaný neplatný rozsah pre výpis iterácií → Detekcia chyby

-c 120 –m 10

Test 6: Správnosť výpočtu – výpočet výšky a vzdialenosti → Predpokladaná správna hodnota

-c 1.7 -m 0.15 1.3
1.1248205560e+01
4.2217188781e+01

4 Popis riešenia

4.1 Ovládanie programu

Program je naprogramovaný ako konzolová aplikácia bez grafického rozhrania, dá sa ovládať len zadanými príkazmi. Pri spustení programu program očakáva niektorý z nižšie spomenutých argumentov.

--help program s týmto parametrom vypíše nápovedu, ktorá obsahuje vysvetlenie syntaxe jednotlivých príkazov a hodnoty, ktoré je možné zadať.

--tan A N M prepínač porovná presnosť výpočtu tangensu uhlu **A (v radiánoch)** medzi funkciou tangens z matematickej knižnice a výpočtom tangensu pomocou Taylorovho rozvoja a zreťazených zlomkov. Argumenty **N** a **M** definujú, v ktorom iteračnom výpočte má porovnanie prebiehať **0 < N <= M < 14.**

[-c X] -m A [B] prepínač vypočíta vzdialenosť a výšku meraného objektu pomocou zreťazených zlomkov. Prepínač **–c** nastavuje výšku meracieho prístroja, ak nie je definovaný, tak je výška implicitne nastavená na **1.5**. Uhol **A (v radiánoch)** určuje uhol pomocou(Obrázok 1), ktorého sa vypočítava vzdialenosť. Pokiaľ je zadaný uhol **B(v radiánoch)**, ktorý je nepovinný vypočíta sa aj výška meraného objektu.

Pokiaľ nie je zadaný argument syntakticky správny alebo je zadaný neplatný uhol, výška, program vypíše chybu na štandardný chybový výstup. V prípade syntakticky správneho argumentu a správnych hodnôt program vykoná požadovanú činnosť.

4.2 Vlastná implementácia

Parametre programu spracováva hlavná funkcia **main,** ktorá podľa toho volá funkcie. V prípade, že bol zadaný príkaz **--help** vypíše sa reťazec, ktorý obsahuje text nápovedy.

Ak je zavolaný príkaz, ktorý porovná presnosť výpočtu tangensu uhla, je zavolaná funkcia **tan_print**, ktorý zistí, či sú zadané údaje správne a ak áno, zavolá funkciu **cmp,** ktorá volá funkcie **tan, taylor_tan** a **cfrac_tan** a vypisuje z nich tangen zadaného uhla a ich absolútne chyby v danej iterácií.

Ak bol zadaný príkaz **–m ,** zavolá sa funkcia **m_print,** ktorá následne v sebe volá funkcie **cm_ab, cm_a, m_ab, m_a.** Tieto funkcie na základe zadaných argumentov vypočítajú vzdialenosť a prípadne aj výšku. Spomenuté funkcie v sebe volajú funkciu **cfrac_tan,** na výpočet vzdialenosti a výšky. Každá funkcia vracia chybový kód.

Chybové kódy sú definované v dátovej štruktúre **errcode** a na konci programu sa overuje, či bol nastavený nejaký chybový kód ak áno zistí sa, ktorý a vypíše sa definovaná chyba na štandardný chybový výstup.

5 Záver

Program na výpočet vzdialenosti a výšky meraného objektu na základe zadaných parametrov bol úspešne otestovaný na neočakávané vstupy a aj na výpočet správnych hodnôt tangensu pomocou zreťazených zlomkov a Taylorovho rádu. Program dodržuje požiadavky kladené na vstupné a výstupné dáta, je vytvorený podľa zadania. Odvodená presnosť 9 iterácii na dosiahnutie presnosti na 10 desatinných miesť je dostatočná a presná.

Musíme brať do úvahy, že veľkosť dátových typov sa môže meniť na rôznych architektúrach a tak isto sa tým bude meniť aj maximálna hodnota, ktorú môžeme uložiť do dátových typov. Tento poznatok môže spôsobiť nepresnosti pri výpočtoch Taylorovho rozvoja, hlavne pri posledných rádoch. Tak isto nám to môže spôsobovať problémy ak by sme chceli vypisovať s presnosťou na viac desatinných miest. Tieto problémy sa týkajú architektúr, kde *double* má menej ako 8bajtov.

Program splňuje štandardy C99 a je prenositeľný na rôzne platformy, pretože dodržuje štandardy a nepoužíva knižnice, ktoré sú viazané len k určitej platforme. Bol testovaný na operačných systémoch Microsoft Windows 7 x64 a GNU/Linux(Linux Mint).

Príloha A

Metriky kódu

Počet súborov:	1 súbor
Počet riadkov zdrojového súboru:	480
Veľkosť statických dát:	776B
Veľkosť spustiteľného súboru:	15238B (OS Linux, 64bitová architektúra, bez ladiacich informácií)