Dokumentácia k projektu pre predmety IZP a IUS

Iteračné výpočty

projekt č. 2

26. november 2014

Autor: Juraj Sokol,
xsokol08@stud.fit.vutbr.cz
Fakulta Informačních Technologií
Vysoké Učení Technické v Brně

Obsah

1 Úvod

V praxi sa bežne pri výpočtoch uhlov a vzdialeností používajú goniometrické funkcie sínus, kosínus, tangens, a kotangens. Aj keď ich človek berie ako samozrejmosť, počítač im nerozumie. Preto tu vzniká otázka, ako by počítač mohol vypočítať napr. tangens uhlu. Z pohľadu algoritmizácie to nie je ľahká úloha. Postupne vznikali rôzne algoritmy, no mali rozdielne vlastnosti ako efektívnosť a presnosť. Vzniká tu otázka, ktorý z nich je najlepší.

Tento dokument popisuje návrh a implementáciu aplikácie pre výpočet goniometrickej funkcie tangens a ako ukážku implementuje výpočet vzdialenosti od objektu a jeho výšku. Program funguje ako konzolová aplikácia, ktorá zo štandardného vstupu pri spustení prečíta dáta a z nich podľa zadaných prepínačov vykoná operáciu nad zadanými operandami.

Dokument sa skladá z nasledovných častí:

- V 2. kapitole sa rozoberá problém a definícia goniometrických funkcii
- V 3. kapitole sú navrhnuté všeobecné algoritmy na výpočet tangensu uhla v radiánoch.
- V 4. kapitole sú definované testy vychádzajúce zo záverov v predchádzajúcich kapitolách
- V 5. kapitole je rozoberaná konkrétna konečná implementácia

2 Analýza problému a princíp jeho riešenia

V tejto kapitole vysvetlím a popíšem goniometrickú funkciu tangens, ďalej navrhnem a popíšem jej výpočet z pohľadu počítača.

2.1 Zadanie problému

Cieľom projektu je implementácia výpočtu uhlu zadaného v radiánoch pomocou zreťazených zlomkov a Taylorovho polynómu. Program porovná a vypíše odchýlku medzi implementovanými algoritmami a funkciou tan() z matematickej knižnice jazyka C. Program navyše vypočíta vzdialenosť a výšku meraného objektu pomocou zreťazených zlomkov. Všetky údaje sú zadávané na štandardný vstup a výsledok je vypísaný na štandardný výstup.

2.2 Definícia goniometrických funkcií

Pod týmto pojmom rozumieme funkcie sínus, kosínus, tangens a kotangens.

2.2.1 Pravouhlý trojuholník

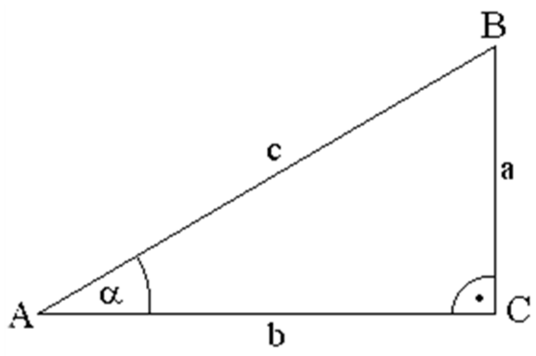
Pri tejto definícii sú jednotlivé prvky trojuholníka ABC nasledujúce:

- pravý uhol γ je pri vrchole C
- určovaným uhlom je uhol α pri vrchole A, vzhľadom k nemu je:
    - strana a označovaná protiľahlá odvesna
    - strana b označovaná priľahlá odvesna
    - najdlhšia strana c je nazývaná prepona trojuholníka
- súčet vnútorných uhlov trojuholníka je 180°

Sínus α je pomer dĺžky odvesny protiľahlej tomuto uhlu a dĺžky prepony; sin α = $\frac{a}{c}$.

Kosínus je pomer dĺžky odvesny priľahlej k tomuto uhlu a dĺžky prepony; cos α = $\frac{b}{c}$.

Tangens α je pomer dĺžok odvesny protiľahlej k tomuto uhlu a dĺžky odvesny k nemu priľahlej; tg α = $\frac{a}{b} = \frac{\sin\alpha}{\cos\alpha}$.



Obrázok 1: Ukážka pravouhlého trojuholníka a označenia jednotlivých strán, uhlov a vrcholov.

2.2.2 Rady

Za pomoci geometrie a vlastností limít je možné ukázať, že derivácia sínusu je kosínus a derivácia kosínusu je mínus sínus. Potom je možné pomocou Taylorových radov vyjadriť sínus, kosínus a tangens pre všetky komplexné čísla x takto:

$$\text{tg } x = x + \frac{x^3}{3} + \frac{2x^5}{15} + \frac{17x^7}{315} + \dots \text{ kde } \left(-\frac{\pi}{2} < x < \frac{\pi}{2}\right)$$

2.2.3 Zreťazené zlomky

Tangens možno vypočítať podľa nasledujúcich zlomkov:

$$\tan(z) = \cfrac{z}{1 - \cfrac{z^2}{3 - \cfrac{z^2}{5 - \cfrac{z^2}{7 - \cfrac{z^2}{9 - \cfrac{z^2}{11}}}}}} \quad /;\ \frac{z}{\pi} - \frac{1}{2} \notin \mathbb{Z}$$

Obrázok 3 Časť zreťazeného zlomku, kde z reprezentuje uhol v radiánoch

$$\tan(z) = \cfrac{1}{\frac{1}{z} - \cfrac{1}{\frac{3}{z} - \cfrac{1}{\frac{5}{z} - \cfrac{1}{\frac{7}{z} - \cfrac{1}{\frac{9}{z} - \cfrac{1}{\frac{11}{z} - \ldots}}}}}} \quad /;\ \frac{z}{\pi} - \frac{1}{2} \notin \mathbb{Z}.$$

Obrázok 2: Zlomok na obrázku č.3 prepísaný inak

Presnosť hodnoty závisí od počtu zanorených zlomkov. Ideálna hodnota by sa mala vypočítať nekonečným zreťazeným zlomkom. V praxi by sme sa ale nikdy nedopočítali žiadneho výsledku. Pri výpočte sa hodnota blíži k presnej hodnote. Ak je odchýlka - EPSILON predchádzajúceho a aktuálneho medzivýpočtu dostatočne malá, tak sa s touto hodnotou uspokojíme a ďalej nepočítame EPSILON >= $|Y_i - Y_{i-1}|$. Pri menších uhloch stačí pre vysokú presnosť už 5 zreťazených zlomkov, kým u väčších uhloch je treba viac zreťazených zlomkov.

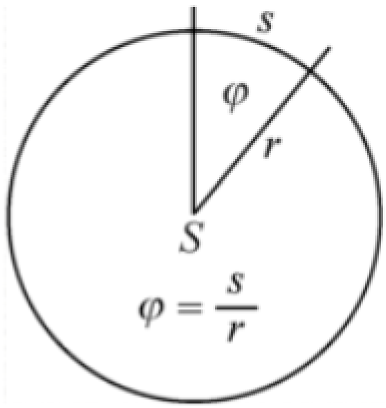
2.3 Merné jednotky uhlov

Uhly sa bežne merajú v stupňoch, no vo fyzike sa používa oblúková miera uhla, v ktorej jednotkou uhla je radián.

Uhol v oblúkovej miere sa vyjadrí ako podiel dĺžky oblúka a polomeru kružnice. Pre plný uhol (360° v stupňovej miere) platí:

Plný uhol (v stupňoch) = $\dfrac{\textit{dĺžka oblúku}}{\textit{polomer kružnice}} = \dfrac{s}{r}$ (rad)

Z toho vyplýva že 180° odpovedá π, preto 1 rad = $\dfrac{\pi}{180°}$.



Obrázok 4: Obrázok znázorňuje označenia pri výpočte radiánu uhla

2.4 Vhodný spôsob výpočtu tangensu

Výpočet tangensu z definície v pravouhlom trojuholníku sa javí ako najrýchlejší a najjednoduchší spôsob, no je potrebné vedieť dĺžku protiľahlej odvesny a priľahlej odvesny alebo sínus a kosínus daného uhla. Vypočítať tangens treba iba zadaním uhlu a žiadnych ďalších údajov. Preto tento spôsob nie je najvhodnejší.

Pre Taylorov polynóm je potrebný uhol a hodnoty, ktoré sa vyskytujú v jednotlivých zlomkoch. Keďže tieto hodnoty sú konštantné, takýto polynóm je vhodnejší ako predošlý spôsob výpočtu.

Zreťazené zlomky predstavujú najlepší spôsob, pretože pre výpočet je potrebný len uhol a nič viac, ostatné hodnoty sa dopočítajú počas výpočtu, pričom je možné dosiahnuť presnosť s takmer nulovou odchýlkou.

3 Návrh riešenia problému

Po analýze problému výpočtu tangensu som implementoval jeho výpočet pomocou zreťazených zlomkov a Taylorovho polynómu, keďže ako vstup program dostane len veľkosť uhlu v radiánoch. Uhol je možné zadať len v danom rozsahu 0 < uhol < 1,4.

3.1 Voľba premenných

V programe pre uhly, hodnoty tangensu a výsledky výpočtov je vhodné použiť dátový typ double, keďže hodnoty sú desatinné čísla a tento typ má aj dostatočný rozsah ($2{,}3*10^{-308}$ až $1{,}7*10^{308}$).

3.2 Výpočet tangensu

Funkcie na výpočet tangensu uhla, ktoré podľa zadania majú byť implementované dvakrát a to pomocou zreťazených zlomkov - cfar_tan() a Taylorovho polynómu – taylor_tan(). Obe funkcie majú rovnaké argumenty:

- double x -> veľkosť uhla v radiánoch
- unsigned int n -> počet iterácii

Obe funkcie vrátia tangens zadaného uhla typu double.

3.2.1 Zreťazené zlomky

Tangens sa počíta podľa definície v kapitole 2.2.3 znázorneného na obrázku č. 2. Algoritmus som prispôsobil na konečný nasledovne:

- Vypočíta sa čitateľ najzanorenejšieho zlomku tak, že sa počet iterácii vynásobí 2 a toto číslo sa dekrementuje, keďže má byť nepárne.
- Algoritmus pred vstupom do iteračného cyklu vydelí čitateľ uhlom a podiel sa použije vo vyššom zlomku, od čitateľa sa odčíta 2.
- Následne začne iteračný cyklus, kde sa číslo vypočítané v predošlom bode odčíta od podielu čitateľa a uhlu, od čitateľa sa odčíta 2. Toto číslo sa použije vo vyššom zlomku. Tento cyklus sa opakuje pokiaľ je čitateľ väčší od 0.
- Po ukončení cyklu algoritmus vráti hodnotu tangensu zadaného uhla.

3.2.2 Taylorov polynóm

Tangens sa počíta podľa definície v kapitole 2.2.2. Algoritmus som implementoval na 14 iterácii. Čísla, ktoré sú v čitateli a v menovateli som uložil do polí.

- Na začiatku algoritmu sa výsledok inicializuje na nulu.
- Algoritmus vstúpi do cyklu, kde vypočíta hodnotu aktuálneho zlomku a výsledok pripočíta k výsledku. Hodnoty, ktorými má vynásobiť a vydeliť uhol zoberie z polí týchto hodnôt.
- Po ukončení cyklu vráti hodnotu tangensu daného uhla.

3.3 Problém presnosti výpočtu

Keďže algoritmy sú implementované ako konečné, vzniká tu odchýlka pri výpočte od skutočnej hodnoty. Pri Taylorovom polynóme sa odchýlka veľmi korigovať nedá, keďže je implementovaný na maximálne 14 iterácii.

Pri zreťazených zlomkoch sa dá odchýlka korigovať zvýšením počtu iterácii. Experimentálne som zistil, že pri malých uhloch stačí 5 (0 – 0,5) iterácii a odchýlka je menšia než na 6 desatinných miest. Všeobecne platí, že pri 10-tej iterácii je odchýlka menšia než na 10^{-16} desatinných miest.

3.4 Meranie

Program vypočíta vzdialenosť a výšku meraného objektu od meracieho prístroja z údajov zadaných na štandardný vstup. Výsledok je vypísaný na štandardný výstup. Tangens uhla sa vypočíta pomocou zreťazených zlomkov.

3.4.1 Vzdialenosť objektu

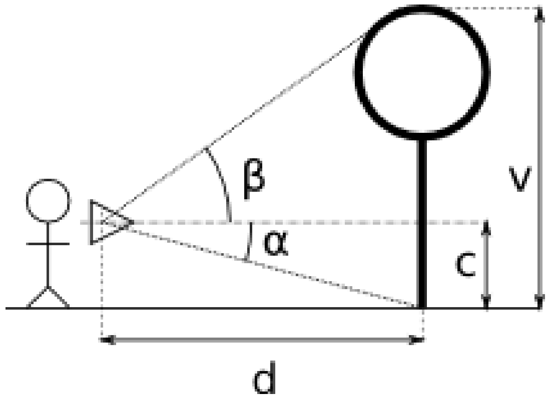
Funkcia vypočíta vzdialenosť objektu na základe nasledujúceho vzorca:

$$d = \frac{c}{\tan \alpha}$$

3.4.2 Výška objektu

Výška objektu sa vypočíta podľa vzorca:

$$v = c + \tan \beta * d$$



3.5 Analýza vstupných dát

V zadaní je presne špecifikovaný formát vstupných dát. Zadané sú tieto prepínače:

--help -> program vypíše nápovedu používania programu a skončí.

--tan –> porovná presnosti výpočtu tangens uhlu A (v radiánoch) medzi volaním tan z matematickej knižnice, a výpočtu tangens pomocou Taylorovho polynómu a zreťazeného zlomku. Argumenty N a M udávajú, v ktorých iteráciách iteračného výpočtu má porovnanie prebiehať. 0 < N <= M < 14

-m -> vypočíta a zmeria vzdialenosti. Uhol α (viď. obrázok) je daný argumentom A v radiánoch. Program vypočíta a vypíše vzdialenosť meraného objektu. 0 < A <= 1.4 < π/2. Pokiaľ je zadaný, uhol β udáva argument B v radiánoch. Program vypočíta a vypíše i výšku meraného objektu. 0 < B <= 1.4 < π/2

-c -> nastavuje výšku meracieho prístroja pre výpočet. Výška c je daná argumentom X (0 < X <= 100). Argument je voliteľný - implicitná výška je 1.5 metrov.

4 Špecifikácia testov

Aby program fungoval správne, treba otestovať jeho správanie v rôznych situáciách.

Test 1: Chybná syntax -> detekcia chyby

-c –m 1.0

-tan 1.1

Test 2: Nezmyselné dáta -> detekcia chyby

-x 0.5c

Test 3: Argumenty mimo povolený rozsah hodnôt -> detekcia chyby.

-m -0.5

-c 101 –m 0.2 10.2

Test 4: Správnosť výpočtu -> predpokladaná správna hodnota

vstup	očakávaný výstup
--tan 0 1 1	1 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
--tan 1.024 6 10	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
-m 0.3 0.9	4.8490922156e+00 7.6106234032e+00
-m 0.3 0.9	4.8490922156e+00 7.6106234032e+00

Poznámka: Výsledné hodnoty nemusia odpovedať daným hodnotám. Výsledok závisí od implementácie a optimalizácie.

5 Popis môjho riešenia

Pri programovaní a implemetácii funkcii som vychádzal zo záverov popísaných v predošlých kapitolách.

5.1 Ovládanie programu

Program sa ovláda výhradne z príkazového riadku pomocou prepínačov popísaných vyššie.

Po spustení program očakáva argumenty na štandardnom vstupe. Ak sú argumenty zadané nesprávne program vypíše chybu a to aj vtedy ak sú hodnoty mimo rozsah. Argumenty spracovávam po jednom funkciou strcmp(), čo zistí čo bude vykonávať a aké nasledujúce argumenty očakáva. Číselná hodnoty spracovávam funkciou strtod().

5.2 Vlastná implementácia

Pri spustení programu je volaná funkcia main(). Nachádza sa tu jediný príkaz a to volanie funkcie spracuj_argumenty(), ktorá spracuje jednotlivé argumenty príkazového riadku a podľa nich volá dalšie funkcie. Ako argumenty sú počet argumentov pri spustení a pole ukazateľov na jednotlivé argumenty. Po skončení vráti hodnotu 0.

Ak funkcia zavolá porovnanie_vypoctu(), tak program porovná a vypíše odchýlku vo výpočte tangensu medzi funkciou z math.h, cfrac_tan() a taylor_tan() vo formáte printf("%d %e %e %e %e %e\n), kde (číslo iterácie, tan(), taylor_tan(), tan() – taylor_tan(), cfrac_tan(), tan() – cfrac_tan()).

Ak zavolá funkciu vzdialenost() tak program vypočíta a vypíše vzdialenosť objektu, ak zavolá vyska() tak vypočíta a vypíše aj výšku meraného objektu. Tangens sa vypočíta v oboch prípadoch pomocou cfrac_tan(). Môže byť zmenená aj výška meracieho prístroja, no jeho implicitná výška je 1.5.

6 Záver

Program spĺňa všetky zadané požiadavky projektu a je úspešne otestovaný všetkými navrhnutými testovacími hodnotami.

Program počíta s uhlami v rozsahu 0 až π/2, pretože uhly od π/2 do π sú záporné a v π/2 nemá definovanú hodnotu. Tento nedostatok by sa dal odstrániť pridaním funkcie, ktorá by uhol konvertovala do prvého kvadrantu (0 - π/2) a zmenila by znamienko výsledku, no zadanie takúto funkciu nevyžadovalo.

Referencie

Tangent: Introduction to the Tangent Function. Wolfram research [online]. [cit. 2014-11-30]. Dostupné z: http://functions.wolfram.com/ElementaryFunctions/Tan/introductions/Tan/ShowAll.html

[SPRACOVALI PAVEL ČERMÁK, Petra Červinková a Z českého originálu ... preložil Marián HANULA].Zmaturuj! z matematiky. Bratislava: Didaktis, 2003. ISBN 80-891-6001-8.

A Metriky kódu

Počet súborov: 1 súbor

Počet riadkov zdrojového súboru: 217 riadkov

Veľkosť statických dát: 6 992 bajtov

Veľkosť spustiteľného súboru: 13 384 bajtov (systém Linux, 64 bitová architektúra, kompilovaný s gcc -std=c99 -Wall -Wextra -pedantic proj2.c -lm -o proj2)