Dokumentace k projektu pro předměty IZP A IUS

Iterační výpočty

projekt č. 2

1. 12. 2014

Autor: Jan Pavlica, xpavli78@stud.fit.vutbr.cz

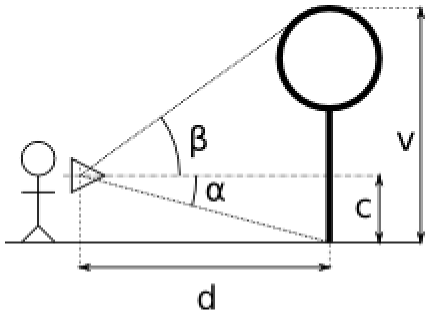
Fakulta Informačních Technologií

Vysoké Učení Technické v Brně

Obsah

1 Úvod

Dokumentace k druhému projektu do předmětů IZP - základy programování, a IUS - úvod do softwarového inženýrství, vyučovaným na VUT v Brně, fakultě informačních technologií. Navržený program funguje jako konzolová aplikace, která podle zadaných parametrů provede výpočet vzdálenosti a případné výšky pozorovaného objektu. Na vstupu je očekáván úhel α a případná výška (v) ve které je objekt pozorován. Pro výpočet výšky objektu je nutné zadat další úhel β.



Obr. 1 - nákres zadání

Dokument je rozdělen do několika částí. V kapitole **2** se zabývám analýzou a rozborem problémů spojených s výpočtem. V kapitole **3** popisuji samotný návrh řešení problému. Kapitola **4** je věnována testování správnosti jednotlivých funkcí programu. Implementace v kapitole **5** vyplývá s analýzy a návrhu řešení.

2 Analýza problému a princip jeho řešení

2.1 Zadání problému

Bylo požadováno vytvoření programu v jazyce C, který vypočítá hodnotu tangens, jehož úhel bude zadán v radiánech a to v intervalu $\langle 0;1,4 \rangle$. Tangens byl počítán pomocí **Taylorova polynomu a zřetězených zlomků** v maximálním počtu 13 iterací. Tyto hodnoty měly být porovnány s hodnotou vypočítanou funkcí *tan* a vypočítána absolutní odchylka. Uživatel zadá pomocí argumentů velikost úhlu a dále vymezí počet iterací, ve kterých má být tangens vypočítán.

Program měl být dále schopen vypočítat vzdálenost a případně i výšku měřeného objektu pozorovaného v určité výšce pomocí hodnoty tangens vypočítané zřetězenými zlomky s přesností na 10 desetinných míst. K dosažení této přesnosti bylo nutné odvodit počet iterací.

V programu bylo zakázáno použití jakékoliv funkce z knihovny *<math.h>* vyjímaje funkci *tan* použitou pouze pro srovnání výpočtů, funkce *isnan* a *isinf* a konstanty *NAN* a *INF*.

2.2 Taylorův polynom pro tangens

Taylorův polynom aproximuje hodnoty funkce, která má v daném bodě derivaci, pomocí polynomu, jehož koeficienty závisí na derivacích funkce v tomto bodě.

$$\operatorname{tg} x = x + \frac{1}{3}x^3 + \frac{2}{15}x^5 + \frac{17}{315}x^7 + \cdots \quad \text{pro } x \in (-\frac{\pi}{2}, \frac{\pi}{2})$$

Obr. 2 - Taylorův polynom pro tangens

Jelikož je způsob odvození jednotlivých členů v čitateli a jmenovateli velice složité, bylo možné použít 13 prvních členů již vypočítaných. Vzhledem k tomu, že proměnná X mohla být pouze z intervalu (0;1,4⟩, byla podmínka uvedená pro X vždy splněna.

2.3 Zřetězené zlomky

Příklad zřetězeného zlomku pro výpočet hodnoty tangens můžeme vidět na obrázku.

$$\tan(x) = \cfrac{1}{\cfrac{1}{x} - \cfrac{1}{\cfrac{3}{x} - \cfrac{1}{\cfrac{5}{x} - \cfrac{1}{\cfrac{7}{x} - \cdots}}}}$$

Obr. 3 - zřetězené zlomky pro tangens

Problémem je, že k výpočtu členu a_n je nutné vypočítat člen a_{n+1}. Proto je nutné určit si pevný počet iterací a vypočítat nejprve nejvyšší člen. Omezením maximálního počtu iterací na 13 dostáváme konečný zřetězený zlomek, kde nejvyšší člen může nabývat hodnoty $\frac{x}{25}$.

2.4 Určení počtu iterací pro dosažení zadané přesnosti

Pro výpočet vzdálenosti a výšky objektu bylo nutné dosáhnout přesnosti na 10 desetinných míst. Je však nezbytné počítat s vzniknutím dalších nepřesností při provádění následného násobení a dělení. Nestačí proto dosáhnout absolutní odchylky s přesností na 10 desetinných míst nýbrž s přesností na 13 desetinných míst. Tím by mělo zůstat prvních 10 desetinných míst ve výsledku nepozměněných.

3 Návrh řešení problému

3.1 Výpočet tangens pomocí Taylorova polynomu

Výpočet hodnoty tangens pomocí Taylorova polynomu jsem se rozhodl řešit pomocí cyklu. Vzhledem k zákazu užívání funkcí z knihovny <*math*.h> bylo nutné vytvořit uvnitř samotné funkce proměnnou počítající mocninu. Dále bylo nutné použít datový typ *long long* pro uložení jednotlivých členů kvůli jejich velikosti. Pro výpočet jednotlivých členů (A_n) jsem použil následující vzorec, kde n určuje, o kolikátou iteraci jde. K výsledku jsem došel sečtením všech vypočítaných členů.

$$A_n = \frac{n - tý\ čitatel\ Taylorovy\ řady}{n - tý\ jmenovatel\ Taylorovy\ řady} * X^{(2*n)-1}$$

3.2 Výpočet tangens pomocí zřetězených zlomků

Při výpočtu pomocí zřetězených zlomků jsem opět zvolil počítání pomocí cyklu. Bylo zde však nutné začít výpočtem posledního členu (A_n) zřetězeného zlomku, který se dal vypočítat jednoduchým vztahem, kde n určuje, o kolikátou iteraci jde.

$$A_n = \frac{(n*2)-1}{X}$$

Výpočet dalších členů iterace již probíhal způsobem, při kterém využíval předchozí vypočtený člen:

$$A_{n-1} = \frac{(n*2)-1}{X} - \frac{1}{A_n}$$

Na závěr je nezbytné provést výpočet první iterace:

$$\frac{1}{A_1}$$

3.3 Určení intervalů, ve kterých dosahujeme požadované přesnosti

K určení přesnosti jsem porovnával výsledek mé funkce počítající pomocí zřetězených zlomků a funkce *tan()* z knihovny *<math.h>*. Porovnávání probíhalo v cyklu s úhlem zadaným s přesností na 10 desetinným míst. V případě, že absolutní odchylka byla větší než 1E-13 byla vypsána hodnota a cyklus pokračoval s vyšším počtem iterací. Výsledkem byly intervaly v tabulce.

Počet iterací:	***Interval:***
1	⟨0; 0,00006695000⟩
2	⟨0,00006695000; 0,00537825000⟩
3	⟨0,00537825000; 0,03976726000⟩
4	⟨0,03976726000; 0,12882882997⟩
5	⟨0,12882882997; 0,28243970989⟩
6	⟨0,28243970989; 0,49621669978⟩
7	⟨0,49621669978; 0,75592270106⟩
8	⟨0,75592270106; 1,03872845205⟩
9	⟨1,03872845205; 1,30776122042⟩
10	⟨1,30776122042; 1,4⟩

Tabulka 1 - počet iterací v zadaných intervalech

4 Testování programu

Test jsem zaměřil na správné zadání parametrů, zadání hodnot v zadaném rozsahu a také přesnost výsledků.

4.1 Chybná syntaxe

Tyto testy měly za úkol ověřit správnost zadaných parametrů.

./proj2	Chybný počet argumentů
./proj2 --tan 1.2 1 13	Správně --tan
./proj2 --tan -1.5 1 13	Chybná velikost úhlu v radiánech

./proj2 sadssa	Chybný 1. parametr příkazové řádky
./proj2 -c -6 -m 0.5 0.13	Chybná výška
./proj2 -m 0.7	Správně -m
./proj2 --tan 1 -5 8	Chybný dolní parametr rozsahu

4.2 Správnost výsledku

4.2.1 Funkce --tan A N M

./proj2 --tan 1.024 6 10

Očekávaný 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

Můj 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

4.2.2 Funkce [-c X] -m A

	Očekávaný výstup:	Můj výstup:
./proj2 -m 0.15	9.9248872584e+00	9.9248872584e+00
./proj2 -m 0.75	1.6101392228e+00	1.6101392228e+00
./proj2 -c 7 -m 0.9	5.5548580349e+00	5.5548580349e+00

4.2.3 Funkce [-c X] -m A [B]

	Očekávaný výstup:	Můj výstup:
./proj2 -m 0.7 1.3	1.7808627482e+00	1.7808627482e+00
	7.9148500648e+00	7.9148500648e+00
./proj2 -c 1.15 -m 0.7 1.3	1.3653281069e+00	1.3653281069e+00
	6.0680517163e+00	6.0680517163e+00
./proj2 -c 7 -m 0.9 0.45	5.5548580349e+00	5.5548580349e+00

9.6833023125e+00	9.6833023125e+00

5 Vlastní implementace

Ihned po spuštění programu je zavolána funkce *control*, která otestuje správnost zadaných parametrů. Dále je pak zavolána funkce *call_f*, která zpracuje vstupní parametry a předá je požadované funkci podle zadaného argumentu (--help, --tan, -m).

Funkce *print_tan* osahuje cyklus pro tisk hodnot vypočtených funkcemi *taylor_tan, cfrac_tan* a jejich absolutních odchylek oproti hodnotě vypočtené funkcí *tan* při jednotlivých iteracích. V této funkci je také použita funkce *abso*¸ která vrací absolutní hodnotu daného čísla.

Funkce *taylor_tan* obsahuje konečný cyklus, kde je počet iterací zadán uživatelem. V cyklu je počítán aktuální člen podle vzorce v kapitole **3.1** a ten je přičítán k celkovému součtu všech předchozích členů. V této funkci je také uloženo prvních 13 čitatelů a jmenovatelů Taylorova polynomu pro tangens. Při výpočtech používám proměnnou *pow*_x, ve které počítám aktuální mocninu zadaného čísla.

Funkce *cfrac_tan* vypočítá nejprve poslední člen zadané iterace, od kterého jsou pak dále počítány ostatní členy. Používám k tomu vzorec v kapitole **3.2**. Počet iterací je buďto zadán uživatelem nebo odvozen pomocí funkce *iterations*, která pracuje s tabulkou () odvozenou v kapitole **3.3**

Výška a vzdálenost jsou počítány funkcemi *distance* a *high*, které užívají hodnoty vypočítanou pomocí funkce *cfrac_tan*

6 Závěr

Program úspěšně počítá hodnotu tangens bez použití funkce z knihovny <math.h>. Veškeré výsledky mých funkcí byly porovnány s výsledky vypočítané pomocí funkce *tan*, oba výsledky jsou s ohledem na požadovanou přesnost stejné.

Program samotný však postrádá smysl, jelikož při použití zmiňované funkce *tan* by bylo řešení nejen rychlejší nýbrž i přesnější. Aby se zvýšila přesnost výpočtu u Taylorova polynomu, muselo by být použito o mnoho více členů, které jsou však limitovány velikostí datových typů.

Výsledná aplikace byla testována v operačním systému GNU / Linux, kde fungovala bez problému a vypisovala odpovídající výsledky.

A Metriky kódu

Počet souborů: 1

Počet řádků zdrojového textu: 346

Funkcí: 13

Statická data: 636B

Velikost programu: 17461B