

# ÚVOD DO SOFTWAREVÉHO INŽENÝRSTVÍ 2014/2015

## Projekt č. 2 – Dokumentace k projektu z IZP **Iterační výpočty**

29. listopadu 2014

Autor: Ondřej Valeš, [xvales03@stud.fit.vutbr.cz](mailto:xvales03@stud.fit.vutbr.cz)  
Fakulta Informačních Technologii  
Vysoké Učení Technické v Brně

# Obsah

1 Úvod.....	1
2 Analýza problému a princip jeho řešení.....	1
2.1 Zadání problému .....	1
2.2 Výpočet pomocí Taylorova polynomu.....	1
2.3 Výpočet pomocí zřetěžených zlomků .....	2
2.4 Určení počtu iterací pro dosažení požadované přesnosti .....	2
2.5 Výpočet vzdálenosti a výšky.....	3
3 Návrh řešení .....	3
3.1 Výpočet vzdálenosti a výšky.....	3
3.2 Forma vstupních dat.....	3
4 Specifikace testů.....	3
5 Popis implementovaného řešení.....	5
5.1 Ovládání programu.....	5
5.2 Implementace programu.....	5
6 Závěr .....	6
A Reference.....	7
B Metriky kódu .....	7

# 1 Úvod

Využití Taylorových polynomů a zřetězených zlomků pro výpočet hodnot goniometrických funkcí představuje z pohledu algoritmizace metodu, kterou lze implementovat pomocí iteračních výpočtů. Takovýto výpočet lze provést s využitím pouze základních matematických operací  $+$ ,  $-$ ,  $*$  a  $/$ . Vypočtené hodnoty goniometrické funkce tangens jsou využity k řešení úlohy o trojúhelníku. V tomto konkrétním případě k určení vzdálenosti a výšky měřeného objektu.

Tento dokument popisuje návrh a implementaci aplikace pro výpočet vzdálenosti objektu od měřicího přístroje a jeho výšky na základě předem naměřených hodnot úhlů. Aplikaci lze také využít pro srovnání přesnosti různých metod výpočtu tangens. Navržený program je konzolová aplikace spouštěná s argumenty.

## 2 Analýza problému a princip jeho řešení

Funkci tangens lze implementovat pomocí Taylorova polynomu a zřetězených zlomků. V obou případech vznikají při výpočtech nepřesnosti. V této kapitole jsou popsány výhody a nevýhody obou metod a je určen počet iterací výpočtu potřebný k dosažení určité přesnosti.

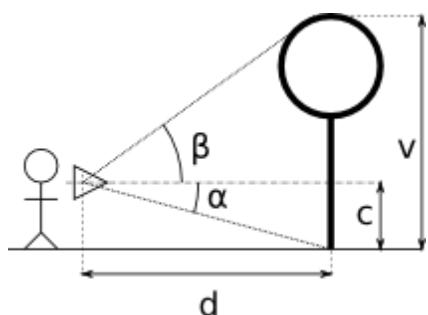
### 2.1 Zadání problému

Cílem projektu je vytvoření programu v jazyce C, který vypočte vzdálenost ( $d$ ) a výšku ( $v$ ) měřeného objektu pomocí naměřených úhlů ( $\alpha$ ,  $\beta$ ) a výšky, ve které je umístěn měřicí přístroj ( $c$ ). Výpočet hodnoty funkce tangens má být proveden metodou zřetězených zlomků. Využití matematických operací je omezeno na operace  $+$ ,  $-$ ,  $*$ ,  $/$ .

Naměřené hodnoty jsou programu předány v podobě argumentů. Výsledky program vypisuje na standardní výstup ve vědeckém formátu.

V projektu je také nutno zohlednit nepřesnosti vznikající při výpočtech. Výsledky mají být přesné na 11 platných číslic.

Součástí zadání je i implementace funkce srovnávající přesnost výpočtu tangens pomocí výše uvedených metod. Maximální počet iterací, pro které je srovnání možné provádět, je omezen na 13.



### 2.2 Výpočet pomocí Taylorova polynomu

Jednotlivé členy Taylorova polynomu představují členy nekonečné posloupnosti. Hodnotu tangens určíme sečtením všech těchto členů, což nelze provést. Přibližnou hodnotu lze určit sečtením prvních několika členů, čím více členů, tím přesnější hodnota tangens.

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

Každý člen je  $x^2$  krát větší než předchozí, výpočet koeficientů jednotlivých členů přesahuje rámec projektu. Tyto hodnoty jsou pro prvních 13 členů uloženy jako konstanty v programu, proto nelze provést více než 13 iterací.

Hodnota prvního členu se uloží do výsledku. Při každé iteraci se vypočítá hodnota dalšího členu a přičte se k výsledku.

Výhodou výpočtu pomocí Taylorova polynomu je možnost již vypočítaný výsledek nadále zpřesňovat přičítáním dalších členů dokud není dosaženo požadované přesnosti, což u výpočtu pomocí zřetězených zlomků není možné. Hlavní nevýhodou je omezený maximální počet iterací.

## 2.3 Výpočet pomocí zřetězených zlomků

Zřetěžený zlomek má ve jmenovateli výraz obsahující další členy zřetěženého zlomku. Proto při jeho výpočtu potřebuji dopředu vědět, s kolika členy celý výpočet proběhne. Výsledek celého výrazu je hledaná hodnota tangens.

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

Čitatel je v tomto zřetěženém zlomku konstantní, ve jmenovateli se nachází výrazy typu  $b/x$ , kde  $b$  jsou lichá čísla začínající od 1.

Výpočet probíhá od nejvíce zanořeného členu. Mezivýsledek se ukládá pro použití při další iteraci výpočtu. V každé iteraci se od jmenovatele odečte mezivýsledek a vypočte se hodnota zlomku, která se uloží jako mezivýsledek.

Výhodou této metody je jednoduché určení vztahu mezi jednotlivými koeficienty, což umožňuje provést libovolný počet iterací a dosáhnout vysoké přesnosti. Pokud přesnost není dostatečná, nelze spočítanou hodnotu nijak upravit, ale je nutno celý výpočet provést znovu s větším počtem iterací.

## 2.4 Určení počtu iterací pro dosažení požadované přesnosti

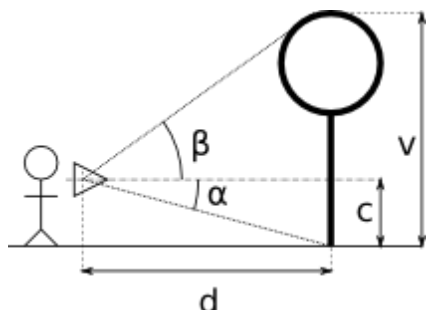
Počet iterací byl určen experimentálně v jiném programu. Pro všechny úhly počínaje 0 zvětšující se vždy o 0,000001 až po 1,55 (kvůli zjištění počtu iterací pro hraniční hodnotu 1,4 byla vybrána horní hranice pro ověřování vyšší) byla vždy určena hodnota tangens metodou zřetězených zlomků a porovnána s hodnotou vypočítanou funkcí `tan()`. Když se prvních 11 platných číslic neshodovalo, byl proveden výpočet znovu s o jedna větším počtem iterací. Pokud se počet potřebných iterací pro stávající a předešlý úhel lišil, program vypsal počet potřebných iterací a hodnotu tohoto úhlu. Výsledný výpis:

1	0.000000
2	0.005379
3	0.055247
4	0.214282
5	0.425555
6	0.695765
7	1.121962
8	1.387270
9	1.534294

Na základě tohoto výpisu byl počet potřebných iterací stanoven na 8. Devět iterací bylo nutno až pro úhel, který je mimo interval, se kterým pracuje tento program.

## 2.5 Výpočet vzdálenosti a výšky

Vzdálenost  $d$  se vypočítá:  $d = c / \tan(\alpha)$ . Výška  $v$  se vypočítá:  $v = d * \tan(\beta) + c$ .



## 3 Návrh řešení

Již v zadání bylo určeno rozmezí vstupních hodnot a metoda výpočtu tangens. Proto byla pro výpočet tangens použita metoda zřetězených zlomků.

Analýza pouze potvrdila, že tato metoda je výhodnější. Při stejném počtu iterací byl výsledek přesnější než při použití Taylorova polynomu. Navíc jsou jednotlivé koeficienty rovnou počítány a nemusí být nikde uloženy. Výpočet pomocí Taylorova polynomu nedosahoval požadované přesnosti ani pro maximální možný počet iterací.

### 3.1 Výpočet vzdálenosti a výšky

Výpočty se provádí podle vzorců uvedených v kapitole 2.5. Nejprve se spočítá vzdálenost měřeného objektu, která je následně použita při výpočtu výšky objektu. Pokud jsou dodržena omezení předepsaná v zadání, nemůže při výpočtech dojít k chybě dělení nulou.

### 3.2 Forma vstupních dat

Vstupní data jsou zadávána jako parametry funkce, jejich analýzu provádí funkce `strtoi` a `strtod`. Chyby v rozsahu vstupních dat jsou ověřeny přímo před výpočtem (např. hodnota úhlu mimo interval nebo záporný počet iterací).

## 4 Specifikace testů

Ze zadání vyplývá několik možných chybových stavů, které je nutno ošetřit: chybně zadané parametry, číselný údaj obsahující jiné znaky než číslice, chybný výběr rozsahu srovnání, úhel mimo povolený interval, příliš velká nepřesnost vzniklá při výpočtech.

**Test 1:** Chybně zadané parametry

Očekávaný výstup: chybové hlášení.

```
./proj2
./proj2 --blabla
./proj2 --help 42
./proj2 -c 42.42 --tan 1.2 5 6
```

```
./proj2 -c 42.42
./proj2 -c 200 -m 1.1
./proj2 -c -1.1 -m 1.1
./proj2 -m 1.2 1.2 1.3
```

Výsledek testování: program reagoval podle očekávání.

**Test 2:** Číselný údaj obsahující jiné znaky než číslice

Očekávaný výstup: chybové hlášení.

```
./proj2 --tan 0.5 1a 2b
./proj2 --tan 1.2t4 5 8
./proj2 -m asdf
./proj2 -c 10 -m 1..1
./proj2 -c 5.1 -m 0.*
./proj2 -c +-10 -m 0.5
```

Výsledek testování: program reagoval podle očekávání.

**Test 3:** Úhel mimo povolený interval

Očekávaný výstup: chybové hlášení.

```
./proj2 -m 0
./proj2 -m 1.5
./proj2 -m -.1
./proj2 -m 1 1.41
./proj2 -c 10 -m 2
./proj2 -c 10 -m 1 2
```

Výsledek testování: program reagoval podle očekávání.

**Test 4:** Výpis nápovědy

Očekávaný výstup: vypsání nápovědy.

```
./proj2 --help
```

Výsledek testování: program vypsál nápovědu.

**Test 5:** Správnost výpočtu

Vstup	očekávaný výstup	
./proj2 -m 0.3	4.8490922156e+00	
./proj2 -m 1	9.6313892390e-01	
./proj2 -c 100 -m 1.024	6.0870591173e+01	
./proj2 -m 0.3 0.1	4.8490922156e+00	1.9865320774e+00
./proj2 -m 1.4 1.4	2.5871508875e-01	3.0000000000e+00
./proj2 -c 5.1 -m 0.23 1.1	2.1781527133e+01	4.7895465784e+01

Výsledek testování: program reagoval podle očekávání.

## 5 Popis implementovaného řešení

Při implementaci byl použit výpočet tangens pomocí zřetězených zlomků, protože k dosažení požadované přesnosti stačí menší počet iterací než u jiných metod. Pro funkci srovnávající přesnost byl implementován i výpočet pomocí Taylorova polynomu.

### 5.1 Ovládání programu

Program je konzolová aplikace s textovým ovládáním. Veškeré vstupní hodnoty jsou zadávány jako parametry spouštěného programu.

Při spuštění s parametrem `--help` se na standardní výstup vypíše nápověda.

Při spuštění s parametry `--tan X N M` se na standardní výstup vypíše srovnání výpočtu tangens pomocí volání funkce `tan()` z knihovny `math.h`, Taylorova polynomu a metody zřetězených zlomků. Program za parametrem `--tan` očekává úhel v radiánech (reálné číslo) a dolní a horní hranici počtu iterací (přirozená čísla menší než 14, dolní hranice nesmí být větší než horní) pro které má srovnání proběhnout. Na standardní výstup jsou pro každý počet iterací ze zadaného intervalu na řádek vypsaný: počet iterací, hodnota vypočítaná funkcí z matematické knihovny, hodnota vypočítaná pomocí Taylorova polynomu, absolutní odchylka mezi funkcí z matematické knihovny a Taylorovým polynomem, hodnota vypočítaná metodou zřetězených zlomků a absolutní odchylka mezi funkcí z matematické knihovny a metodou zřetězených zlomků (krom prvního vše ve vědeckém formátu).

Při spuštění s parametry `[-c X]-m A [B]` program počítá vzdálenost a výšku měřeného objektu. Závorky `[]` značí nepovinné parametry. Parametr `-c` nastavuje výšku, ve které se nachází měřicí přístroj. Za tímto parametrem program očekává zadanou výšku (reálné číslo,  $0 < X \leq 100$ ), pokud není zadán, je hodnota nastavena na 1,5. Parametr `A` je hodnota úhlu  $\alpha$  v radiánech (reálné číslo,  $0 < A \leq 1,4$ ). Parametr `B` je hodnota úhlu  $\beta$  v radiánech (reálné číslo,  $0 < B \leq 1,4$ ). Pokud je zadán pouze parametr `A`, program vypíše na standardní výstup vzdálenost měřeného objektu od přístroje (ve vědeckém formátu zobrazujícím 10 desetinných míst). Pokud je zadán i parametr `B`, program vypíše také výšku měřeného objektu (na samostatný řádek, ve vědeckém formátu zobrazujícím 10 desetinných míst).

Pokud je program spuštěn s jinými parametry, vypíše chybové hlášení na chybový výstup.

### 5.2 Implementace programu

Pro zpracování parametrů jsou použity funkce `process_params` a `check_params` spouštěné z hlavního programu které vrací kód příslušné operace a ověřují platnost zadaných parametrů. Nastavení výšky obstarává funkce `set_height`, funkci je předán odkaz na proměnnou v hlavním programu, kam se uloží výsledek. Na základě zpracovaných argumentů je spuštěna jedna z následujících funkcí: `print_help`, `tan_compare`, `calc_both`. Funkcím jsou předány hodnoty zpracovaných parametrů.

Funkce `print_help` vypíše nápovědu k programu.

Funkce `tan_compare` provádí srovnání přesnosti výpočtu. Ověřuje rozsah, ve kterém má srovnání proběhnout, a vrací chybu, pokud je rozsah neplatný. Volá funkce `cfrac_tan` a `taylor_tan` a vypisuje výsledky srovnání (viz kapitola 4.1). Všechny parametry jsou předávány hodnotou.

Funkce `calc_both` provádí výpočet vzdálenosti a výšky objektu. Ověřuje zadané hodnoty úhlů, pokud jsou mimo vymezený rozsah, vrací chybu. Při výpočtu tangens volá funkci `cfrac_tan` a vypisuje vzdálenost a výšku měřeného objektu. Všechny parametry jsou předávány hodnotou.

Funkce `taylor_tan` provádí výpočet tangens pomocí Taylorova polynomu (viz. kapitola 2.2). Vrací vypočítanou hodnotu.

Funkce `cfrac_tan` provádí výpočet tangens pomocí zřetězeného zlomku (viz kapitola 2.3). Vrací vypočítanou hodnotu.

Funkce `my_abs` provádí výpočet absolutní hodnoty. Vrací vypočítanou hodnotu.

Ošetření chybových stavů provádí funkce `print_err` volaná s kódem chyby.

Všechny výpočty jsou ukládány do datového typu `double`, jak požaduje zadání. Pro uložení zadaných počtů iterací jsem použil datový typ `unsigned int`, protože se jedná o přirozená čísla.

## 6 Závěr

Program počítá s úhly z intervalu  $(0; 1,4)$  a výškou měřicího přístroje z intervalu  $(0; 100)$ , výsledky jsou přesné na prvních 11 platných číslic. Tyto omezení je možno odstranit v další verzi programu přepočítáním počtu iterací potřebných pro dosažení dané přesnosti pro úhly mimo zadaný interval. Stejným způsobem lze upravit požadovanou přesnost.

Další omezení vyplývá z uložení hodnot koeficientů Taylorova polynomu. Implementací funkce vypočítávající tyto koeficienty by toto omezení bylo odstraněno a bylo by možné srovnávat přesnost výpočtů v neomezeném intervalu počtu iterací.

Program byl otestován v prostředí operačních systémů Windows 8.1 a GNU 8.4 s navrženými testovacími hodnotami.



## A Reference

A156769 - OEIS: A 'look-a-like' of the denominators in Taylor series for  $\tan(x)$ . [online]. [cit. 2014-12-05]. Dostupné z: <https://oeis.org/A156769>

A002430: Numerators in Taylor series for  $\tan(x)$ . [online]. [cit. 2014-12-05]. Dostupné z: <https://oeis.org/A002430>

POLÁK, Josef. *Přehled středoškolské matematiky*. 8. vyd. Praha: Prometheus, 2005, 608 s. ISBN 80-719-6267-8.

## B Metriky kódu

<b>Počet Souborů:</b>	1 soubor
<b>Počet řádků zdrojového textu:</b>	289 řádků
<b>Velikost statických dat:</b>	6771 B
<b>Velikost spustitelného souboru:</b>	12396 B