

Dokumentácia k projektu pre predmety IZP a IUS

## **Iteračné výpočty**

Projekt č. 2

25. novembra 2013

Autor: Filip Gulán, [xgulan00@stud.fit.vutbr.cz](mailto:xgulan00@stud.fit.vutbr.cz)  
Fakulta Informačných Technológií  
Vysoké Učenie Technické v Brne

# Obsah

1	Úvod	3
2	Analýza problému a princíp jeho riešenia	4
2.1	Zadanie problému	4
2.2	Druhá odmocnina z čísla	4
2.3	Arcus sinus	5
2.4	Výpočet uhlov trojuholníka daného 3 stranami	5
3	Návrh riešenia problému	6
3.1	Výpočet druhej odmocniny z čísla	6
3.2	Výpočet arcus sinus	6
3.3	Výpočet uhlov trojuholníka daného 3 stranami	6
3.4	Počet platných čísiel	7
4	Špecifikácia testov	8
5	Popis vlastného riešenia	10
5.1	Ovládanie programu	10
5.2	Voľba dátových typov	10
5.3	Vlastná implementácia	10
5.4	Optimalizácia	11
6	Záver	12
A	Metriky kódu	14

# Kapitola 1

## Úvod

Dokumentácia k druhému projektu do predmetu IZP – základy programovania, a IUS – softwarové inžinierstvo vyučovaných na Vysokom učení technickom v Brne, fakulte informačných technológií.

Tento dokument popisuje návrh a implementáciu aplikácie, ktorá počíta druhú odmocninu čísla a arcus sinus, ktoré sú využívané tiež pre uľahčený výpočet uhlov trojuholníku. Výpočty sú realizované na základe iteračných výpočtov, čo znamená, že sa opakuje určitý vzorec, dokiaľ nie je splnená určitá podmienka. Vzorec a podmienka závisí od vybranej funkcie. Program funguje ako konzolová aplikácia. Dáta číta z parametrov programu a výsledky vypisuje na štandardný výstup.

Dokument sa skladá z niekoľkých častí. V kapitole 2 sa venujem analýze problému použitia druhej odmocniny z čísla a goniometrických funkcií arcus sinus a arcus cosinus. V kapitole 3 popisujem návrh vhodného riešenia problematiky a v kapitole 4 sa venujem konkrétnej finálnej implementácii.

## Kapitola 2

### Analýza problému a princíp jeho riešenia

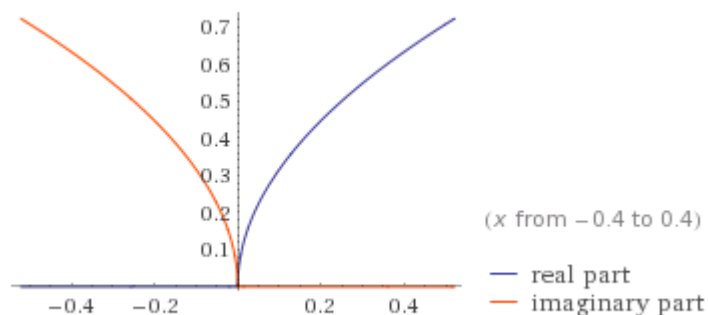
Pretože iteračné výpočty nie je triviálna problematika, pozrieme sa na nich v tejto kapitole podrobnejšie. V tejto dokumentácii je popísaný výpočet odmocniny pomocou Newtonovej metódy, výpočet arcusu sinusu pomocou Taylorovej (Maclaurinovej) rady a výpočet uhlov trojuholníka pomocou Pytagorovej vety a vzťahu arcusu cosinu a arcusu sinusu.

#### 2.1 Zadanie problému

Cieľom tohto projektu je vytvorenie programu v jazyku C, ktorý vypočíta druhú odmocninu z čísla, arcus sinus a vypíše uhly všeobecného trojuholníku. Výpočet sa uskutočňuje pomocou operácií  $+$ ,  $-$ ,  $*$ ,  $/$ . Program bude načítavať hodnoty z parametrov programu daných užívateľom. Výsledok bude vypisovaný na štandardný výstup. Tento údaj bude pri výpočte odmocniny v desiatkovej sústave a bude reprezentovať číslo po odmocnení, pri výpočte arcusu sinusu bude výsledok v jednotke radián, a výsledok výpočtu uhlov bude taktiež v jednotke radián. Na výpočet pomocou iteračných výpočtov pri odmocnине máme použiť Newtonovu metódu a pri arcus sinus zase Taylorovu radu. Presnosť aproximačných funkcií má byť na 11 platných čísiel. Všetky výpočty sa budú vypisovať pomocou formátového výstupu „%.10e“.

#### 2.2 Druhá odmocnina z čísla

Druhá odmocnina z čísla je jeden zo špeciálnych typov všeobecnej odmocniny. Druhá odmocnina sa označuje ako  $\sqrt{a}$  a v projekte je nazvaná ako sqrt. Prvým krokom pri tvorbe funkcie na výpočet odmocniny je určenie si definičného oboru a oboru hodnôt.



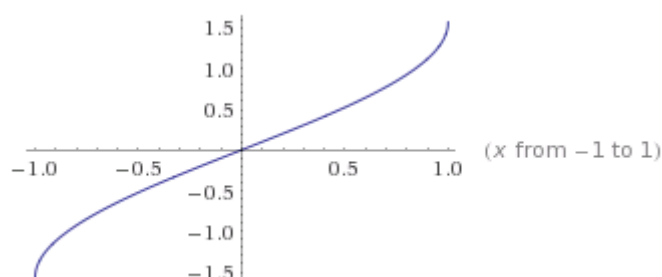
Obrázok 1: Ukážkový graf funkcie odmocniny

Ako môžeme vidieť na obrázku ukážky grafu funkcie odmocniny, definičný obor funkcie je obmedzený na interval  $0 < x$ , teda definičným oborom odmocniny sú všetky kladné čísla

vrátane nuly. Obor hodnôt sú všetky reálne čísla, napr. odmocnina z čísla 4 môže byť tak ako 2, tak aj -2. Pre zjednodušenie budeme brať výsledok odmocniny definovaný na množine reálnych čísiel len ako kladnú hodnotu, teda v tomto prípade 2.

## 2.3 Arcus sinus

Arcus sinus je inverzná funkcia k funkcii sinus. Je označovaná ako  $\sin^{-1}(x)$  a v projekte je nazvaná ako asin. Tak, ako sme postupovali pri druhej odmocnine, tak aj tu je dobré si najprv definovať definičný obor a obor hodnôt.

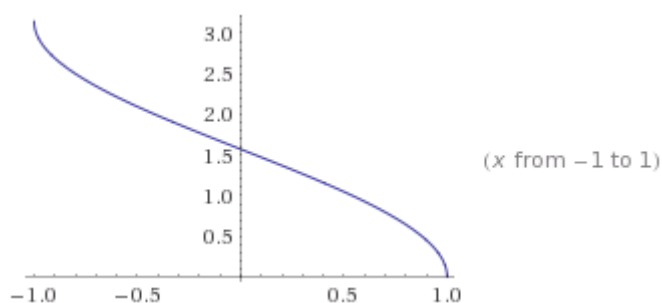


Obrázok 2: Ukážkový graf funkcie arcus sinus

Z obrázku vidíme, že definičný obor sú čísla z intervalu  $<-1,1>$  a obor hodnôt je  $<-\pi/2, \pi/2>$  čo znamená, že najmenšia prípustná hodnota v stupňoch je  $-90^\circ$  a najväčšia je  $90^\circ$ .

## 2.4 Výpočet uhlov trojuholníka daného 3 stranami

Výpočet uhlov trojuholníka daného 3 stranami môžeme robiť viacerými metódami. Najjednoduchšou a najbežnejšou metódou je arcus cosinus, teda inverzná funkcia ku cosinu a označuje sa  $\cos^{-1}(x)$ .



Obrázok 3: Ukážkový graf funkcie arcus cosinus

Definičný obor funkcie je  $<-1,1>$  a obor hodnôt je  $<0, \pi>$ . Ako uvidíte v kapitole 3.3, tak existuje vzťah medzi funkciou arcus sinus a funkciou arcus cosinus, ktorý pre zrejme uľahčenie práce a zefektívnenie programu, aj využijeme.

## Kapitola 3

### Návrh riešenia problému

Pri návrhu riešenia problému som striktne vychádzal zo zadania a dodržal som všetky body, ktoré boli spísané v zadní projektu.

#### 3.1 Výpočet druhej odmocniny z čísla

Riešenie druhej odmocniny je možné hneď niekoľkými postupmi. Po analýze problému a preštudovaní zadania, som sa však rozhodol použiť Newtonovu metódu pre druhú odmocniny z čísla, ktorej základom je vzorec zobrazený nižšie.

$$y_{i+1} = \frac{1}{2} * (x/y_i + y_i) \quad (3.1)$$

Z rekurentného vzorca môžeme jasne vidieť, ako sa bude druhá odmocnina z čísla počítať.

#### 3.2 Výpočet arcus sinus

Arcus sinus som sa rozhodol riešiť pomocou Taylorovej rady.

$$\arcsin x = x + \frac{1}{2} * \frac{x^3}{3} + \frac{1 * 3}{2 * 4} * \frac{x^5}{5} + \frac{1 * 3 * 5}{2 * 4 * 6} * \frac{x^7}{7} \dots \quad (3.2)$$

Zo vzorca Taylorovej rady, pre výpočet arcus sinus, som si vybral 2 a 3 člen a zistil, čím sa líšia, ďalej som skúšal ďalšie susediace dvojice a nakoniec som zistil vzťah jednotlivých členov Taylorovej rady.

#### 3.3 Výpočet uhlov trojuholníka daného 3 stranami

Najprv som si vypočítal súradnice vrcholov a vďaka vzorcu z goniometrie a Pytagorovej vete som vypočítal jednotlivé strany.

$$\begin{aligned} c &= (BX - AX)^2 + (BY - AY)^2 \\ b &= (CX - AX)^2 + (CY - AY)^2 \\ a &= (CX - BX)^2 + (CY - BY)^2 \end{aligned} \quad (3.4)$$

Výpočet uhlov trojuholníka daného 3 stranami, už nie je veľmi náročný proces. Keďže máme vzťah pre odmocninu (kapitola 3.1) a arcus sinus (kapitola 3.2) už definovaný, pri výpočte uhlov trojuholníka som sa rozhodol nepočítať arcus cosinus, ale použiť vzťah medzi arcus sinus a arcus cosinus, a tak si ušetriť čas a zefektívniť prácu.

$$\arccos x = \frac{\pi}{2} - \arcsin x \quad (3.5)$$

### 3.4 Počet platných čísiel

Keďže iteračné vzťahy tvoria nekonečne dlhú postupnosť a čísla, ktoré sa pripočítavajú k rade, sú stále menšie a menšie, je teda nevyhnutné si zvoliť určité číslo, po ktorom prekročení sa počítanie zastaví. Takéto číslo označujeme  $\varepsilon$  a nazývame ho presnosť výpočtu. Výpočet rekurentného vzťahu prebieha dovtedy, kým platí nasledujúca podmienka:

$$|Y_{i-1} - Y_i| \geq \varepsilon * |Y_i| \quad (3.6)$$

Takáto podmienka sa nazýva relatívna presnosť, pretože určuje relatívnu veľkosť rozdielu posledných 2 členov k celkovej sume (posledného členu) vynásobeného  $\varepsilon$ .

# Kapitola 4

## Špecifikácia testov

Program som testoval na školskom stroji Melrin a Eva, ktorý bežia na operačnom systéme CentOS 6.4. Pri testovaní som sa snažil zamerať na všetky typy možností, ktoré môžu nastať. Testoval som správne, ale aj nesprávne použitia parametrov. Výsledky testov sú z Merlina, keďže podľa zadania, by mal program pracovať primárne práve na ňom. Všetky výsledky sú totožné s výsledkami internetovej vedeckej kalkulačky menom Wolfram Alpha.

**Test 1:** Druhá odmocnina z čísla 2.

Parametre: `./proj2 --sqrt 2`  
Výstup: `1.4142135624e+00`

**Test 2:** Hodnota mimo definičný obor druhej odmocniny z čísla → vypíše nan (not a number).

Parametre: `./proj2 --sqrt -2`  
Výstup: `nan`

**Test 3:** Chybný parameter programu → vypíše chybovú hlášku a ukončí sa.

Parametre: `./proj2 --sqrt 4e`  
Výstup: `Argumenty neboli zadane spravne!`

**Test 4:** Arcus sinus z čísla 0.5.

Parametre: `./proj2 --asin 0.5`  
Výstup: `5.2359877560e-01`

**Test 5:** Arcus sinus z čísla -0.5.

Parametre: `./proj2 --asin -0.5`  
Výstup: `-5.2359877560e-01`

**Test 6:** Chybný parameter programu → vypíše chybovú hlášku a ukončí sa.

Parametre: `./proj2 --asin 0.5r`  
Výstup: `Argumenty neboli zadane spravne!`

**Test 7:** Hodnota mimo definičný obor arcus sinus → vypíše nan (not a number).

Parametre: `./proj2 --asin 2.5`  
Výstup: `nan`

**Test 8:** Vypočíta 3 uhly trojuholníka.



Parametre: ./proj2 --triangle 0 0 1 0 0 2  
Výstup: 1.5707963268e+00  
1.1071487178e+00  
4.6364760900e-01

**Test 9:** Zadané vrcholy netvorí trojuholník → vypíše nan (not a number).

Parametre: ./proj2 --triangle 0 0 0 0 0 0  
Výstup: nan  
nan  
nan

**Test 10:** Chybný parameter programu → vypíše chybovú hlášku a ukončí sa.

Parametre: ./proj2 --triangle 0 0 1a 0 0 2  
Výstup: Argumenty neboli zadane spravne!

# Kapitola 5

## Popis vlastného riešenia

Pri implementácii som vychádzal zo záverov, ktorých problematiku som opísal v predchádzajúcich kapitolách. Všetky výpočty jednotlivých funkcií, sú tvorené na základe vzorcov odvodených v kapitole 3.

### 5.1 Ovládanie programu

Program je konzolová aplikácia a teda všetka interakcia užívateľa s programom prebieha cez príkazový riadok (konzolu). Program sa spúšťa s nasledujúcimi parametrami:

--help	vypíše nápovedu k programu.
--sqrt X	program vypočíta a vypíše na štandardný výstup druhú odmocninu z čísla X.
--asin X	program vypočíta a vypíše na štandardný výstup arcus sinus z čísla X.
--triangle AX AY BX BY CX CY	program vypočíta uhly trojuholníka, ktorý je daný troma vrcholmi A=AX,AY, B=BX,BY a C=CX,CY.

Výhodou spomenutého ovládania programu je, že program môže byť využitý v skriptoch alebo spolu s inými programami vo väčšom celku.

```
xgulan00@merlin: ~$ ./proj2 --triangle 0 0 1 0 0 2
1.5707963268e+00
1.1071487178e+00
4.6364760900e-01
xgulan00@merlin: ~$
```

Obrázok 4: Ukážka spustenia programu na Merlinovi (CentOS 6.4)

### 5.2 Voľba dátových typov

Keďže v zadaní úloh bolo požadované, aby všetky výpočty boli typu double, všetky moje funkcie okrem main vracajú typ double. Taktiež väčšina mojich premenných je typu double. Okrem double, som použil typ ukazateľ na char kvôli spracovaniu argumentov programu.

### 5.3 Vlastná implementácia

Parametre príkazového riadku spracováva funkcia main vďaka funkcií strtod z knižnice stdlib.h. V prípade nesprávne zadaných parametrov, program vypíše chybovú hlášku a ukončí

sa. V prípade správneho tvaru parametrov, sa spustí jedna z mojich naprogramovaných funkcií.

Funkcia `my_sqrt(x)` vypočíta druhú odmocninu z čísla  $x$ . Funkcia `my_asin(x,y)` vypočíta arcus sinus  $x$  s presnosťou  $y$ . A funkcia `triangle(ax,ay,bx,by,cx,cy)` vypočíta 3 uhly trojuholníka. Všetky funkcie si strážia svoj definičný obor a v prípade jeho prekročenia, vypisujú na štandardný výstup hlášku `nan`. Okrem toho všetky funkcie využívajú pri výpočte moju funkciu `my_abs(x)`, ktorá počíta absolútnu hodnotu čísla  $x$ . Všetky funkcie, po skončení počítania, vypíšu výsledok svojej operácie na štandardný výstup pomocou `printf()`.

## 5.4 Optimalizácia

Po naprogramovaní programu, som sa ho rozhodol ešte optimalizovať. Keďže viem, že arcus sinus je pre 0 vždy 0, tak som tejto vedomosti využil, a aby nemusel program zbytočne počítať, dal do kódu, tak povediac „natvrdo“ `if`. Niečo podobné som použil pri hodnote arcus sinus 1 a -1, kde som tiež použil `if` a vypočítal arcus sinus týchto hodnôt jednoducho pomocou vzorca

$$vysledok = \pi/2 \tag{5.1}$$

a potom už iba doplnil znamienko podľa toho či išlo o 1 alebo -1.

Optimalizovať sa dala taktiež funkcia `triangle`, kde sme nemusel počítať všetky uhly pomocou vzorca (3.4), ale stačili vypočítať 2 uhly a tretí dostať vďaka vlastnosti trojuholníka, že po súčte všetkých troch uhlov dostaneme rovno  $\pi$ .

## Kapitola 6

### Záver

Program počíta druhú odmocninu z čísla, arcus sinus z čísla a uhly trojuholníku. Všetky výpočty som porovnal s výsledkami daných funkcií z knižnice math.h a výsledkami vedeckej online kalkulačky Wolfram Alpha, a na požadovanú presnosť boli výsledky rovnaké. Program presne splňuje požiadavky vstupných a výstupných dát, ktoré boli dané v zadaní. Program je možné bezproblémovo ďalej používať v ďalších programoch alebo skriptoch.

Program bol úspešne otestovaný na niekoľkých operačných systémoch, zahrňujúc systémy Windows a Linux (presnejšie Windows 8.1, Windows XP, CentOS 6.4, Ubuntu 13.1).

Program je prenositeľný na všetky druhy operačných systémov, ktoré dokážu spustiť programy naprogramované v jazyku C.

## Použitá literatura a zdroje

HANS-JOCHEN BARTSCH. *Matematické vzorce*. 3. vyd. Praha: Mladá fronta, 1996, 831 s. ISBN 80-204-0607-7.

WOLFRAM ALPHA LLC—A WOLFRAM RESEARCH COMPANY. *Wolfram/Alpha: Computational Knowledge Engine*[online]. © 2013 [cit. 2013-11-30]. Dostupné z: <http://www.wolframalpha.com/>

# Príloha A

## Metriky kódu

**Počet súborov :** 1 súbor

**Počet funkcií:** 6 (aj s main)

**Počet riadkov zdrojového textu:** 250 riadkov

**Veľkosť statických dát:** 520B

**Veľkosť spustiteľného súboru:** 10518B (systém Linux, 32 bitová architektúra)