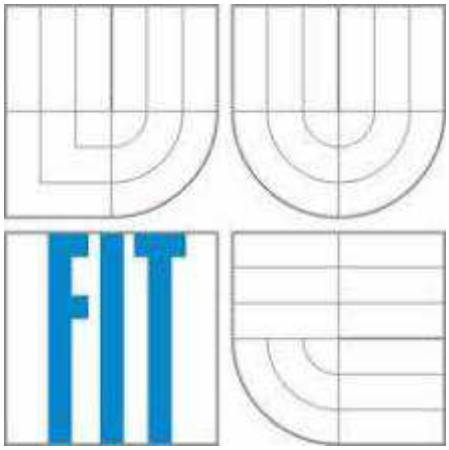


Autor:
Václav Bayer, xbayer05@stud.fit.vutbr.cz
Fakulta Informačních Technologií

Dokumentace k projektu pro předměty IZP a IUS

Iterační výpočty

projekt č. 2

6. prosince 2013

Obsah

1 Úvod

Tato dokumentace se zabývá návrhem a implementací aplikace pro výpočet úhlů v obecném trojúhelníku, druhé odmocniny a arkus sinus. Výsledný program, fungující jako konzolová aplikace načte data ze vstupu, se kterými pak dále pracuje při svých dalších výpočtech. Celé chování programu může ovlivnit uživatel zadanými parametry a zadanými hodnotami na vstupu. Program při správném průběhu vypíše na standartní výstup výsledek, při nesprávném průběhu chybové hlášení, případně nápovědu.

2 Analýza problému a princip jeho řešení

V této kapitole rozeberu, jakými způsoby lze algoritmus řešit, jak se dívat na program jako na jednotlivé části místo celku. Pokusím se zaměřit na více možností jak dojít k požadovanému výsledku.

2.1 Zadání problému

Cílem projektu je vytvořit program v jazyce C, který vypočítá druhou odmocninu, arkus sinus nebo velikosti úhlů v trojúhelníku. Velikosti úhlů se musí počítat striktně dle Pythagorovy a Kosinovy věty. Typ výpočtu si volí sám uživatel na standartním vstupu pomocí argumentů specifických pro daný výpočet. Po udání argumentu je potřeba zadat hodnoty, se kterými bude program provádět výpočet.

V případě výpočtu úhlů se uvádí šest číselných hodnot značící vrcholy obecného trojúhelníku A=AX,AY, B=BX,BY a C=CX,CY. Na standartní výstup jsou pak vytisknuty tři hodnoty vyjadřující úhly přesně v tomto pořadí: alfa, beta, gama. V ostatních případech uživatel zadá jen jednu hodnotu a na standartním výstupu se vytiskne druhá odmocnina nebo arkus sinus – záleží na zvoleném argumentu.

Ve všech případech je rozsah výstupu na 11 platných číslic. V případě zadání neplatných hodnot (neodpovídajícím daným pravidlům) je program ukončen chybovým výstupem.

2.2 Konstanty

Předdefinované konstanty:

a) **EPS 1e-13** -> ze zadání zní, že přesnost je na 11 platných číslic, já nadstavil pro jistotu a větší přesnost výpočtu 13 platných číslic.
b) **PI 3.14159265358979323846** -> hodnota π na dvacet desetinných míst by měla být pro tento výpočet dostatečná.
c) **PIPUL 1.57079632679489661923** -> $\frac{\pi}{2}$ na dvacet desetinných míst.

Standartní vstup se skládá ze dvou částí:

a) **Parametr:**
   a. --sqrt -> výpočet druhé odmocniny
   b. --asin -> výpočet arkus sinus
   c. --triangle -> výpočet úhlů v trojúhelníku
b) **Hodnota**

Uživatel si pomocí argumentu volí typ výpočtu, který chce povést. Při zadání argumentu --sqrt nebo --asin musí uživatel zadat také hodnotu, která je oddělena mezerou. Při výpočtu úhlů se uvádějí souřadnice bodu trojúhelníku v následujícím pořadí:

AX AY BX BY CX CY

2.3 Přesnost výpočtu

Přesnost nám určuje, jak dlouho bude program počítat a na jaký počet desetinných míst bude výpočet realizován. Přesnost výpočtu je dle zadání na 11 platných číslic, avšak já zvolil pro přesnější výpočet 13 platných číslic. První platnou číslicí se rozumí první číslice z leva, která má nenulovou hodnotu. Výsledek výpočtu je vypsán na 11 platných číslic, ale je počítán s přesností na 13 platných číslic. Jestliže je číslo delší jak 13 číslic, zbytek už není důležitý.

Pro představu mějme interval, pro který platí, že X je libovolné číslo:

$$< X - X * 10^{-13}; X + X * 10^{-13} >$$

Číslo, nacházející se v tomto intervalu splňuje požadavky na naši přesnost.

2.4 Konvergence funkcí

Je potřeba si hned na začátek říci co to **konvergence** vůbec je. Konvergence – název pocházející z latinského convergere tj. com + Vergere = ohýbat dohromady. Jinak řečeno konvergence značí sbíhání, splývání, sbližování apod. Opakem konvergence je **divergence**.

Pro výpočet některých řad nastává problém omezené konvergence a to tehdy, když řada konverguje jen v omezeném intervalu, který nemusí obsahovat celý definiční obor funkce. Jestliže je konvergence velmi náročná, v lepším případě může program počítat nechtěně dlouho. V horším případě se program může nečekaně zacyklit.

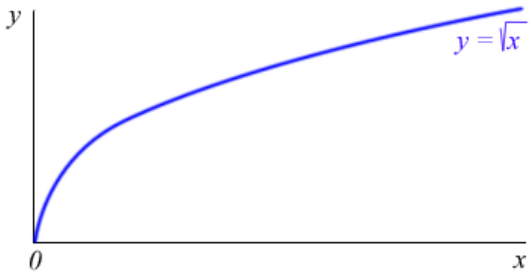
Aby se této situaci dalo předejít a abychom se chtěli výsledku co nejrychleji přiblížit, musí se najít interval, pro který daná funkce s použitím určité řady bude konvergovat nejlépe a pro který interval nebude výpočet příliš dlouhý.

2.5 Druhá odmocnina

Než se začnu zabývat odmocninou, připomenu trochu středoškolské matematiky pro zopakování: odmocňování je inverzní funkcí k umocňování a výsledkem operace je odmocnina. Záporné číslo odmocňovat nelze.

K našemu výpočtu použijeme iterační numerickou metodu nazvanou Newtonová metoda. Rekurentní rovnice je následující:

$$\sqrt{x} = y_n, \quad y_0 = 0, \quad y_{i+1} = \frac{1}{2}\left(\frac{x}{y_i} + y_i\right)$$



2.6 Arkus sinus

Arkus sinus je cyklometrická funkce, která je inverzní ke goniometrické funkci sinus.

Definiční obor: <-1; 1>

Obor hodnot: $\left\langle -\frac{\pi}{2}; \frac{\pi}{2} \right\rangle$

Taylorův polynom pro výpočet arkus sinus:

$$\text{Arcsin(x)} = x + \frac{1}{2} \cdot \frac{x^3}{3} + \frac{1 \cdot 3}{2 \cdot 4} \cdot \frac{x^5}{5} + \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6} \cdot \frac{x^7}{7} + \cdots$$
$$= \sum_{n=0}^{\infty} \left(\frac{(2n)!}{2^{2n}(n!)^2} \cdot \frac{x^{2n+1}}{2n+1} \right), |x| < 1$$

Arkus sinus se dá také spočítat nepřímo a to pomocí Taylorova polynomu pro arkus tangens. Taylorův polynom pro arkus tangens a vztah mezi funkcemi:

$$\text{Arctan(x)} = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \cdots$$
$$\sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{2n+1}, \; x \in \langle -1; 1 \rangle$$

$$\text{Arcsin(x)} = \text{Arctan}\left(\frac{x}{\sqrt{1-x^2}}\right)$$

3 Návrh řešení problému

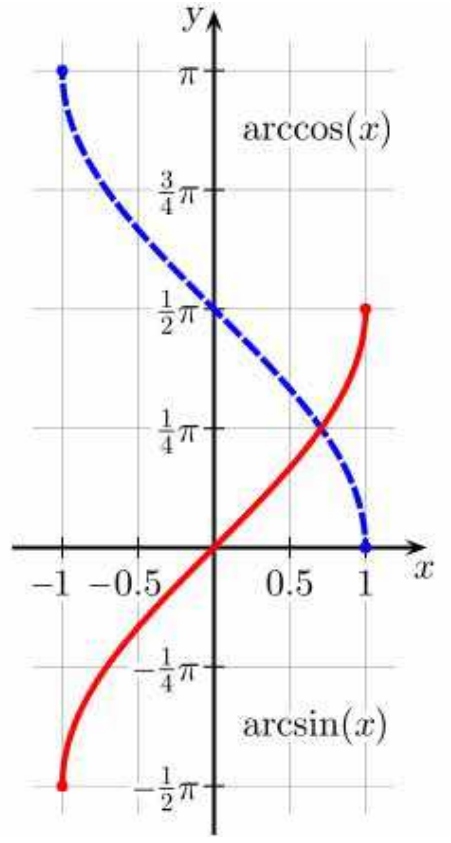
3.1 Zpracování vstupu

Věnuji pár vět ošetření vstupních argumentů a hodnot. Někomu se může zdát, že tato část není tak důležitá, aby se jí člověk vůbec zabýval. Ale položte si otázku: co provede program při nechtěném vstupu? Načtením a zpracováním nesprávných hodnot můžeme dostat nesprávné výsledky a nesprávný výpočet vůbec.

Ošetřit vstup argumentů není žádný problém, pomocí funkce strcmp() hned rozeznáme, jedná-li se o chtěný argument či nikoli. Jestliže se jedná o nechtěný argument, program vrací defaultně nápovědu.

Ošetřit ale vstup hodnot, které následují hned za argumentem už je o něco málo těžší záležitost. Na vstupu se může objevit jakékoli reálné číslo, které může navíc obsahovat písmeno „e“ značící exponent, znaménko „-˝ nebo „.“ značící desetinnou čárku. V tomto případě je nejlepší procházet jednotlivý znak po znaku a porovnávat jestli spadá do intervalu povolených číslic a znaků. Při tomto postupu je nutno brát ohled na to, že znak „-„ se může vyskytnout nejen na začátku vstupní hodnoty.

3.2 Arkus sinus



Pro výpočet arkus sinus jsme si uváděli v kapitole **2.6** dva postupy. Není nutné popisovat dopodrobna oba dva postupy. Je možné, že někomu by vyhovoval druhý postup pomocí arkus kotangens, avšak v mém výpočtu jsem se zabýval jen prvním postupem a to je přímý výpočet arkus sinus.

Potřeba je ale dát si pozor při výpočtu hodnot blízkým krajním hodnotám. Tak jak jsem již zmiňoval v kapitole **2.4** o konvergencích, bude potřeba si najít interval, kde bude výpočet konvergovat nejrychleji, bez problémů a který bude spadat do definičního oboru funkce.

Demonstruji na příkladu: ve sjednoceném intervalu $\langle -\mathbf{1};\ -\mathbf{0},\mathbf{5}) \cup (\mathbf{0},\mathbf{5};\ \mathbf{1}\rangle$ začíná funkce arkus sinus konvergovat pomaleji a její počet iterací se každým desetinným místem rapidně zvyšuje. Abychom tomu zabránili, využijeme vztahu $\boldsymbol{\arccos x = arcsin\sqrt{1 - x^2}}$. Jestli tento vzorec nahání od pohledu hrůzu, nebo uvažujete, kde se tam vzal arkus kosinus, není se čeho bát. Arkus kosinus má stejný Taylorův polynom jako arkus sinus, jenom s tím rozdílem že se výsledek odečítá od $\frac{\pi}{2}$.

Pro ještě větší efektivnost výpočtu jsem nadstavil okamžité vypsání výsledku, jestliže uživatel zadá na vstup hodnoty spadající do okrajových hodnot definičního oboru funkce arkus sinus. Tzn., pokud uživatel na vstup zadá hodnotu -1, je mu bez výpočtu vrácená hodnota - $\frac{\pi}{2}$. Stejně tak tahle podmínka platí i pro číslo 1, kde je vrácená opačná hodnota $\frac{\pi}{2}$.

Taylorův polynom pro arkus kosinus:

$$\text{Arccos(x)} = \frac{\pi}{2} - x - \frac{1}{2} \cdot \frac{x^3}{3} - \frac{1 \cdot 3}{2 \cdot 4} \cdot \frac{x^5}{5} - \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6} \cdot \frac{x^7}{7} + \cdots$$
$$= \frac{\pi}{2} - x - \sum_{n=0}^{\infty} \left(\frac{(2n)!}{2^{2n}(n!)^2} \cdot \frac{x^{2n+1}}{2n+1} \right), |x| < 1$$

3.3 Úhly v obecném trojúhelníku

Trojúhelník je rovinný geometrický útvar se třemi vrcholy a třemi stranami. Úsečky spojující vrcholy se nazývají strany trojúhelníka, úhly které svírají strany, se nazývají vnitřní úhly. Ze zadání plyne, že máme při výpočtu striktně vycházet z Pythagorovy a Kosinovy věty.

Na následujících příkladech budu demonstrovat postup výpočtu úhlů.

Délka strany, uplatnění Pythagorovy věty

$$a_1 = |BX - CX|, a_2 = |BY - CY|$$

$$a = \sqrt{a_1^2 + a_2^2}$$

Pro strany v obecném trojúhelníku platí, že součet délek dvou stran je vždy větší než třetí strana. To znamená, že pokud nelze projít touto podmínkou, bude asi chyba někde ve výpočtu nebo byly zadány nesprávné hodnoty na vstupu, ze kterých trojúhelník nelze sestrojit. Program bude ukončen s vypsáním chybové zprávy.

Následně využijeme Kosinovy věty:

$$cos\alpha = \frac{-a^2 + b^2 + c^2}{2bc}$$

V závislosti na vztahu $\arccos(x) = cos^{-1}(x)$ je výpočet opět o něco blíže výsledku. Pokud některá z výsledných hodnot $(cos\alpha, cos\beta, cos\gamma)$ spadá do intervalu $\langle -0{,}5;\ 0 \rangle$ jedná se o tupý úhel a výsledný vzorec pro výpočet je $\arccos(x) = \frac{\pi}{2} + \arcsin|x|$. Pokud do tohoto intervalu nespadá, počítá se dle vzorce $\arccos(x) = \frac{\pi}{2} - \arcsin|x|$.

3.4 Specifikace testů

Je načase nechat potrápit výsledný program nejen na bezproblémových ale i na problémových a okrajových hodnotách. Na následujících testech lze vidět, jak si program poradí s těmito hodnotami.

Druhá odmocnina

Příkazová řádka	Kalkulačka	Výstup
$ ./proj2 --sqrt 0	0	0.0000000000e+000
$ ./proj2 --sqrt 1	1	1.0000000000e+000
$ ./proj2 --sqrt 20	4.4721359550	4.4721359550e+000
$ ./proj2 --sqrt 2e-30e	NaN	nan

Arkus sinus

Příkazová řádka	Kalkulačka	Výstup
$ ./proj2 --asin 0.999999999	1.570751605	1.5707516054e+000
$ ./proj2 --asin -0.999999999	-1.570751605	-1.5707516054e+000
$ ./proj2 --asin 0	0	0.0000000000e+000
$ ./proj2 --asin 2e-30	0.000000000	2.0000000000e-030
$ ./proj2 --asin 0.75	0.848062079	8.4806207898e-001

Úhly v obecném trojúhelníku. Zkouška všech kvadrantů pro bod c.

Příkazová řádka	Wolfram Alpha	Výstup
$ ./proj2 --triangle 0.5 0 1 1 2 2	0.1799	1.7985349979e-001
	2.82	2.8198420992e+000
	0.1419	1.4189705460e-001
$ ./proj2 --triangle 0.5 0 1 1 -1 2	1.107	1.1071487178e+000
	1.571	1.5707963268e+000
	0.4636	4.6364760900e-001
$ ./proj2 --triangle 0.5 0 1 1 -1 -2	2.962	2.9617391538e+000
	0.1244	1.2435499455e-001
	0.055	5.5498505246e-002
$ ./proj2 --triangle 0.5 0 1 1 1 -2	0.245	2.4329663815e-000
	0.4636	4.6364760900e-001
	2.433	2.4497866313e-001

Extrémně ostrý úhel α.

Příkazová řádka	Wolfram Alpha	Výstup
$ ./proj2 --triangle -100 1 100 1 100 2	0.00499996	4.9999583368e-003
	1.5708	1.5707963268e+000
	1.5658	1.5657963685e+000

Extrémně tupý úhel α.

Příkazová řádka	Wolfram Alpha	Výstup
$ ./proj2 --triangle -100 1 100 1 0 2	0.00999967	9.9996666874e-003
	0.00999967	9.9996666874e-003
	3.12159	3.1215933202e+000

Není trojúhelník ale přímka.

Příkazová řádka	Wolfram Alpha	Výstup
$ ./proj2 --triangle 0 0 1 1 2 2	0	nan
	0	
	3.14159	

4 Popis řešení

Při implementaci jsem postupoval podle výše uvedených řešení daných problémů.

4.1 Volba datových typů

V celém programu je převážně použit datový typ double. Datový typ int je zde zastoupen minimálně. Nejmenší zlomkem z použitých datových typů by se ještě dal zmínit datový typ char. Nejdůležitějším datovým typem však pro nás je typ již zmíněný typ double, který nejlépe odpovídá požadavkům na rozsah potřebných dat.

4.2 Implementace funkcí

Začnu logicky od ověřování zadaných parametrů, které se porovnávají ve funkci main pomoci funkce switch. Projdou-li parametry porovnáním, porovnávají se hodnoty funkcí je_cislo a dále jsou volány určité funkce dle zadaných parametrů.

Arkus sinus. Funkce my_asin ověří hodnotu, se kterou má pracovat, jestli není mimo povolený interval, když nevyhovuje, vrací nan. Jestliže vyhovuje, pokračuje dál a provede iterační výpočet. Iterace probíhají do té doby, dokud podmínka, která při každé iteraci přepočítává epsilon, dokud nespadá do intervalu požadované přesnosti. Jakmile je tento cyklus ukončen dostáváme výsledek.

Arkus kosinus. Funkce my_acos je volána v případě, když se vstupní hodnota blíží okrajovým hodnotám a výpočet by byl v tom případě zdlouhavý (viz. kapitola **3.2**.)

Druhá odmocnina. Funkce my_sqrt probíhá stejně jako funkce my_asin. Rozdíl je akorád v intervalu a iteračním výpočtu. Podmínka pro výpočet přesnosti zůstává pořád stejná.

Ověření hodnot. Funkce je_cislo vezme vstupní hodnotu a porovnává znak po znak. Porovnávaný znak musí spadat do intervalu číslic 0-9. Může být tečkou, znaménkem mínus,

nebo písmenem „e“, který značí exponent. Znaménko mínus se přitom může vyskytovat dvakrát.

Absolutní hodnota. Vlastní funkce my_abs nahrazuje funkci abs, která spadá do knihovny <math.h>, která není pro tento projekt povolena.

Délka strany. Funkce delka_strany vypočítá délku strany obecného trojúhelníka pomocí Pythagorovy věty. Funkci si volá funkce uhly.

Výpočet úhlů. Funkce uhly počítá velikosti úhlů obecného trojúhelníku pomocí Kosinovy a Pythagorovy věty. V těle funkce je volána funkce delka_strany, díky které získáme parametry délek stran. Využitím Kosinovy věty a funkce my_acos se dostaneme k požadovaným výsledkům.

Závěr

Tento program lze implementovat do náročnějších programů jako podprogram.
Program byl testován na systémech Windows.

Reference

BARTSCH, H.-J.: *Matematické vzorce*. Praha: Mladá fronta, 3.vydání, 1996, 831 s., ISBN 80-204-0607-7

Metriky kódu

Počet funkcí:	9
Počet souborů:	1 soubor
Počet řádků zdrojového textu:	342
Velikost kódu programu:	2170 B
Velikost statických dat:	512 B
Velikost spustitelného souboru:	8925 B (systém Windows 32bitová verze)