Dokumentace k projektu pro předměty IZP a IUS

Iterační výpočty

projekt č. 2

19. listopadu 2012

Autor: Roman Blanco, xblanc01X@stud.fit.vutbr.cz
Fakulta Informačních Technologií
Vysoké Učení Technické v Brně

Obsah

1 Úvod

V tomto dokumentu je popsáno mé řešení 2. projektu do předmětu IZP - základy programování na VUT v Brně, fakultě informačních technologií. Jsou zde popsány postupy řešení při vytváření matematických funkcí, jež byly zadané, tedy mocninná funkce (s reálným exponentem), arkus tangens a argument hyperbolického sinu.

Dokument se skládá z několika částí. V kapitole 2 se věnuji analýze problémů spojených s programováním matematických funkcí. Popisem možných řešení se zabývám v kapitole 3.

2 Analýza problému

2.1 Zadání problému

Zadáním druhého projektu bylo vytvořit program v programovacím jazyce C (ISO C99), který načte vstupní hodnotu, a podle parametrů vypočítá hodnotu požadované funkce pomocí základních matematických operací ($+, -, *, /$). Vypočet probíhá pomocí iterací, které probíhají, dokud není u výsledné hodnoty dosaženo požadované přesnosti, jež je zadána uživatelem při spuštění programu. Program by měl být schopný rozpoznat chybně zadané parametry, a zareagovat na ně patřičným chybovým hlášením. Ošetřeny jsou i matematicky nedefinované výrazy, jako např. u mocninné funkce 0^0

2.2 Mocninná funkce

Mocninná funkce (v projektu jako `powxa`) je dána vztahem $y = x^a$ kde $x \in (0; \infty)$ a $a \in R$. Pro výpočet funkce je použito Eulerovo číslo e a přirozený logaritmus. Zápis x^a lze nahradit jako e^{a*lnx}. Součtový rozvoj pro mocninnou funkci tedy bude

$$x^a = e^{a*lnx} = \sum_{k=0}^{\infty} \frac{(a*lnx)^k}{k!}$$

2.3 Arkus tangens

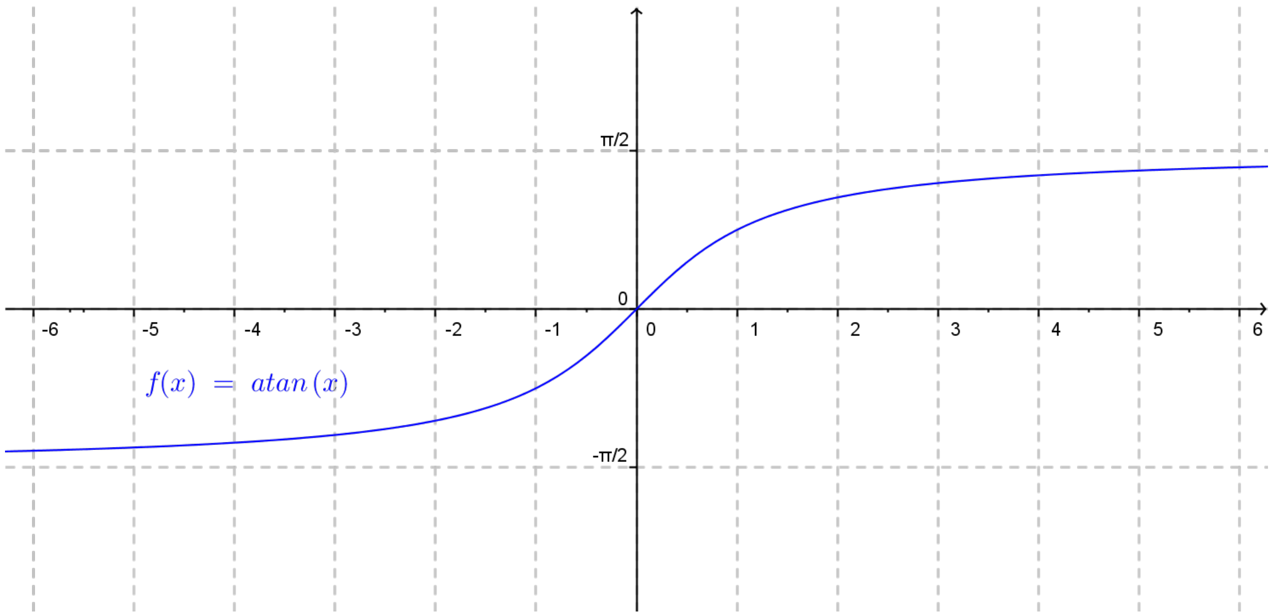
Funkce arkus tangens (v projektu nazvaná zkratkou `arctg`) je cyklometrická funkce, a je inverzní funkcí ke goniometrické funkci tangens. Definičním oborem funkce jsou všechna reálná čísla, oborem hodnot je interval $(-\frac{\pi}{2}; \frac{\pi}{2})$

Arkus tangens lze vyjádřit součtovým vzorcem

$$\arctan x = \frac{\pi}{2} - \sum_{k=1}^{\infty} (-1)^k * \frac{1}{(2k-1)*x^{(2k-1)}}$$

Tento vztah platí pro čísla, jejichž absolutní hodnota je větší než 1, a které je nutné odečíst od periody. Pro čísla, jejichž absolutní hodnota je menší než 1 lze použít vztah

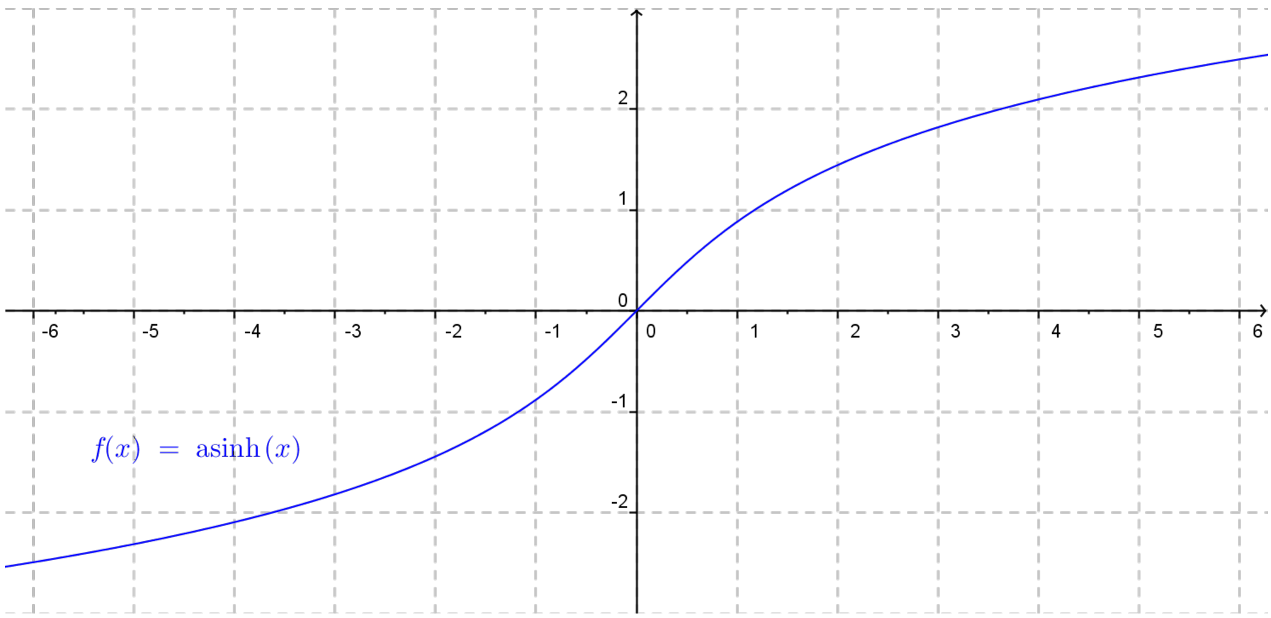
$$\arctan x = \sum_{k=0}^{\infty} (-1)^k * \frac{x^{1+2k}}{1+2k}$$



Obrázek 1: Graf funkce arkus tangens

2.4 Argument hyperbolického sinu

Argument hyperbolického sinu (v projektu jako argsinh) je jednou z hyperbolomických funkcí. Jsou to funkce inverzní k funkcím hyperbolickým. Definičním oborem i oborem hodnot funkce argsinh jsou všechna reálná čísla.



Obrázek 2: Graf funkce argument hyperbolického sinu

Pro výpočet funkce je potřeba přirozený logaritmus:

$$argsinhx = \ln(2x) + \sum_{k=1}^{\infty}(-1)^{k-1} * \frac{(2k-1)!!}{2k * (2k!!) * x^{2k}}$$

3 Návrh řešení problému

3.1 Mocninná funkce

Nekonečnou součtovou řadu mocninné funkce je potřeba přizpůsobit pro výpočet převedením na rekurentní vztah:

$$1 + \frac{a * lnx}{1} + \frac{a * lnx^2}{2} + \frac{a * lnx^3}{3} + \dots$$

Pro získání nového členu z původního je potřeba vynásobit původním prvkem:

$$t_{y+1} = t_y * \frac{a * lnx}{i}$$

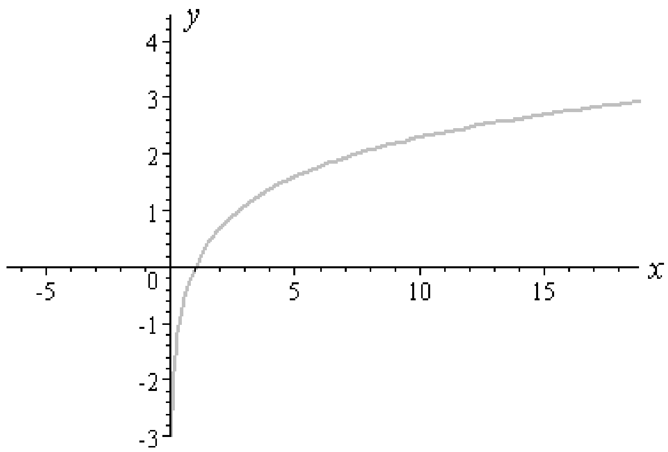
Pro zjednodušení výpočtu rekurentním vztahem lze nahradit součin $a * lnx$ jako konstantu r. Součtová řada tedy bude:

$$1 + \frac{r}{1} + \frac{r^2}{2} + \frac{r^3}{3} + \dots$$

3.1.1 Přirozený logaritmus

Pro přirozený logaritmus platí součtový vzorec

$$lnx = 2 \sum_{k=1}^{\infty} \frac{\frac{x-1}{x+1}^{2n-1}}{2n-1} = 2 * (\frac{x-1}{x+1} + \frac{1}{3} * \frac{x-1}{x+1}^3 + \frac{1}{5} * \frac{x-1}{x+1}^5 + \dots)$$



Obrázek 3: Graf funkce přirozený logaritmus

Přirozený logaritmus je defionvaný pouze pro čísla v intervalu $(0; \infty)$, při zadání záporné hodnoty tedy funkce počítající logaritmus vypíše hlášení o tom, že pro dané číslo není tato funkce definována (`NaN` - *Not a Number*)

Součtové členy se u přirozeného logaritmu učují vzorcem

$$t_{y+1} = t_y * (-\frac{(i-1) * (i-2)}{i^2 * x^2})$$

V intervalu $(0;1)$ je však lnx záporný a při výpočtu těchto hodnot běžným způsobem dochází k zacyklení programu. Při výpočtu těchto hodnot je tedy nutné udělat úpravu výrazu lnx, aby při vypočtu těchto hodnot nedocházelo k chybám za pomocí vzoce $log_e x^r = r * ln_e x$

Příklad: $\ln 0.5 = \ln(\frac{1}{0.5})^{-1} = -1 * \ln \frac{1}{0.5}$

3.1.2 Odmocnina

Jelikož s vysokými čísly mocninná funkce počítá nepřesně, je ve výpočtu logaritmu použito zjednodušení pomocí odmocniny, dosazením do téhož vzorcem, jako při převodu převáděj í čísel z intervalu $(0;1)$:

$$log_e x^2 = 2 * ln_e \sqrt{x^2}$$

Odmocnina je vypočítaná pomocí Babylonské řady. Výpočet nového členu je daný vzorcem:

$$t_{y+1} = \frac{1}{2} * (\frac{x}{t_y} + t_y), \text{ přičemž } t_0 = 1$$

3.2 Arkus tangens

Pro výpočet funkce arkus tangens jsou potřebné dvě součtové řady. Řady jsou různé pro čísla, jejichž absolutní hodnota je menší než 1, a čísla, jejichž absolutní hodnota je větší než 1. Pro arkus tangens 1 platí, že se rovná $\frac{\pi}{4}$, pro -1 je arkus tangens roven $-\frac{\pi}{4}$

Součtová řada pro $|x| > 1$:

$$\frac{\pi}{2} - \frac{1}{x} + \frac{1}{3x^3} - \frac{1}{5x^5} + \dots$$

Výpočet nového členu z předchozího se získá:

$$t_{y+1} = t_y * (-\frac{i-2}{x^2 * i})$$

Jelikož vynásobením prvního členu, tedy $\frac{\pi}{2}$ by nebyl získán správný tvar druhého členu, je první člen již započítaný mimo součtový rozvoj, a i v iteracích se začíná počítat od $i = 3$.

Počet provedených operací je také možné zredukovat vytvořením proměnné (např. `x2`), ve které bude uložena hodnota výrazu x^2. Není tedy potřeba zapisovat do výpočtu funkce `x*x`

Jelikož s čísly v intervalu $(-1;1)$ součtová řada nepočítá přesně a dochází k zacyklení programu, je potřeba tento interval počítat pomocí jiného rozvoje:

$$x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots$$

Získání nového členu z původního pak vypadá následovně:

$$t_{y+1} = t_y * (-\frac{(i-2) * x^2}{i})$$

První člen součtového rozvoje bude započítán v celkové sumě ještě před cyklem. Z toho důvodu bude v iteracích při počítání funkce v tomto intervalu i opět začínat hodnotou $i = 3$. I zde je také možno použít proměnnou `x2`, k uložení hodnoty `x*x` a snížit tak počet operací v cyklu

3.3 Argument hyperbolického sinu

Tato funkce potřebuje k výpočtu přirozený logaritmus. Pro výpočet logaritmu byl použit stejný vzorec jako u mocninné funkce, s tím rozdílem, že se do funkce odesílala hodnota $2*x$. Výsledek logaritmu je po vypočítání uložen do celkové sumy, spolu s prvním členem součtového rozvoje, který je zde zapsán, jelikož při výpočtu rekurentním vztahem by se vynásobením prvního členu nezískal správný následující člen. Proto je také i číslováno na začátku iterací od 4

$$ln2x + \frac{1!!}{2*(2!!)*x^2} - \frac{3!!}{4*(4!!)*x^4} + \frac{5!!}{6*(6!!)*x^6} + \dots$$

Výpočítání následujícího prvku z předchozího:

$$t_{y+1} = t_y * (-\frac{(i-1)(i-2)}{i^2 * x^2})$$

4 Specifikace testů

Kromě ošetření extrémně malých a vysokých hodnot u výpočtu funkcí je třeba v programu také ošetřit parametry, tedy zajistit, že se budou do funkcí počítat pouze hodnoty, se kterými funkce dokáže počítat.

Například je nutné ověřit, zda na místě kde je možné zadat pouze jeden parametr (konkrétně v části programu, který zajišťuje vypsání nápovědy) je daný parametr zadaný správně, nebo jestli na místě kde je očekávaná číslice není napsaný znak

Test 1: Chybně zadané parametry ⟶ Detekce chyby.

vstup	očekávaný výstup
`./proj2`	Byly nesprávně zapsané parametry.
`./proj2 --help`	Byly nesprávně zapsané parametry.
`./proj2 --argsinh`	Byly nesprávně zapsané parametry.
`./proj2 --powxa -4`	Byly nesprávně zapsané parametry.

Test 2: Chybně zadaná přesnost ⟶ Detekce chyby.

```
7x
-4
```

Test 3: Chybně zadaný exponent mocninné funkce ⟶ Detekce chyby.

```
abc
-42
```

Test 4: Neplatný znak na standartním vstupu ⟶ Program vrátí hodnotu `NaN`

```
aleluja
)
5c5
```

Test 5: Správnost výpočtu ⟶ Předpokládaná správná hodnota.

–powxa vstup	očekávaný výstup
`0^2`	0.0000000000e+00
`-10^10`	NaN
`4^0.5`	2.0000000000e+00
`200^100`	Inf
`NaN^10`	NaN

–arctg vstup	očekávaný výstup
`5`	1.3734007669e+00
`0`	0.0000000000e+00
`-52`	-1.5515679277e+00
`0.5`	-4.6364760900e-01
`NaN`	NaN

–argsinh vstup	očekávaný výstup
`5`	2.3124383413e+00
`-5`	-2.3124383413e+00
`0`	0.0000000000e+00

4.1 Ovládání programu

Program funguje jako konzolová aplikace, má tedy pouze textové ovládání. Program lze spouštět se čtyřmi typy parametrů

parametr	funkce programu
`-h`	vypsání nápovědy
`--powxa`	mocninná funkce
`--arctg`	arkus tangens
`--argsinh`	argument hyperbolického sinu

4.2 Volba datových typů

V projektu je u funkcí použit datový typ `double`, stejně tak u většiny proměnných, se kterými se v nich počítá. Také je zde použit datový typ `boolean` u proměnných, které nábývají hodnot `TRUE` nebo `FALSE`

4.3 Vlastní implementace

V hlavní funkci `main` se testují vstupní parametry, a ověřuje se jejich správnost. Pakliže jsou parametry zadány v souladu s požadavky programu, očekává program vstupní hodnotu, se kterou program výpočítá funkci, která byla zvolena v parametrech programu, jestliže parametry nejsou správně zadané, uživatel je na jejich nesprávnost upozorněn chybovým hlášením. Po vypočítání funkce program opět očekává vstupní hodnotu pro další výpočet. Tento cyklus probíhá, dokud není zadáno `EOF`

5 Závěr

Program slouží k výpočtu tří matematických funkcí - mocninné funkce s realným exponenetem, funkce arkus tangens a argumentu hyperbolického sinu. V parametrech funkce uživatel stanoví, s jakou přesností chce počítat vybranou funkci.

Pro správné vypracování programu bylo nutné pochopit v analytické části princip jednotlivých operací a celkový princip výpočtu. Důležité taktéž bylo správně zvolit řadu pro výpočet funkcí a zjistit, v jakýchh intervalech funkce nejrychleji konvergují

Reference

[1] BARTSCH, H.-J.; HOLFORD-STREVENS, L.: *Matematické vzorce.* Praha: Mladá fronta, třetí vydání, 1996, 831 s., ISBN 80-204-0607-7.

A Metriky kódu

Počet souborů: 1 soubor

Počet řádků zdrojového textu: 494 řádků

Velikost statických dat: 300B

Velikost spustitelného souboru: 9330B (systém Linux, 32 bitová architektura, při překladu bez ladicích informací)