Dokumentace k projektu pro předměty IZP a IUS

Iterační výpočty

projekt č. 2

10. května 2016

Autor: Pavel Frýz, xfryzp00@stud.fit.vutbr.cz
Fakulta Informačních Technologií
Vysoké Učení Technické v Brně

Obsah

1 Úvod

Tato dokumentace popisuje návrh a implementaci programu pro výpočet vybraných matematických funkcí. Jedná se o $\tanh x$, $\log_a x$, vážený aritmetický a kvadratický průměr. V kapitole 2 najdeme seznámení s problémem, v kapitole 3 najdeme řešení jednotlivých problémů, rozsah definičních oborů a specifikaci testů. V kapitole 4 se nachází popis vlastní implementace vycházející z analýzy a návrhu řešení. V poslední kapitole 5 nalezneme celkove zhodnocení programu.

2 Analýza problému

2.1 Zadání problému

Cílem tohoto projektu je vytvoření programu v jazyce C pro výpočet následujích matematických funkcí:

- $y = \tanh x$
- $y = \log_a x$
- Vážený aritmetický průměr
- Vážený kvadratický průměr

Program musí být schopen zpracovat libovolně dlouhou posloupnost číselných hodnot typu `double`, která budou zapsána na standartní vstup. Výstupní data mají být zapsána na standartní výstup.

2.2 Konvergence

Věta o konvergenčním intervalu [1]
Ke každé mocninné řadě $\sum_{k=0}^{\infty} a_k x^k$ existuje takové číslo (tzv. poloměr konvergence) $r > 0$ (připouštíme i $r = +\infty$), že pro všechna čísla x, pro něž $|x| < r$ (tzv. interval konvergence [konvergenční interval]), je mocninná řada absolutně konvergentní a pro všechna čísla x, pro něž $|x| > r$, je divergentní. V bodech x, pro něž $|x| = r$, nelze obecně rozhodnout, zda mocninná řada je konvergentní nebo divergentní. Je-li $r = 0$, pak mocninná řada je konvergentní jen v bodě $x = 0$. Jestliže $r = +\infty$, pak mocninná řada je konvergentní na množině R.

Vydíme, že může nastat problém pokud vstupní hodnota nepatří do konvergenčního intervalu. Hodnoty mimo tento interval musíme vhodně převést do tohoto intervalu, jinak by mohlo dojít k zacyklení programu.

3 Řešení problému

3.1 Definiční obor funkcí

Obor vstupních hodnot je omezen nejen definičním oborem funkcí, ale i použitým datovým typem `double`, který je omezen rozsahem i přesností na počet platných cifer. Datový typ `double` také dokáže reprezentovat hodnoty ∞, $-\infty$ a NaN(Not a Number). Každá matematická funkce, tedy musí kontrolovat jestli vstupní hodnoty spadají do jejich definičního oboru. S ohledem na použitý datový typ je definiční obor pro funkci logaritmus $(0;\texttt{DBL\_MAX}\rangle$, kde DBL_MAX je maximální konečná hodnota datového typu double, obvykle $1.79 \cdot 10^{308}$. Pro hyperbolický tangens $(-(\texttt{DBL\_DIG+REZERVA});\texttt{DBL\_DIG+REZERVA})$, kde DBL_DIG je počet platných cifer datového typu double, obvykle 15. Pro hodnoty mimo tento interval nemá smysl funkce počítat, neboť výsledek by byl vždy 1, případně -1.

3.2 Výpočet hyperbolického tangensu

$$\tanh x = x - \frac{1}{3}x^3 + \frac{2}{12}x^5 - \frac{17}{315}x^7 + \dots \tag{1}$$

$$\tanh x = \frac{\sinh x}{\cosh x} = \frac{x + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \dots}{1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \frac{x^6}{6!} + \dots} \tag{2}$$

$$\tanh x = \frac{e^{2x} - 1}{e^{2x} + 1} \tag{3}$$

Jedna možnost výpočtu hyperbolického tangensu je využití taylorovy řady pro hyperbolický tangens, vzorec 1. Druhou možnost je využít vztahu hyperbolických funkcí, vzorec 2. Třetí možností je vypočítat e^{2x} a použít vzorec 3 vycházející z definice hyperbolického tangensu.

Při bližším prozkoumání všech možností se ukázalo, že první varianta není pro výpočet příliš vhodná, protože vyjádření dalšího člena posloupnosti by bylo příliš složité a neefektivní. Třetí varianta s ohledem na omezení datového typu double není také vhodná, jelikož u čísel blížících se nule by nebylo dosaženo požadované přesnosti. Druhá varianta se ukázala nejvhodnější, protože vyjádření dalšího člena posloupnosti je jednoduché a efektivní. Výhodou také je, že uvedené řady pro hyperbolický sinus a cosinus kovergují pro $x \in R$ narozdíl od řady pro hyperbolický tangens, která konverguje pouze pro $x \in \left(-\frac{\pi}{2}; \frac{\pi}{2}\right)$. Z těchto důvodů jsem se rozhodl pro druhou variantu a tu také implementoval.

3.3 Výpočet logaritmu

$$ln\ x = 2\left(\frac{x-1}{x+1} + \frac{(x-1)^3}{3(x+1)^3} + \frac{(x-1)^5}{5(x+1)^5} + \dots\right) \tag{4}$$

$$log_a x = \frac{ln\ x}{ln\ a} \tag{5}$$

$$ln\ xy = ln\ x + ln\ y \tag{6}$$

$$ln\ x^n = n \cdot ln\ x \tag{7}$$

Pro výpočet $\log_a x$ můžeme použít vzorec 5. Výpočet přirozeného logaritmu poté vypočteme pomocí taylorovy řady pro přirozený logaritmus, vzorec 4. Tato řada konverguje pro $x \in (0; \infty)$.

Při bližším prozkoumání této řady zjistíme, že řada v uvedeném intervalu konverguje různě rychle a že nejlépe konverguje v blízkém okolí bodu $x = 1$. Do tohoto okolí můžeme dostat x využitím vzorců 6 a 7.

3.4 Výpočet váženého aritmetického průměru

$$\bar{x} = \frac{x_1h_1 + x_2h_2 + \ldots + x_nh_n}{h_1 + h_2 + \ldots + h_n} = \frac{\sum\limits_{i=1}^{n} x_ih_i}{\sum\limits_{i=1}^{n} h_i} \tag{8}$$

Výpočet aritmetického průměru provedeme pomocí vzorce 8.

3.5 Výpočet váženého kvadratického průměru

$$\bar{x}_k = \sqrt{\frac{x_1^2h_1 + x_2^2h_2 + \ldots + x_n^2h_n}{h_1 + h_2 + \ldots + h_n}} = \sqrt{\frac{\sum\limits_{i=1}^{n} x_i^2h_i}{\sum\limits_{i=1}^{n} h_i}} \tag{9}$$

Výpočet kvadratického průměru provedeme pomocí vzorce 9.

3.6 Specifikace testů

Jelikož uživatel může zadat chybná data – chybný parametry, nesmyslné hodnoty (znaky, slova, apod.), je potřeba tyto stavy otestovat a ošetřit.

Test 1: Chybná parametry $\longrightarrow$ Detekce chyby.

```
./proj2 --logax text
./proj2 --wam text
./proj2
./proj2 tanh 3
./proj2 --text
```

Test 2: Nesmyslné hodnoty $\longrightarrow$ Detekce chyby, výstup = nan.

```
./proj2 --tanh 14 <<<"0 12text2 0 text 0 0 12text"
```

očekávaný výstup
0.0000000000e+00
nan
0.0000000000e+00
nan
0.0000000000e+00
0.0000000000e+00
nan

Test 3: Správnost výpočtu $\tanh x$ $\longrightarrow$ Předpokládaná správná hodnota.

```
./proj2 --tanh 14
```

vstup	očekávaný výstup
`12.4536`	9.9999999997e-01
`-0.234`	-2.2982054821e-01
`0.234`	2.2982054821e-01
`23e-300`	2.3000000000e-299
`2e303`	1.0000000000e+00
`-inf`	-1.0000000000e+00
`nan`	nan
`0`	0.0000000000e+00

Test 4: Správnost výpočtu $\log_a x \longrightarrow$ Předpokládaná správná hodnota.

```
./proj2 --logax 14 1.423
```

vstup	očekávaný výstup
`13.4536`	7.3681619227e+00
`-3532.234`	nan
`23543254.24`	4.8117693941e+01
`23e-300`	-1.9492736329e+03
`2e303`	1.9797084155e+03
`inf`	inf
`nan`	nan
`0`	-inf

Test 5: Správnost výpočtu aritmetického průměru $\longrightarrow$ Předpokládaná správná hodnota.

```
./proj2 --wam <<<"1 12 2 2 12.321 1 1803.32"
```

očekávaný výstup
1.0000000000e+00
1.1428571429e+00
1.8880666667e+00
nan

Test 6: Správnost výpočtu kvadratického průměru $\longrightarrow$ Předpokládaná správná hodnota.

```
./proj2 --wqm <<<"1 12 2 2 12.321 1 1803.32"
```

očekávaný výstup
1.0000000000e+00
1.1952286093e+00
3.3843467218e+00
nan

4 Popis řešení

Při implementaci jsem vycházel ze závěrů popsaných v předchozích kapitolách.

4.1 Ovládání programu

Program funguje jako konzolová aplikace, má tedy pouze textové ovládání. Při spouštění programu s parametr `-h` vypíše obrazovku s nápovědou. Další možnost spuštění je s parametrem `--wam` nebo `--wqm`, při tomto spuštění se počítá vážený aritmetický průměr nebo vážený kvadratický průměr. Při spuštění s parametrem `--tanh` nebo `--logax` očekavá program ještě druhý parametr, a to přesnost na počet platných cifer. A v případě výpočtu logaritmu očekává ještě jeden parametr reprezentující základ logaritmu.

Po spuštění s těmito parametry program očekává na standardním vstupu hodnoty typu `double` oddělené posloupností bílých znaků. Každá hodnota je předána zvolené funkci a výsledek operace je vytisknut na standartní výstup v zadaném formátu.

Výhodou tohoto ovládání je, že program může být použít ve skriptech a jím produkovaný výsledek může být použit jiným programem pro další výpočet.

4.2 Volba datových typů

Pro uložení hodnot výsledku jsem zvolil datový typ `double`. Pro uložení hodnot pro výpočet vážených průměrů slouží struktura `TAverage`, která obsahuje čtyři položky typu `double` pro součet hodnot, součet vah, aktualní hodnotu a aktuální váhu.

4.3 Vlastní implementace

Parametry příkazové řádky zpracovává funkce `getParams`, která je spouštěna jako první ve funkci `main`. Načtené parametry se poté předají funkci `calculate`, která v cyklu pomocí funkce `scanf` načítá vstupní data a pomocí funkce `testInput` detekuje případné chyby. Poté je podle parametrů zavolaná funkce pro výpočet zvolené funkce. Výsledek je poté vytisknut pomocí funkce `printf` ve zvoleném formátě.

Pro výpočet váženého kvadratického průměru slouží funkce `wam`, která má jediný parametr strukturu `TAverage`, která obsahuje součet hodnot, součet vah a aktualní hodnotu a váhu. Z těchto hodnot vypočte aritmetický průměr a příslušně změní součet hodnot a vah. Velice podobně funguje funkce `wqm` pro výpočet kvadratického aritmetického průměru. Pro výpočet hyperbolického tangensu slouží funkce `mytanh`. Výpočet $\log_a x$ zajišťuje funkce `logax`. Funkce prvně vypočítá pomocí funkce `ln` přirozený logaritmus pro základ a hodnotu. A tyto hodnoty poté podělí podle vzorce 4.

5 Závěr

Výsledný algoritmus počítá s poměrně velkou přesností. Zvolené algoritmy počítají co nejefektivněji. Program skončí i pro mezní hodnoty v rozumném čase.

Navržené řešení je přenositelné na všechny platformy, které podporují datový typ `double`. Při přenosu na platformu, která tento typ nepodporuje, by bylo nutné upravit datové typy.

Program striktně dodržuje formát vstupních a výstupních dat, díko tomu je možné využití ve skriptech nebo spolupráce s jinými programi.

Reference

[1] BARTSCH, Hans-Jochen: *Matematické vzorce*. Praha: Mladá fronta, 1996, ISBN 80-204-0607-7.

A Metriky kódu

Počet souborů: 1 soubor

Počet řádků zdrojového textu: 687 řádků

Velikost statických dat: 7597B

Velikost spustitelného souboru: 12164B (systém Linux, 64 bitová architektura, při překladu bez ladicích informací)