Dokumentácia k projektu pre predmety IZP a IUS

Iteračné výpočty

projekt č.2

17.november 2012

Autor: Ladislav Šulák , xsulak04@stud.fit.vutbr.cz

Fakulta Informačných Technológií

Vysoké Učení Technické v Brne

Obsah

1 Úvod

V tomto dokumente je analyzovaný návrh , implementácia a problematika troch matematematických funkcií: mocninová funkcia s reálnym exponentom , a prirodzeným základom , funkcia pre výpočet arkus tangensu a funkcia pre výpočet argumentu hyperbolického sínusu.

Pri výpočtoch sa využívajú iteračné algoritmy.V programovaní sú obzvlášť dôležité kvôli ich využitiu při počítaní rôznych matematických funkcií.

Tento dokument sa skladá z viacerých častí. Kapitola č.2 je venovaná analýze problémov , ktoré sú spojené s danými matematickými funkciami a načrtnutiu riešenia. Kapitola č.3 sa zaoberá algoritmom pre riešenie daných funkcií, a tiež spôsobom ako ukončiť iteračný výpočet. Kapitola č.4 je venovaná špecifikácií testov,ktoré overujú funkčnosť samotného programu. Kapitola č.5 obsahuje vlastnú implementačnú metódu ktorá bola použitá , a tiež popis ovládania programu.

2 Analýza problému a princíp jeho riešenia

Pre výpočty boli použité iba základné matematické operácie: súcet , rozdiel ,súčin a podiel. V programe je prítomná knižnica <math.h> , no využívajú sa z nej iba makrá NAN a INFINITY. Všetky ostatné funkcie z tejto knižnice ,vrátane funkie pre výpočet absolútnej hodnoty sú implementované samostatne.

2.1 Zadanie projektu

Cieľom tohto projektu je vytvoriť program v jazyku C , ktorý vypočíta tri matematické funkcie- mocninová funkcia s reálnym exponentom,a prirodzeným základom, funkcia pre výpočet arkus tangensu, a funkcia pre výpočet argumentu hyperbolického sinu.Program počíta s presnosťou, ktorá sa nastavuje pri spustení programu zadaním parametru. Program načíta čísla zo štandardného vstupu, a vypíše ich na štandardný výstup.

2.2 Stručná charakteristika riešenia

Pri výpočte daných matematických funkcií som vychádzal z Taylorovho rozvoja. Taylorov rozvoj je postupnosť, ktorá konverguje iba na úzkom intervale hodnôt v rámci definičného oboru funkcie. Pri určitých hodnotách (resp. intervale) konverguje rýchlejšie a pri určitých hodnotách pomalšie. Bolo treba zaviesť proti tomu opatrenia-heuristiku- ktorá viedla k rýchlejším a presnejším výsledkom.Algoritmus musí byť efektívny - nemal by obsahovať zbytočné výpočty ,najmä vnútri cyklov, ktoré spomalujú priebeh konečného výpočtu.

2.3 Počet platných číslic

Počet platných číslic(significant digits) sa zadáva ako tretí parameter-sigdig (prvý je názov súboru, druhý je matematická funkcia ktorá sa má vykonať, ale o tom podrobnejšie v kapitole č.5).Je to vlastne číslo ktoré udáva počet platných cifier konečného výsledku.Pokiaľ počítame na zadaný počet platných číslic, znamená to, že nás zaujímajú iba tie.Napríklad ak bude počet platných číslic 10, tak hodnota jedenástej čislice nás nezaujíma.Aplikovanie sigdigu nájdeme v kapitole 3.5.

3 Návrh riešenia problému

3.1 Výber dátových typov, definičný obor

Zváženie aké dátové typy použiť je veľmi dôležité. Pri použití nesprávnych môže dôjsť k implicitnému pretypovaniu alebo zlému zaokrúhleniu čo vedie k nesprávnemu(skreslenému) výsledku. Ďalší faktor ktorý musíme zohľadniť je rozsah datových typov.Rozhodol som sa skoro všade používať dátový typ *double* , ktorí má presnosť na 15 desatinných miest (číslo 15 z tohto dôvodu bude maximálne číslo aké može dosiahnuť parameter sigdig).Typ *double* može vrátiť aj nečíselné hodnoty NAN (not a number) , a INF (infinity), ktoré nájdu uplatnenie pri zlom zadaní vstupných hodnôt.

3.2 Arkus tangens

Pre výpočet arkus tangens je výhodné použiť Taylorov rozvoj, ktorý konverguje na intervale <-1, 1> rýchlo, a počíta presne.

Vzorec pre výpočet arkus tangensu (Taylorov rozvoj) :

$$\mathrm{arctg(x)} = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \ \dots = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{2n+1} \ \mathrm{pre\ x} < -1{,}1 >$$

Ak by hodnota premennej x (vstupná hodnota) väčšia než 1 alebo menšia než -1 , výsledok by bol nepresný, a so zväčšujúcimi sa hodnotami pomalý a ešte nepresnejší. Treba urobiť jedno opatrenie ,ktoré zabezpečí že funkcia arctg bude počítať na celom definičnom obore rýchlejšie a presnejšie.

Použijeme nasledujúci vzorec:

$$arctg(x) = \frac{\pi}{2} - arctg(\frac{1}{x})$$

Posledná časť heuristiky: pre x<0 . Vychádzame z poznatku: arkus tangens je nepárna funkcia, teda výsledné hodnoty pre x<0 budú mať opačné znamienko ako výsledné hodnoty pre x>0.

$$arctg(-x) = -\frac{\pi}{2} + arctg(\frac{1}{|x|})$$

3.3 Argument hyperbolického sínusu

Pre výpočet danej funkcie je výhodné použiť nasledujúci vzorec:

$$argsinh(x) = \ln(x + \sqrt{x^2 + 1})$$

Vo vzorci je použitá funkcia pre výpočet prirodzeného logaritmu, a tiež použitá odmocnina.

Implementovať vlastnú funkciu na výpočet odmocniny je zbytočné. Vychádzame zo základného vzorca pre počítanie s odmocninami:

$$\sqrt[2]{x} = x^{\frac{1}{2}}$$

Po implementácií mocninovej funkcie teda nieje nutné použiť odmocninu.

3.3.1 Prirodzený logaritmus

Prirodzený logaritmus nájde využitie v mocninovej funkcii, a tiež vo funkcii argumentu hyperbolického sínusu. Implementujeme teda vlastnú funkciu pre výpočet prirodzeného logaritmu, pomocou Taylorovho rozvoja:

$$\ln(x) = 2\left[\frac{x-1}{x+1} + \frac{(x-1)^3}{3(x+1)^3} + \frac{(x-1)^5}{5(x+1)^5} + \cdots\right]$$
$$= 2\sum_{n=1}^{\infty}\frac{1}{2n-1}\left(\frac{x-1}{x+1}\right)^{2x-1} \quad pre\ x > 0$$

Logaritmovať záporné číslo sa nedá, a logaritmus čísla 0 nieje definovaný.Daný vzorec konverguje v intervale (1,2) rýchlo a počíta presne,no akonáhle zadáme číslo mimo tohto intervalu, dostaneme nepresnejší výsledok.Heuristika sa bude skladať z dvoch častí: pre x>2 a pre x<1.

Pre x>2 vychádzame zo základných vzorcov pre počítanie s logaritmami:

$$\ln(x * y) = \ln(x) + \ln(y)$$

Pre x z intervalu (0,1) vychádzame zo základného vzorca pre počítanie s logaritmami:

$$\ln\left(\frac{x}{y}\right) = \ln(x) - \ln(y)$$

Úprava vstupnej hodnoty (premennej x). Bez tejto heuristiky by sa pri vyšších hodnotách počítalo nepresnejšie a pomalšie.

3.4 Mocninová funkcia s prirodzeným exponentom

Pre výpočet mocninovej funkcie s prirodzeným exponentom je výhodné použiť Taylorov rozvoj.

Vzorec na výpočet mocninovej funkcie:

$$x^a = 1 + \frac{\mathrm{a}\,.\ln(x)}{1!} + \frac{a^2 .\, ln^2(x)}{2!} + \frac{a.\, ln^3(x)}{3!} + \cdots$$
$$= \sum_{n=0}^{\infty} \frac{(a.\ln(x))^n}{n!} \;\; pre\ x > 0\,, a \in R$$

Po aplikovaní tohto vzorca nebudeme môcť počítať so záporným základom,avšak to je nad rámec tohto projektu.

3.5 Ukončenie výpočtu.

Pri výpočtoch som využíval Taylorov rozvoj, čo je nekonečná postupnosť.Keby nebola použitá ukončovacia podmienka, program by sa zacyklil.

Výpočet sa skončí, ak je absolútna hodnota rozdielu dvoch po sebe idúcich členov postupnosti väčšia ako presnosť ε. Jedná sa o absolútnu chybu,ktorá sa mení každým výpočtom-každou iteráciou.Výpočet presnosti ε vypočítame nasledujúcim spôsobom:

$$\varepsilon = 0.1^{\mathrm{sigdig}}$$

Exponent, sigdig bol spomenutý v kapitole 2.3.

4 Špecifikácia testov

Nie vždy sa stáva, že užívateľ zadá správny parameter, správnu vstupnú hodnotu, a s týmto treba počítať.Aby sme sa vyhli tomu, že program spadne alebo sa zacyklí, musíme program ošetriť aby bol pripravený aj na neočakávané situácie.

Test č.1 : Zlé zadané parametre(chýba sigdig)

$proj2 --argsinh <<<2

Výsledok : Chybové hlásenie,zlý parameter. Program sa nespustí.

Test č.2: Žiadny parameter

$proj2

Výsledok:Chybové hlásenie,žiadny parameter.Program sa nespustí.

Test č.3:Nesprávny parameter č. 3 (sigdig)

$proj2 --arctg a12 <<<2

Výsledok:Chybové hlásenie,Zadali ste nesprávny parameter.Program sa nespustí.

Test č.4:Nesprávna forma vstupného argumentu

$proj2 --argsinh 2 <<<2a

Výsledok:Program vypočíta danú funkciu so vstupným číslom 2, písmeno „a“ ignoruje.

Test č.5:Extrémne hodnoty argumentov

$proj2 --arctg 15 <<<1e-100

Výsledok:1.0000000000e-100

Test č.6:Extrémne hodnoty argumentov

$proj2 --powxa 11 2.0 <<<1e-100

Výsledok:1.0000000000e-200

Test č.7:Extrémne hodnoty argumentov

$proj2 --powxa 11 2.0 <<<1e300

Výsledok:inf

Test č.8:Mocninová funkcia:záporný základ

$proj2 --powxa 11 2.0 <<<-10

Výsledok:nan

Test č.9:Test zameraný na presnosť výsledku

$proj2 --argsinh 15 <<<40

Výsledok: **4.3821828481e+00**

Výsledok http://www.wolframalpha.com/

4.382182848065498

Výsledky sa líšia až na desiatom desatinnom mieste.

5 Popis riešenia

5.1 Ovládanie programu

Program je vlastne konzolová aplikácia.Parametry programu sa vkladajú pri spustení programu,a sú nasledovné:

-h alebo --help	zobrazí nápovedu
--powxa sigdig a	vypočíta mocninu čísla (vstupný argument)
	a=exponent , sigdig =počet platných číslic
--arctg sigdig	vypočíta arkus tangens čísla (vstupný argument)
--argsinh	vypočíta argument hyperbolického sínusu čísla (vstupný argument)

Príklad zadania vstupného argumentu v terminály pod systémom Linux (napríklad číslo 2):

--argsinh <<<2

Vstupné hodnoty sa zapisujú na štandardný vstup, Výstupné hodnoty sa zapisujú na štandardný výstup, a chybové hlásenia sa zapisujú na štandardný chybový výstup.

5.2 Implementácia

Parametry príkazového riadku sa spracúvajú priamo vo funkcii main.Pokiaľ by pri spracúvaní došlo k nejakej chybe, vypíše sa chybové hlásenie.Chybové hlásenia spolu s konštantami, a s funkciou pre nápovedu sú umiestnené na začiatku programu.Každá matematická funkcia je implementovaná zvlášť.

V prípade že sa zadajú správne parametre, vloží sa vstupný argument pomocou funkcie fscanf() , a zavolá sa príslušná funkcia. Po zavolaní sa výpočet uskutoční pomocou cyklu, ktorý končí za podmienky ktorú som uviedol kapitole 3.5. Potom sa výsledok zobrazí na štandardný výstup pomocou funkcie fprintf().

6 Záver

Program úspešne rieši všetky tri matematické funkcie. Moje funkcie nie sú tak efektívne ako matematické funkcie z knižnice <math.h> , avšak výsledky sú viac než uspokojivé. Výpočty sú rýchle a takmer úplne presné. Podarilo sa mi implementovať program, který je prenosný na 32-bitových architektúrach, bol testovaný pod operačnými systémami Windows a Linux.

A Použitá literatura

[1] Bartsch, Hans-Jochen - Matematické vzorce, Praha: Academia, 2006, ISBN 80-200-1448-9

[2] Herout, Pavel – Učebnice jazyka C , České Budějovice, Kopp, 2011, ISBN 80-7232-383-8

B Metriky kódu

Počet súborov: 1

Počet riadkov zdrojového textu: 309

Veľkosť statických dát: 344B

Veľkosť spustitelného súboru : 16112B (Systém Linux(Ubuntu 12.04 LTS) , 32-bitová architektúra)