Dokumentace k projektu pro předměty IZP a IUS

Iterační výpočty

projekt č. 2

18. listopadu 2012

Autor: Mark Birger, xbirge00@stud.fit.vutbr.cz
Fakulta Informačních Technologií
Vysoké Učení Technické v Brně

Obsah

1 Úvod

Výpočet matematických funkcí, které vedou k iracionálních čísel je docela zajímavý problém z hlediska informačních technologií. Pri vytvoření takových algoritmů, hlavním cílem je optimizace algoritmusu, jakož i vymezení přesnosti. V tomto úkolu musim získat co nejvíce efektivní vypořádání u předurčeného přesností.

Vypočtené zde funkce (mocninná funkce, arctg(), arsinh()) jsou používány v mnoha oblastech vědy - fyzika, matematika, některé oblasti informačních technologií. Proto, zkušenosti psaní těchto algoritmů je velmi vysoký.

Dokumentace se skládá ze tří hlavních částí: analýza problému (kapitola 2), pojem řešení (kapitola 3) a popis na rozhodnutí společně s testováním (kapitola 4 a 3.4). Rovněž v tomto dokumentu, existují další témata, jako je toto. Kromě toho, každá kapitola obsahuje podtémata, která poskytuje podrobnější informace o jednotlivých prvcích problému.

2 Analýza problému a princip jeho řešení

Nejprve musíme pochopit, jaké hodnoty jsou akceptovány pro tyto funkce, a to, co se tyto funkce vrací. Tato kapitola obsahuje analýzu dat pro konkrétní funkce.

2.1 Zadání problému

Účelem tohoto úkolu - naučit počítat hodnotu funkce pomocí nekonečných řad, výpočet iracionální čísla s definovanou přesností. Konkrétně, vstup programu přijme přesnost a argument, ktery definuje jednu ze tří funkcí: mocninna funkce, arcus tangens, argument sinus hyperbolicky. V případě mocninny funkcí se další parametr exponent. Potom program čte posloupnost racionálních čísel, oddělených bílými znaky, zobrazí se pro každý přečíst hodnoty dané funkce s zadanou presnosti.

2.2 Analýze funkce

Mocninna funkce:

$$y = a^x \tag{1}$$

Tato funkce definovaná pro každou hodnotu x a a (nekonečno je také určité hodnoty). Poprve budeme pracovat s funkcí pro kladné hodnoty x a a, postupně rozšiřuje rozsah dostupných hodnot.

Poznámka: Pro zjednodušení povolené vracet NaN, kdyz x menší než nula.

Arcus tangens:

$$y = arctg(x) \tag{2}$$

Tato funkce může mít jakoukoli hodnotu x , ale vrací pouze hodnoty v tomto rozsahu:

$$arctg(x) \in (-\frac{\pi}{2}; \frac{\pi}{2}) \tag{3}$$

Argument sinus hyperbolicky:

$$y = arsinh(x) \tag{4}$$

Funkce je definována na celém intervalu racionálních čísel. Hodnota funkce může být libovolné číslo.

2.3 Orientační řešení

Výpočet funkcí se provede pomocí nekonečných řad. Také musíme optimizovat příchozí data, zjednodušením vzorce platnych pro tyto funkce. Kromě toho je třeba pomocné funkce, například přirozený logaritmus a odmocnina z čisla. O pomocnych funkcich si můžete přečíst v další části. Pro více informací o tom, jak jde výpočet funkčních hodnot, naleznete v příští kapitole.

2.4 Pomocné funkce

Druhá odmocnina:

$$y = \sqrt{x} \tag{5}$$

Výpočet odmocniny provádí pomocí Newtonovy metody. To je jeden z nejefektivnějších algoritmů pro výpočet odmocniny. Algoritmus je graficky zobrazen níže:

$\sqrt{2}$	Podíl	Aritmetický průměr
1	$\frac{2}{1} = 2$	$\frac{2+1}{2} = 1.5$
1.5	$\frac{2}{1.5} = 1.333$	$\frac{1.333...+1.5}{2} = 1.4167$
1.4167	$\frac{2}{1.4167} = 1.4118$	$\frac{1.4167+1.4118}{2} = 1.4142$

Přesnost algoritmu je vysoká vzhledem k tomu, že rozdíl ve změnách spadá rychle. Používá se stejné přesnosti jako u volající funkce.

Přirozený logaritmus:

$$y = log(x) \tag{6}$$

Výpočet přirozeného logaritmu je potřeba pro výpočet mocninny funkci a pro výpočet hyperbolického sinusu argumentu. Zpočátku, vstup musi být usnadněn podle následujícího vzorce:

$$log(\frac{x}{e}) = log(x) - 1 \tag{7}$$

rovnocenně:

$$log(x * e) = log(x) + 1 \tag{8}$$

Tam pouzita konstanta e. To staví některých omezení na přesnost - 20 číslic. Pro zpřesnění hodnot lze dynamicky vypočítat počet e s definovanou přesností. Pokud je číslo větší než jedna, pak by měla být rozdělena na e v cyclu, dokud to bude méně než jeden. Pokud je to méně než jeden násobte na e v cyclu, dokud je tak blízko k jedničce (zvyšuje rychlost a přesnost výpočtů). V době těchto operací je sčítač. Výpočet by měl být proveden s Taylorovy řady (taky Merkatoruva řada):

$$log(1 + x) = \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n} * x^n = x - \frac{x^2}{2} + \frac{x^3}{3} ... \qquad for\ |x| \leq 1 \tag{9}$$

Je však lepší použít již převeden formuli Taylorovy řady:

$$log(x) = (x-1) - \frac{(x-1)^2}{2} + \frac{(x-1)^3}{3} - \frac{(x-1)^4}{4} ... \quad for\ |x-1| \leq 1\ unless\ x = 0 \tag{10}$$

Přirozený logaritmus je počítán se stejnou přesností jako pomocí jeho funkce.

Také třeba poznamenat, že logaritmus nejistoty na hodnoty X menší než nula, a na nule, je roven záporné nekonečno. Tento problém bude vyřešen jinými funkcemipřed volání funkce logaritmu.

2.5 Problém přesnosti

Problém přesnosti jeden z nejdůležitějších problémů při řešení problémů tohoto typu. Pomocí danneho počtu přesných čísel za desetinnou čárkou by měla být stanovena přesnost vypoctu. V případě konvergentní řady - je to docela jednoduché. Přesnost se vypočítá pomocí tohoto vzorce:

$$\epsilon = 10^{-(sigdig+1)} \tag{11}$$

Pokud je počet jenom se zvyšuje nebo snížuje jediný způsob, jak vypočítat číslo s dostatečnou přesností - je zvýšit přesnost předchozí počáteční vzorce. **V obou případech, je přesnost definována jako rozdíl mezi předchozí a další iteraci.**

3 Návrh řešení problému

Tato kapitola se bude diskutovat o aktuální algoritmy počítání funkce, testy algoritmu, a další aspekty algoritmů.

Poznámka: Pomocné funkce v této kapitole nebudou považovány, že budou brát za samozřejmost. Pro více informací o nich naleznete v sekci 2.4.

3.1 Struktura programu

Program se skládá z funkcí, které jsou psány oddělené prvky a mohou být použity nezávisle na sobě. Funkce *main()* trvá jen hodnoty a vytváří streamov vstup. Pro každe uznáne racionalní číslo se vola funkce. Některé funkce se vztahují na pomocné funkce.

3.2 Výpočet hodnot

Mocninna funkce: Možná nejjednodušší řešení, které okamžitě přijde na mysl by jednoduchá smyčka s násobením, ale budeme pracovat s touto funkcí (stejně jako se všemi ostatními) v oblasti racionálních čísel. Rada pro spočítani mocninne funkce je uvedena níže:

$$x^a = 1 + \frac{a * log(x)}{1!} + \frac{a^2 * log^2(x)}{2!} + \frac{a^3 * log^3(x)}{3!} + ... = \sum_{n=0}^{\infty} \frac{(a * log(x))^n}{n!} \tag{12}$$

Každá iterace algoritmu bude násobit predchozi člen řady na a a na *log(x)* a dělit na cislo iteraci.

$$Y_{i+1} = \frac{Y_i * a * log(x)}{i} \tag{13}$$

Pro výpočet této řady vyžaduje pomocnou funkci logaritmusu. Nyní se snaží rozšířit rozsah hodnot, ze kterých naše funkce. Nejprve v případě záporného exponentu, musíme sdílet 1 na základ mocniny a pracovat s absolutnou hodnotou exponenty. Kromě případů, kdy jeho základní číslo je nula.

$$x^{-a} = \frac{1}{x^a} \tag{14}$$

Pro zjednodušení algoritmu, pokud základ stupně nula, a index kladný, budeme ihned vrátít tu nulu.

$$0^a = 0 \qquad for \ a > 0 \tag{15}$$

Poznámka: Je-li základna je nula, a index je negativní, algoritmus sám považuje za negativní nekonečno.

$$0^a = -\infty \qquad for \ a < 0 \tag{16}$$

Velmi malé čísla jsou převedeny pomocí vzorce:

$$(\frac{1}{x})^a = \frac{1}{x^a} \qquad for \ 0 < x < 1 \tag{17}$$

Všechny tyto optimizace po výpočtu konvertuji stupně znovu.

Arcus tangens: Pro spocitani *arctg()* pouzivam Tayloruvu radu. Rada pro spočítani mocninne funkce je uvedena níže:

$$arctg(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + ... = \sum_{n=0}^{\infty}(-1)^n \frac{x^{2n+1}}{2n+1} \qquad for \ x \in < -1; 1 > \tag{18}$$

Optimizace, kdyz *x¡0* takova:

$$arctg(-x) = -arctg(x) \tag{19}$$

Navíc, musíme poskytnout příchozí hodnoty v interval pro ktery plati Tayloruva rada. Program dela to podle vzrocu:

$$arctg(\frac{1}{x}) = \frac{\pi}{2} - arctg(x) \tag{20}$$

Avšak to je problém výpočet hodnoty arctangensu jedničcy. V tomto případě program jednoduše vráti konstantu $\frac{\pi}{4}$ (povoleno používání).

Argument sinus hyperbolicky: Vzorec pro výpočet této funkce, jeden z nejjednodušších, protože to závisí na výpočtu přirozeného logaritmu.Program používá tuto funkci:

$$arsinh(x) = log(x + \sqrt{1 + x^2}) \tag{21}$$

Zjednodušení v této situaci je pouze toto:

$$arsinh(-x) = -arsinh(x) \tag{22}$$

3.3 Analýza vstupních dat

Příchozí hodnoty jsou libovolné diki tomu, že funkce sami odpovídaji za špatné příchozí hodnoty. Výjimkou jsou hodnoty znaků. Pokud je vstup řetězec vytýká, že hodnota funkce tohoto argumentu není definována. Avšak jestliže je vstupní by bylo něco takoveho *12ABSC*, program spočítá cislo *12*. Je to způsobeno použitím funkce scanf().

3.4 Specifikace testů

Testování programu bude probíhat pri různých funkcích. Program se zobrazí číslo pomocí mantisy. Také třeba poznamenat, že program je sestaven takto:

Compilace:

```
gcc -srd=c99 -Wall -pedantic -Wextra -g - o proj2 proj2.c -lm
```

Test 1: powxa 11 2.3

vstup	výstup	pravidelny výstup
`1`	1.0000000000e+00	1.0000000000e+00
`40`	4.8388034325e+03	4.8388034325e+03
`1e-100`	1.0000000000e-230	1.0000000000e-230
`1e300`	inf	inf
`-5`	nan	nan

Test 2: arctg 11

vstup	výstup	pravidelny výstup
`0`	0.0000000000e+00	0.0000000000e+00
`1.3`	9.1510070055e-01	9.1510070055e-01
`-5.6`	-1.3940874707e+00	-1.3940874707e+00
`50`	1.5507989928e+00	1.5507989928e+00
`asfd`	nan	nan

Test 3: argsinh 11

vstup	výstup	pravidelny výstup
`0`	0.0000000000e+00	0.0000000000e+00
`1`	8.8137358701e-01	8.8137358701e-01
`-5`	-2.3124383412e+00	-2.3124383412e+00
`40`	4.3821828480e+00	4.3821828480e+00
`as`	nan	nan

4 Popis řešení

Při implementaci jsem vycházel ze závěrů popsaných v předchozích kapitolách. V podstatě všechny vzorců uvedeny výše.

4.1 Ovládání programu

Program funguje jako konzolová aplikace, má tedy pouze textové ovládání. Při spouštění program reaguje na nekolil parametru. Dva z nich (*-h* , *–help*) umožní program zobrazít nápovědu.

Při použití programu pro výpočet funkci, musíte nejprve určit, které funkce by měly být brány v úvahu. To definuje jeden ze tří argumentů: *–powxa* , *–arctg* , *–arsinh* . Další parametr, který je nastaven při spuštění programu je přesnost. Je definován přírodní nezáporné. Pokud je tento parametr chybí, program zobrazí chybu. Třetí parametr je nutné pouze tehdy, pokud chcete vypočítat mocninnu funkci, pro ni je to potřeba. Teto parametr je racionalni cislo. Pokud program bude zahájen s více argumenty, než je požadováno, je to všechno stejné bude fungovat správně.

Po spuštění programu se zacina streamovy vstup. Program současně výpise hodnoty funkce. Stream bude fungovat dokud uživatel nezadá EOF.

4.2 Volba datových typů

Nejvíce často, program používá datovy typ *double*. Také v cyklech, a jako čítač typ *int*. Pro logické operaci program používá datovy typ *Bool*.

5 Závěr

Na konci jsme dostali program, který pracuje správně a může být použit v mnoha oblastech vědy. Navíc, je-li více složité výpočty portebuji stejni vypocet, funkce softwaru může být použita pro zjednodušení algoritmu. Funkční programování vypadá skvěle. Definované přesnosti výpočtu může výrazně snížit zatížení v řadě úkolů.

Program byl testovány na serverech FIT a počítači s Mac OS. Kód je cross-platformní a nebude těžké portovat program pod Windows.

Reference

[1] Hazewinkel, Michiel, ed. (2001), "Taylor series", *Encyclopedia of Mathematics*, Springer, ISBN 978-1-55608-010-4

A Metriky kódu

Počet funkci: 7 funkci

Počet řádků zdrojového textu: 321 řádků

Velikost kódu programu: 6968B

Velikost spustitelného souboru: 15220B (systém Mac OS X, 32/64 bitová architektura, při překladu bez ladicích informací)