Projekt 2 - Iterační výpočty

Popis projektu

Implementujte výpočet přirozeného logaritmu a exponenciální funkce s obecným základem pouze pomocí matematických operací +,-,\*,/.

Detailní specifikace

**Překlad a odevzdání zdrojového souboru**

Odevzdání: Program implementujte ve zdrojovém souboru proj2.c. Zdrojový soubor odevzdejte prostřednictvím informačního systému.

Překlad: Program překládejte s následujícími argumenty:

$ gcc -std=c99 -Wall -Wextra -Werror proj2.c -lm -o proj2

**Syntax spuštění**

Program se spouští v následující podobě:

./proj2 --log X N

nebo

./proj2 --pow X Y N

Argumenty programu:

* --log X N požadavek pro výpočet přirozeného logaritmu z čísla X v N iteracích (Taylorova polynomu a zřetězeného zlomku).
* --pow X Y N požadavek pro výpočet exponenciální funkce z čísla Y s obecným základem X v N iteracích (Taylorova polynomu a zřetězeného zlomku)

**Implementační detaily**

Je zakázané použít funkce z matematické knihovny. Jedinou výjimkou jsou funkce log a pow použité pouze pro srovnání výpočtů, funkce fabs (kvůli své trivialitě), funkce isnan a isinf a konstanty NAN a INFINITY. Ve všech výpočtech používejte typ double.

**Implementace logaritmu**

Funkci logaritmu implementujte dvakrát a to pomocí Taylorova polynomu a zřetězených zlomků.

**1. podúkol - Implementace Taylorova polynomu**

Logaritmus pomocí Taylorova polynomu implementujte ve funkci s prototypem:

double taylor\_log(double x, unsigned int n);

kde n udává rozvoj polynomu (počet členů). Taylorův polynom pro funkci logaritmu implementujte podle vzorce:



pro 0 < x < 2 a



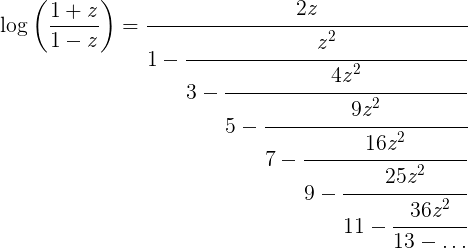
pro x > 1/2. Doporučená mezní hodnota mezi těmito dvěma polynomy je 1.

**2. podúkol - Implementace zřetězeného zlomku**

Logaritmus pomocí zřetězených zlomků (viz demonstrační cvičení) implementujte ve funkci s prototypem:

double cfrac\_log(double x, unsigned int n);

kde n udává rozvoj zřetězeného zlomku. Funkci implementujte podle vzorce:



**3. podúkol - Výpočet exponenciální funkce s obecným základem**

Exponenciální funkci s obecným základem počítejte ve funkci s prototypem:

double taylor\_pow(double x, double y, unsigned int n);

a

double taylorcf\_pow(double x, double y, unsigned int n);

kde n udává rozvoj polynomu (počet členů) a parametry x a y odpovídají parametrům funkce pow z matematické knihovny. Taylorův polynom pro exponenciální funkci implementujte podle vzorce:



pro a > 0.

Pro výpočet přirozeného logaritmu použijte funkci taylor\_log v případě funkce taylor\_pow a funkci cfrac\_log v případě funkce taylorcf\_pow.

**Výstup programu**

V případě výpočtu logaritmu (argument --log) program tiskne následující řádky:

log(X) = LOG\_X

cfrac\_log(X) = CFRAC\_LOG\_X

taylor\_log(X) = TAYLOR\_LOG\_X

V případě výpočtu exponenciální funkce (argument --pow) program tiskne následující řádky:

pow(X,Y) = POW

taylor\_pow(X,Y) = TAYLOR\_POW

taylorcf\_pow(X,Y) = TAYLORCF\_POW

kde:

* X a Y jsou hodnoty zadané argumentem příkazové řádky (odpovídají formátu printf %g),
* LOG\_X je hodnota logaritmu z matematické knihovny,
* CFRAC\_LOG\_ jsou hodnoty logaritmu vypočteného pomocí zřetězeného zlomku,
* TAYLOR\_LOG\_ jsou hodnoty logaritmu vypočteného pomocí Taylorova polynomu,
* POW je hodnota exponenciální funkce z matematické knihovny,
* TAYLOR\_POW je hodnota vypočtená pomocí funkce taylor\_pow,
* TAYLORCF\_POW je hodnota vypočtená pomocí funkce taylorcf\_pow,
* všechny \*LOG\_\* a \*POW hodnoty odpovídají formátu %.12g.

**Příklady vstupů a výstupů**

*Číselné údaje nemusí přesně odpovídat vaší implementaci. Výsledek závisí na způsobu implementace a optimalizaci.*

$ ./proj2 --log 1.131401114526 4

log(1.1314) = 0.123456789012

cfrac\_log(1.1314) = 0.123456789012

taylor\_log(1.1314) = 0.123452108537

$ ./proj2 --pow 1.23 4.2 4

pow(1.23,4.2) = 2.38562110403

taylor\_pow(1.23,4.2) = 2.38026034593

taylorcf\_pow(1.23,4.2) = 2.38079698151

Hodnocení

Na výsledném hodnocení mají hlavní vliv následující faktory:

* implementace algoritmických schemat pro iterační výpočty,
* výpočet logaritmu a exponenciální funkce,
* ošetření neočekávaných stavů.

**Prémie**

Prémiové body (max 4) je možné získat implementací alternativní funkce k funkcím log a pow. Alternativní funkce budou implementovány v prototypech:

double mylog(double x);

a

double mypow(double x, double y);

Obě funkce budou podle hodnoty zadaného argumentu volit nejpřesnější typ výpočtu (Taylorův polynom nebo zřetězené zlomky) a minimální počet iterací pro požadovanou přesnost. Nechť požadovaná přesnost je na 8 významných číslic (tj. odpovídá výstupnímu formátu %.7e přesného výsledku). Podmínkou pro udělení prémiových bodů je úspěšná obhajoba projektu a prémiového vypracování.