



SVEUČILIŠTE U ZAGREBU

Fakultet
elektrotehnike i
računarstva

PARALELIZAM I KONKURENTNOST

183376

1. laboratorijska vježba

Ak. god. 2024./2025.

1 Integracija metodom Monte Carlo

1.1 Opis problema

Metode numeričke integracije obuhvaćaju velik skup metoda za numerički (približni) izračun integrala funkcija. Izbor metode integracije ovisi od svojstvima funkcije koja se integrira, preciznosti koja je potrebna te resursa dostupnih za izračun. Neke metode poput trapezoidne, Simpsonove metode ili Gaussova kvadratura su determinističke metode jer koriste točno odabранe točke i težine za aproksimaciju integrala. Metoda numeričke integracije Monte Carlo koristi nasumično odabranе točke unutar područja integracije za izračun određenog integrala. Ova metoda je posebno korisna kada su problemi visoko-dimenzionalni ili kada je funkcija kompleksna, jer rezultira preciznijim izračunom uz manju računsku složenost. Naravno, što je veći broj nasumično odabranih točaka, to će i numerička aproksimacija biti točnija.

Algoritam za integraciju metodom Monte Carlo svodi se na numeričko računanje vrijednosti aritmetičke sredine funkcije $f(x)$ za N nasumično odabranih vrijednosti x iz intervala $[a, b]$ po formuli:

$$I_{[a,b]} \approx \frac{b-a}{N} \sum_i^N f(x) \quad (1)$$

Točke a i b su donja i gornja granica integracije. Izračunata vrijednost I je procijenjena vrijednost integrala funkcije $f(x)$ u zadanim intervalima. Greška Monte Carlo metode V služi za procjenu preciznosti izračuna i ovisi o broju slučajnih uzoraka N . Definirana je kao standardna devijacija podijeljena s korijenom broja N :

$$V \approx \frac{\sigma}{\sqrt{N}} \quad (2)$$

Standardnu devijaciju je moguće izračunati pomoću sljedeće formule:

$$\sigma = \sqrt{(b-a)(E[f(x)^2] - E[f(x)]^2)} \quad (3)$$

gdje je:

$$E[f(x)^2] = \frac{1}{N} \sum f(x)^2 \quad (4)$$

i

$$E[f(x)] = \frac{1}{N} \sum f(x) \quad (5)$$

1.2 Zadatak

Koristeći Monte Carlo integraciju potrebno je napisati program koji će numerički izračunati integral funkcije:

$$f(x) = \frac{\sin(x)}{x}$$

u granicama: $[a, b] = [0, 10]$.

Programsko rješenje mora biti napisano u programskom jeziku C korištenjem standardne biblioteke, bez vanjskih ovisnosti. Potrebno je predati dvije verzije rješenja:

1. Serijsku (jednodretvenu) verziju, bez korištenja biblioteke OpenMP s nazivom `single.c`;
2. Paralelnu (višedretvenu) verziju uz korištenje biblioteke OpenMP s nazivom `multi.c`.

I jednodretvena i višedretvena verzija programa primaju isključivo jedan parametar kao argument putem naredbenog retka koji predstavlja broj nasumično odabranih točaka N . Višedretvena verzija mora omogućiti specifikaciju broja dostupnih dretvi korištenjem varijable okoline (*environment variable*) `OMP_NUM_THREADS`. Uvezši u obzir sve navedeno, poziv višedretvene inačice vašeg programa izgledat će kao u primjeru:

```
>>OMP_NUM_THREADS=4 ./multi 5000000
```

ili

```
>>export OMP_NUM_THREADS=4  
>>./multi 5000000
```

Format ispisa ispravno izvršenog programa isto tako je dobro definiran te se sastoji redom od sljedećih podataka (svaki podatak isписан u zasebnom retku):

1. Izračuna, odnosno aproksimirane vrijednosti integrala u zadanim rasponu $I_{[a,b]}$ (*float*);
2. Grešku izračuna V (*float*);
3. Vremena potrebnog za izračun vrijednosti integrala u sekundama (*float*). Mjerjenje postavite na način da se mjeri vrijeme izvršavanja cijelog programa.

Svi navedeni podaci iz vašeg programa ispisuju se na standardni izlaz naredbenog retka (`STDOUT`), bez dodatnog teksta, drugih znakova ili praznina. Izlazi su odijeljeni isključivo jednim znakom `LF`. Predani programski kod mora se moći prevesti prevoditeljem `GCC` (verzija 14). U nastavku teksta vidljiv je primjer izlaza programa nakon poziva programa

```
>>OMP_NUM_THREADS=4 ./multi 500000:  
1.658476  
0.000072  
5.147297
```

Zadana kombinacija (1) prevoditelja, (2) načina primanja parametra kao argumenta programa putem naredbenog retka, te (3) formata ispisa rezultata na standardni izlaz (`STDOUT`), definira ispitnu okolinu u kojoj će biti evaluirana vaša rješenja. Odstupanje od specifikacije u nekim slučajevima može značiti gubitak bodova na laboratorijskoj vježbi.

Za dobivanje slučajnih vrijednosti uzoraka iz intervala $[a, b]$ potrebno je koristi *threadsafe* funkciju `rand_r(unsigned int *seed)` iz standardne biblioteke `stdlib`. Funkcija `rand()` se ne preporučuje u višedretvenim programima jer brojevi koji se generiraju ovise o prethodnom stanju kojeg u slučaju višedretvenog programa nekonzistentno dijele sve dretve u timu. U slučaju funkcije `rand_r()` svaka dretva će čuvati svoje stanje na osnovu kojeg će se generirati sljedeći slučajni broj. Sjeme za generiranje slučajnih brojeva u funkciji `rand_r` potrebno je postaviti tako da različiti pozivi programa generiraju različite slučajne brojeve, odn. različite konačne vrijednosti procjene integrala. U višedretvenom programu sjeme je potrebno postaviti zasebno za svaku dretvu tako da funkcija `rand_r()` može ispravno generirati slučajne brojeve za svaku dretvu.

1.3 Predaja

Sve datoteke vezane uz laboratorijsku vježbu (dvije datoteke s programskim kodom) potrebno je zapakirati isključivo u jednu ZIP arhivu ravne hijerarhije (bez poddirektorija itd), te učitati u sustav Moodle u sklopu pripadajuće aktivnosti. Arhiva ne treba sadržavati druge datoteke, uključujući prethodno prevedene programe, datoteke projektnog karaktera (npr. Makefile, Solution datoteke) i sl. Naziv datoteke treba biti u obliku **JM-BAG_Ime_Prezime.zip**.