Programowanie i metody numeryczne

Zadania – seria 9.

Metody Monte Carlo.

Zadanie 1. `pimc` – Liczba π.

Napisz szablon funkcji

```
template<typename F, typename RNG>
double intMonteCarlo(RNG &rng, F f, double a, double b, uint N)
```

która posługując się metodą Monte Carlo znajduje przybliżoną wartość całki oznaczonej

$$I = \int_a^b f(x)\,\mathrm{d}x$$

całkowalnej funkcji $f : \mathbb{R} \supseteq X \to \mathbb{R}$ w przedziale $[a,\, b] \subset X$, wykorzystując N losowo wybranych punktów. Funkcja `intMonteCarlo` przyjmuje następujące argumenty: `rng` – referencję do generatora liczb losowych, który powinna wykorzystać do wylosowania punktów; `f` – implementację funkcji f; `a` i `b` – odpowiadające kolejno liczbom a i b; `N` – odpowiadający liczbie N. Wartością zwracaną przez funkcję `intMonteCarlo` powinna być obliczona przez nią przybliżona wartość całki oznaczonej I.

Korzystając z tego szablonu, napisz program `pimc`, który oblicza metodą Monte Carlo przybliżoną wartość liczby π na podstawie zależności

$$\frac{\pi}{2} = \int_{-1}^{1} \sqrt{1 - x^2}\,\mathrm{d}x.$$

Program ma przyjmować jako argument wywołania dodatnią liczbę całkowitą określającą wartość N. Dla zadanej w ten sposób wartości N program powinien wyznaczać wartość liczby π stukrotnie, a następnie wypisywać na standardowe wyjście średnią wyznaczonych wartości oraz ich wariancję. Jako generator liczb losowych wykorzystaj odpowiednie narzędzia dostarczane przez plik nagłówkowy `<random>`.

Zadanie 2. `lemniscate` – Pole powierzchni ograniczonej lemniskatą Bernoulliego.

Napisz szablon funkcji

```
template<typename RNG>
double lemniscateArea(RNG &rng, uint N)
```

która posługując się metodą Monte Carlo znajduje przybliżoną wartość pola powierzchni ograniczonej lemniskatą Bernoulliego, opisaną równaniem

$$\left(x^2 + y^2\right)^2 \leqslant 2\left(x^2 - y^2\right),$$

wykorzystując N losowo wybranych punktów. Jako obszar całkowania przyjmij $\Omega = [-1,5,\ 1,5] \times [-0,6,\ 0,6]$. Funkcja `lemniscateArea` przyjmuje argument `rng` stanowiący referencję do generatora liczb losowych, który powinna wykorzystać do wylosowania punktów, oraz argument `N` odpowiadający liczbie N. Wartością zwracaną przez funkcję `lemniscateArea` powinna być obliczona przez nią przybliżona wartość szukanego pola powierzchni.

Korzystając z tego szablonu, napisz program `lemniscate`, który przyjmuje jako argument wywołania dodatnią liczbę całkowitą określającą wartość N. Program powinien wypisywać na standardowe wyjście obliczoną przez funkcję `lemniscateArea` wartość. Jako generator liczb losowych wykorzystaj odpowiednie narzędzia dostarczane przez plik nagłówkowy `<random>`.

* Zadanie 3. `nball` – Objętość n–wymiarowej kuli jednostkowej.

Napisz szablon funkcji

```
template<typename RNG>
double nBallVolume(RNG &rng, uint n, uint N)
```

która posługując się metodą Monte Carlo znajduje przybliżoną objętość kuli jednostkowej w n–wymiarowej przestrzeni euklidesowej, wykorzystując N losowo wybranych punktów. Funkcja `nBallVolume` przyjmuje argument `rng` stanowiący referencję do generatora liczb losowych, który powinna wykorzystać do wylosowania punktów, oraz argumenty `n` i `N` odpowiadające kolejno liczbom n i N. Wartością zwracaną przez funkcję `nBallVolume` powinna być obliczona przez nią przybliżona objętość n–wymiarowej kuli jednostkowej.

Korzystając z tego szablonu, napisz program `nball`, który przyjmuje jako argumenty wywołania dwie dodatnie liczby całkowite określające odpowiednio wartości n i N. Program powinien wypisywać na standardowe wyjście obliczoną przez funkcję `nBallVolume` objętość. Jako generator liczb losowych wykorzystaj odpowiednie narzędzia dostarczane przez plik nagłówkowy `<random>`.

* Zadanie 4. `demere` – Paradoks kawalera de Méré.

Paradoks kawalera de Méré to problem z dziedziny rachunku prawdopodobieństwa postawiony przez francuskiego matematyka–amatora Antoine'a Gombauda (1607 – 1684). Dotyczy on serii rzutów symetrycznymi sześciennymi kostkami do gry i polega na rozstrzygnięciu, które spośród następujących dwóch zdarzeń jest bardziej prawdopodobne:

- zdarzenie A: wyrzucenie przynajmniej jednej szóstki w czterech rzutach jedną kostką,
- zdarzenie B: wyrzucenie przynajmniej jednej pary szóstek w dwudziestu czterech rzutach dwiema kostkami.

Analityczne rozwiązanie tego problemu podał Blaise Pascal (1623 – 1662).

Napisz szablon funkcji

```
template<typename RNG>
void deMere(RNG &rng, double &pA, double &pB, uint N)
```

która posługując się metodą Monte Carlo znajduje przybliżone prawdopodobieństwa opisanych wyżej zdarzeń A i B. Funkcja `deMere` przyjmuje argument `rng` stanowiący referencję do generatora liczb losowych, który powinna wykorzystać do wylosowania punktów, argumenty `pA` i `pB` stanowiące referencje do zmiennych typu `double` oraz argument `N` odpowiadający liczbie N. Funkcja ta powinna przypisać obliczone prawdopodobieństwa zdarzeń A i B do zmiennych, odpowiednio, `pA` i `pB`.

Korzystając z tego szablonu, napisz program `demere`, który wypisywuje na standardowe wyjście prawdopodobieństwa zdarzeń A i B znalezione przez Ciebie w literaturze oraz ich przybliżone wartości obliczone przez funkcję `deMere`. Jako generator liczb losowych wykorzystaj odpowiednie narzędzia dostarczane przez plik nagłówkowy `<random>`.

Opracowanie: Bartłomiej Zglinicki.