# Sprawozdanie z laboratorium nr 5

# Całkowanie numeryczne – metoda prostokątów, trapezów, Simpsona i MC

Wojciech Matys

Wydział WIMiIP

Kierunek: Inżynieria obliczeniowa

Grupa lab nr 4

Data wykonania: 11.04.2024

## Cel ćwiczenia

Celem laboratorium było zapoznanie się z różnymi metodami całkowania numerycznego, a następnie implentacja ich w wybranym przez siebie języku, ja zdecydowałem się na C++ oraz analiza ich dokładności. Podczas laboratorium skoncentrowaliśmy się głównie na metodach prostokątów, trapezów, Simpsona oraz metodzie Monte Carlo.

## Wstęp teoretyczny

**2. Całkowanie numeryczne** to jedna z kluczowych dziedzin w analizie numerycznej, która zajmuje się przybliżonym obliczaniem wartości całki oznaczonej funkcji. Podczas gdy teoretyczne metody całkowania, takie jak całki nieoznaczone i oznaczone zwykle obejmują podstawowe zasady i reguły matematyczne, całkowanie numeryczne skupia się na wykorzystaniu metod numerycznych do przybliżonego rozwiązania całek, szczególnie tych, dla których nie ma analitycznych rozwiązań.

Istnieje wiele technik całkowania numerycznego, z których każda ma swoje własne zalety i ograniczenia. W praktyce, wybór odpowiedniej metody całkowania numerycznego zależy od rodzaju funkcji, dokładności, jakiej oczekujemy oraz dostępnych zasobów obliczeniowych. W niniejszym sprawozdaniu przeanalizujemy różne metody całkowania numerycznego, ich zasady działania, zalety i ograniczenia, a także porównamy ich skuteczność w przybliżaniu wartości całek dla różnych typów funkcji.

## 2.1 Metoda prostokątów

Metoda prostokątów to jedna z najprostszych technik całkowania numerycznego, która polega na podziale obszaru pod wykresem funkcji na niewielkie prostokątne obszary i obliczeniu sumy pól tych prostokątów.

Istnieją trzy podstawowe warianty metody prostokątów:

**1. *Metoda prostokątów lewych:*** W tej metodzie wysokość każdego prostokąta jest równa wartości funkcji na lewym krańcu przedziału między kolejnymi punktami podziału. Suma pól wszystkich prostokątów daje przybliżoną wartość całki.

**2. *Metoda prostokątów prawych:*** W tym wypadku wysokość każdego prostokąta jest równa wartości funkcji na prawym krańcu przedziału między kolejnymi punktami podziału.

**3. *Metoda prostokątów środkowych:*** Tutaj wysokość prostokąta jest równa wartości funkcji w środku przedziału między punktami podziału.

Rysując to graficznie, dla funkcji o jednym wymiarze, zakładając równy podział przedziału na n prostokątów, całka przybliżona przy użyciu metody prostokątów może być obliczona według wzoru:

$$\text{Całka przybliżona} = h \cdot \sum_{i=1}^{n} f(x_i) \tag{1}$$

Gdzie:

- $h$ to szerokość każdego z prostokątów (czyli różnica między kolejnymi punktami podziału),

- $f(x_i)$ to wartość funkcji $f$ w punkcie $(x_i)$ (który może być lewym, prawym lub środkowym krańcem prostokąta).

Metoda prostokątów jest łatwa do zrozumienia i implementacji, ale może być niedokładna, szczególnie dla funkcji o dużych zmianach wartości w krótkich przedziałach. Jednakże może być użyteczna jako punkt wyjścia do bardziej zaawansowanych technik całkowania numerycznego lub dla funkcji o prostym kształcie, gdzie dokładność nie jest kluczowa.

## 2.2 Metoda trapezów

Metoda trapezów to kolejna popularna technika całkowania numerycznego, która polega na przybliżeniu obszaru pod wykresem funkcji za pomocą trapezów, zamiast prostokątów, jak w metodzie prostokątów. Jest to bardziej zaawansowana metoda niż prostokąty, ponieważ uwzględnia nachylenie funkcji między punktami podziału.

Idea metody trapezów polega na przybliżeniu obszaru pod wykresem funkcji na danym przedziale poprzez zastosowanie trapezów. Dla każdego segmentu między dwoma kolejnymi punktami podziału, pole trapezu jest obliczane na podstawie wysokości funkcji w obu tych punktach. Następnie sumowane są pola wszystkich trapezów, co daje przybliżoną wartość całki.

Matematycznie, dla funkcji o jednym wymiarze, przybliżona wartość całki za pomocą metody trapezów może być obliczona według wzoru:

$$\text{Całka przybliżona} = \frac{h}{2} \cdot \sum_{i=1}^{n} \left( f(x_{i-1}) + f(x_i) \right) \qquad (2)$$

Gdzie:

- $(h)$ to szerokość każdego z segmentów (różnica między kolejnymi punktami podziału),

- $f(x_{i-1})$ i $f(x_i)$ to wartości funkcji $f$ w dwóch kolejnych punktach podziału.

Warto zauważyć, że każdy trapez składa się z dwóch równoległych boków (które są wartościami funkcji w punktach podziału) i dwóch boków skośnych (które łączą te punkty podziału).

Metoda trapezów jest dokładniejsza niż metoda prostokątów, ponieważ uwzględnia nachylenie funkcji między punktami podziału. Jest stosunkowo łatwa do zrozumienia i implementacji, co sprawia, że jest popularnym wyborem w praktyce. Jednakże, podobnie jak inne metody numeryczne, jej dokładność zależy od liczby segmentów podziału oraz od regularności funkcji w danym przedziale.

## 2.3 Metoda Simpsona

Metoda Simpsona, znana również jako reguła Simpsona, jest techniką całkowania numerycznego wykorzystującą interpolację wielomianową stopnia drugiego (czyli parabol) do przybliżenia obszaru pod wykresem funkcji. Jest to bardziej zaawansowana metoda niż metoda trapezów lub prostokątów, ponieważ interpoluje funkcję za pomocą parabol zamiast odcinków prostych, co może prowadzić do dokładniejszych wyników.

Idea metody Simpsona polega na podziale obszaru pod wykresem funkcji na niewielkie obszary podparabolowe (paraboloidy). Następnie, dla każdego takiego obszaru, pole podparabolowe jest obliczane przy użyciu wzoru na pole powierzchni pod parabolą. Suma pól wszystkich podparabolowych obszarów daje przybliżoną wartość całki.

Matematycznie, dla funkcji o jednym wymiarze, przybliżona wartość całki za pomocą metody Simpsona może być obliczona według wzoru:

$$\text{Całka przybliżona} = \frac{h}{3}\left( f(x_0) + 4\sum_{i=1}^{n/2} f(x_{2i-1}) + 2\sum_{i=1}^{n/2-1} f(x_{2i}) + f(x_n) \right) \quad (3)$$

Gdzie:

- $h$ to szerokość każdego z podparabolowych obszarów (różnica między kolejnymi punktami podziału),

- $f(x_i)$ to wartość funkcji $f$ w punkcie $x_i$

- $n$ to liczba równoodległych punktów podziału (musi być liczbą parzystą).

Metoda Simpsona jest znacznie bardziej dokładna niż metoda trapezów, ponieważ uwzględnia zakrzywienie funkcji i interpoluje ją za pomocą parabol. Jest stosunkowo prosta w implementacji, ale wymaga podziału przedziału na parzystą liczbę segmentów. Jednakże, dla funkcji o nieregularnym kształcie lub występujących skokach, może być mniej dokładna niż inne zaawansowane metody całkowania numerycznego.

## 2.4 Metoda Monte Carlo

Metoda Monte Carlo w całkowaniu numerycznym jest techniką probabilistyczną, która opiera się na losowaniu próbek z obszaru całkowania i wykorzystuje je do estymacji wartości całki. Jest to niezwykle elastyczna metoda, która może być stosowana do całkowania funkcji wielowymiarowych oraz funkcji o złożonych kształtach, które mogą być trudne do całkowania za pomocą tradycyjnych metod numerycznych.

Idea metody Monte Carlo polega na losowym wyborze punktów w obszarze, który chcemy zintegrować, oraz na estymacji wartości całki na podstawie tych próbek. Im więcej próbek zostanie wygenerowanych, tym bardziej dokładna będzie estymacja wartości całki.

Matematycznie, przybliżona wartość całki za pomocą metody Monte Carlo może być obliczona według wzoru:

$$\text{Całka przybliżona} = V \cdot \frac{1}{N}\sum_{i=1}^{N} f(x_i) \quad (4)$$

Gdzie:

- $V$ to objętość obszaru całkowania,

- $N$ to liczba losowych próbek,

- $f(x_i)$ to wartość funkcji $f$ w losowo wybranym punkcie $(x_i)$.

Metoda Monte Carlo jest szczególnie przydatna, gdy obszar całkowania ma skomplikowany kształt, ponieważ nie wymaga ona podziału na podprzedziały ani interpolacji funkcji. Poza tym, metoda ta jest łatwa w zastosowaniu i skalowalna, dzięki czemu może być stosowana do całkowania funkcji o bardzo dużych wymiarach.

Jednakże, aby uzyskać dokładne wyniki, konieczne jest wygenerowanie odpowiednio dużej liczby losowych próbek, co może wymagać większych zasobów obliczeniowych w porównaniu z innymi metodami całkowania numerycznego. Ponadto, dla funkcji o bardzo nierównomiernym rozkładzie wartości, konieczne może być zastosowanie zaawansowanych technik próbkowania.

## Implementacja numeryczna

```
double calkowanie_prostokatow(double xp, double xk, int n) {
    double dx = (xk - xp) / n;
    double suma = 0;

    for (int i = 1; i <= n; ++i) {
        suma += funkcja(xp + i * dx);
    }

    return dx * suma;
}
```

*(1. Fragment kodu przedstawiający całkowanie metodą prostokąktów)*

***1***. Deklaracja funkcji `calkowanie_prostokatow`, która przyjmuje trzy argumenty: `xp` - dolny limit całkowania, `xk` - górny limit całkowania, `n` - ilość podziałów przedziału całkowania.

***2***. Obliczenie szerokości pojedynczego podprzedziału `dx` przy użyciu wzoru `(xk - xp) / n`. Szerokość ta jest równa różnicy między górnym i dolnym limitem całkowania podzielonej przez liczbę podprzedziałów.

***3.*** Inicjalizacja zmiennej `suma` na 0, która będzie używana do akumulowania sumy wartości funkcji na podprzedziałach.

***4.*** Pętla `for` iteruje od `i = 1` do `i = n` (włącznie), inkrementując `i` o 1 za każdym przebiegiem.

***5***. W każdej iteracji obliczana jest wartość argumentu dla funkcji, która ma zostać zintegrowana. Jest to wyrażenie `xp + i * dx`, które odpowiada aktualnemu punktowi na osi x wewnątrz aktualnego podprzedziału.

***6***. Wywołanie funkcji `funkcja` z obliczonym argumentem i dodanie jej wartości do sumy `suma`.

***7.*** Po zakończeniu pętli, zwracana jest wartość całki przybliżonej, obliczona jako iloczyn szerokości podprzedziału `dx` i sumy wartości funkcji na tych podprzedziałach.

Kod ten implementuje prostą metodę całkowania numerycznego opartą na metodzie prostokątów (z dokładnością do lewych krańców). Dzieli on obszar całkowania na `n` równych podprzedziałów i przybliża wartość całki sumą wartości funkcji na tych podprzedziałach pomnożoną przez szerokość pojedynczego podprzedziału.

```
double calkowanie_trapezu(double xp, double xk, int n) {
    double dx = (xk - xp) / n;
    double suma = 0;
    xk = xp + dx;
    for (int i = 0; i < n; i++) {

        suma += (funkcja(xp) + funkcja(xk))/2;
        xp = xp + dx;
        xk = xk + dx;

    }

    return dx * suma;
}
```

*(2. Fragment kodu przedstawiający całkowanie metodą trapezów)*

Ten kod implementuje metodę całkowania numerycznego zwaną metodą trapezów.

***1.*** Deklaracja funkcji `calkowanie_trapezu`, która przyjmuje trzy argumenty: `xp` - dolny limit całkowania, `xk` - górny limit całkowania, `n` - liczbę podprzedziałów użytych do podziału przedziału całkowania.

***2.*** Obliczenie szerokości pojedynczego podprzedziału `dx` przy użyciu wzoru `(xk - xp) / n`. Szerokość ta jest równa różnicy między górnym i dolnym limitem całkowania podzielonej przez liczbę podprzedziałów.

***3.*** Inicjalizacja zmiennej `suma` na 0, która będzie używana do akumulowania sumy wartości funkcji na podprzedziałach.

***4.*** Ustawienie zmiennej `xk` jako `xp + dx`. Jest to pierwszy prawy koniec podprzedziału.

***5***. Pętla `for` iteruje od `i = 0` do `i < n`, inkrementując `i` o 1 za każdym przebiegiem.

***6.*** W każdej iteracji obliczana jest wartość całki na danym podprzedziale. Wartość ta jest przybliżana jako średnia wartość funkcji na lewym i prawym krańcu podprzedziału. Stąd `(funkcja(xp) + funkcja(xk)) / 2`.

***7.*** Dodanie tej przybliżonej wartości całki do sumy `suma`.

***8***. Przesunięcie lewego i prawego krańca podprzedziału o szerokość `dx`. Jest to konieczne, aby przejść do następnego podprzedziału.

***9.*** Po zakończeniu pętli, zwracana jest wartość całki przybliżonej, która jest sumą wartości całek na wszystkich podprzedziałach.

Ogólnie, kod ten przybliża wartość całki na danym przedziale za pomocą metody trapezów, która dzieli obszar całkowania na niewielkie trapezy i sumuje ich pola. Ta metoda jest dokładniejsza niż metoda prostokątów, ponieważ uwzględnia nachylenie funkcji między punktami podziału.

```
double calkowanie_simpsona(double xp, double xk, int n) {
    double dx = (xk - xp) / n;
    double suma = funkcja(xp) + funkcja(xk);

    for (int i = 1; i < n; i++) {
        double x = xp + i * dx; // Wartość x w punkcie środkowym podprzedziału
        if (i % 2 == 0) {
            suma += 2 * funkcja(x); // Wartość funkcji dla punktów parzystych
        }
        else {
            suma += 4 * funkcja(x); // Wartość funkcji dla punktów nieparzystych
        }
    }

    return (dx / 3) * suma; // Obliczenie całki metodą Simpsona
}
```

*(3. Fragment kodu przedstawiający całkowanie metodą Simpsona)*

Ten kod implementuje metodę całkowania numerycznego znana jako metoda Simpsona.

***1.*** Deklaracja funkcji `calkowanie_simpsona`, która przyjmuje trzy argumenty: `xp` - dolny limit całkowania, `xk` - górny limit całkowania, `n` - liczbę podprzedziałów użytych do podziału przedziału całkowania.

***2.*** Obliczenie szerokości pojedynczego podprzedziału `dx` przy użyciu wzoru `(xk - xp) / n`. Szerokość ta jest równa różnicy między górnym i dolnym limitem całkowania podzielonej przez liczbę podprzedziałów.

***3.*** Inicjalizacja zmiennej `suma` jako wartość funkcji w punktach krańcowych przedziału całkowania. Jest to początkowe przybliżenie całki metodą Simpsona, które obejmuje krańce przedziału.

***4.*** Pętla `for` iteruje od `i = 1` do `i < n`, inkrementując `i` o 1 za każdym przebiegiem. Iteracja rozpoczyna się od drugiego podprzedziału, ponieważ wartości funkcji na krańcach przedziału zostały już uwzględnione.

***5***. W każdej iteracji obliczana jest wartość zmiennej `x`, która reprezentuje środek bieżącego podprzedziału.

***6***. Następnie sprawdzane jest, czy indeks `i` jest parzysty czy nieparzysty. Jeśli jest parzysty, to znaczy, że punkt `x` znajduje się na parzystym miejscu, więc jego wartość funkcji jest mnożona przez 2. Jeśli jest nieparzysty, wartość funkcji jest mnożona przez 4. To wynika z metody Simpsona, gdzie wartość funkcji jest mnożona przez 4 lub 2 w zależności od tego, czy punkt jest nieparzysty czy parzysty.

***7***. Obliczona wartość jest dodawana do sumy `suma`.

***8***. Po zakończeniu pętli, całka przybliżona jest obliczana jako iloczyn `(dx / 3)` i sumy `suma`, zgodnie z metodą Simpsona.

***9.*** Zwracana jest wartość przybliżonej całki.

Ogólnie, kod ten przybliża wartość całki na danym przedziale za pomocą metody Simpsona, która interpoluje funkcję za pomocą parabol, aby uzyskać dokładniejsze wyniki niż metoda trapezów. Metoda ta jest stosowana do całkowania funkcji, które mają gładkie krzywe, a nie tylko odcinki proste.

```cpp
double calkowanie_monte_carlo(double xp, double xk, int n) {
    random_device rd;
    mt19937 gen(rd());
    uniform_real_distribution<> dis(xp, xk);

    double suma = 0;
    for (int i = 0; i < n; ++i) {
        double x = dis(gen);
        suma += funkcja(x);
    }

    return (xk - xp) * suma / n;
}
```

*(4. Fragment kodu przedstawiający całkowanie metodą Monte Carlo)*

Ten kod implementuje metodę całkowania numerycznego znana jako metoda Monte Carlo.

***1.*** Deklaracja funkcji `calkowanie_monte_carlo`, która przyjmuje trzy argumenty: `xp` - dolny limit całkowania, `xk` - górny limit całkowania, `n` - liczbę próbek losowych użytych do estymacji całki.

***2.*** Tworzenie obiektu `random_device`, który generuje losowe liczby z jednorodnym rozkładem w czasie rzeczywistym.

***3.*** Inicjalizacja generatora liczb pseudolosowych `mt19937` za pomocą `random_device` jako ziarna (seed), co pozwala na generowanie sekwencji pseudolosowych liczb.

***4***. Utworzenie dystrybucji jednostajnej `uniform_real_distribution<>`, która generuje losowe liczby z równomiernym rozkładem w przedziale `[xp, xk]`.

***5***. Inicjalizacja zmiennej `suma` jako 0, która będzie używana do akumulowania sumy wartości funkcji dla wylosowanych punktów.

***6***. Pętla `for` iteruje od `i = 0` do `i < n`, inkrementując `i` o 1 za każdym przebiegiem.

***7***. W każdej iteracji generowana jest losowa wartość `x` z przedziału `[xp, xk]` za pomocą dystrybucji jednostajnej.

***8***. Obliczana jest wartość funkcji dla wylosowanego punktu `x` i dodawana jest do sumy `suma`.

***9.*** Po zakończeniu pętli, całka przybliżona jest obliczana jako iloczyn `(xk - xp)` (czyli szerokość przedziału całkowania) i średniej wartości funkcji dla wszystkich wylosowanych punktów, podzielona przez liczbę próbek `n`.

***10.*** Zwracana jest wartość przybliżonej całki.

Ogólnie, metoda Monte Carlo polega na losowym wyborze próbek z obszaru całkowania i wykorzystaniu ich do estymacji wartości całki. Im więcej próbek zostanie wygenerowanych, tym bardziej dokładny będzie wynik. Metoda ta jest szczególnie przydatna do całkowania funkcji o skomplikowanych kształtach, które mogą być trudne do całkowania za pomocą tradycyjnych metod numerycznych.

# Testy jednostkowe

## Test 1

### Test 1.1

Funkcja: $f(x) = x^3 + 3x - 2$

Przedział: [3; 8] czyli xp = 3, xk = 8

Ilość podprzedziałów: n = 4

Wynik otrzymany przez program:

```
Przyblizona wartosc calki (metoda prostokatow): 1430.23
Przyblizona wartosc calki (metoda trapezow): 1117.73
Przyblizona wartosc calki (metoda Simpsona): 1096.25
Przyblizona wartosc calki (metoda Monte Carlo): 1331.58
```

*(wyniki test 1.1)*

### Test 1.2

Funkcja: $f(x) = x^3 + 3x - 2$

Przedział: [3; 8] czyli xp = 3, xk = 8

Ilość podprzedziałów: n = 40

Wynik otrzymany przez program:

```
Przyblizona wartosc calki (metoda prostokatow): 1127.71

Przyblizona wartosc calki (metoda trapezow): 1096.46
Przyblizona wartosc calki (metoda Simpsona): 1096.25
Przyblizona wartosc calki (metoda Monte Carlo): 1032.08
```

*(wyniki test 1.2)*

### Test 1.3

Funkcja: $f(x) = x^3 + 3x - 2$

Przedział: [3; 8] czyli xp = 3, xk = 8

Ilość podprzedziałów: n = 400

Wynik otrzymany przez program:

```
Przyblizona wartosc calki (metoda prostokatow): 1099.38
Przyblizona wartosc calki (metoda trapezow): 1096.25
Przyblizona wartosc calki (metoda Simpsona): 1096.25
Przyblizona wartosc calki (metoda Monte Carlo): 1113.63
```

(wyniki test 1.3)

## Analiza wyników Test 1

Wynik policzony przez wolfram:

$$\int_3^8 (x^3 + 3x + 2)\,dx = \frac{4385}{4} = 1096.25$$

| Ilość poddziałów | Metoda kwadratów | Metoda trapezów | Metoda Simpsona | Metoda Monte Carlo | Wynik właściwy |
|---|---|---|---|---|---|
| 4 | 1430.23 | 1117.73 | 1096.25 | 1331.58 | 1096.25 |
| 40 | 1127.71 | 1096.46 | 1096.25 | 1032.08 | |
| 400 | 1099.38 | 1096.25 | 1096.25 | 1113.63 | |

Test 2

Test 2.1

Funkcja: $f(x) = \log(x)$

Przedział: [2; 18] czyli xp = 2, xk = 18

Ilość podprzedziałów: n = 4

Wynik otrzymany przez program:

```
Przyblizona wartosc calki (metoda prostokatow): 38.4951
Przyblizona wartosc calki (metoda trapezow): 34.1006
Przyblizona wartosc calki (metoda Simpsona): 34.5493
Przyblizona wartosc calki (metoda Monte Carlo): 25.4663
```

(wyniki test 2.1)

Test 2.2

Funkcja: $f(x) = \log(x)$

Przedział: [2; 18] czyli xp = 2, xk = 18

Ilość podprzedziałów: n = 40

Wynik otrzymany przez program:

```
Przyblizona wartosc calki (metoda prostokatow): 35.0739
Przyblizona wartosc calki (metoda trapezow): 34.6345
Przyblizona wartosc calki (metoda Simpsona): 34.6404
Przyblizona wartosc calki (metoda Monte Carlo): 33.4931
```

(wyniki test 2.2)

Test 2.3

Funkcja: $f(x) = \log(x)$

Przedział: [2; 18] czyli xp = 2, xk = 18

Ilość podprzedziałów: n = 400

Wynik otrzymany przez program:

```
Przyblizona wartosc calki (metoda prostokatow): 34.6843
Przyblizona wartosc calki (metoda trapezow): 34.6403
Przyblizona wartosc calki (metoda Simpsona): 34.6404
Przyblizona wartosc calki (metoda Monte Carlo): 34.5625
```

(wyniki test 2.3)

## Analiza wyników Test 2

Wynik policzony przez wolfram:

$$\int_2^{18} \log(x)\, dx = -16 - \log(4) + 18\log(18) \approx 34.6403972810111$$

| Ilość poddziałów | Metoda kwadratów | Metoda trapezów | Metoda Simpsona | Metoda Monte Carlo | Wynik właściwy |
|---|---|---|---|---|---|
| 4 | 38.4951 | 34.1006 | 34.5493 | **25.4663** | |
| 40 | 35.0739 | 34.6345 | **34.6404** | 33.4931 | 34.6404 |
| 400 | 34.6843 | **34.6403** | **34.6404** | 34.5045 | |

## Test 3

### Test 3.1

Funkcja: $f(x) = \sin(x)$

Przedział: $[0; \frac{3}{4}\pi]$ czyli xp = 0, xk = ; $\frac{3}{4}\pi$

Ilość podprzedziałów: n = 4

Wynik otrzymany przez program:

```
Przyblizona wartosc calki (metoda prostokatow): 1.86562
Przyblizona wartosc calki (metoda trapezow): 1.65733
Przyblizona wartosc calki (metoda Simpsona): 1.70816
Przyblizona wartosc calki (metoda Monte Carlo): 1.61967
```

(wyniki test 3.1)

**Test 3.2**

Funkcja: $f(x) = \sin(\mathrm{x})$

Przedział: $[0; \frac{3}{4}\pi]$ czyli xp = 0, xk = ; $\frac{3}{4}\pi$

Ilość podprzedziałów: n = 40

Wynik otrzymany przez program:

```
Przyblizona wartosc calki (metoda prostokatow): 1.7273

Przyblizona wartosc calki (metoda trapezow): 1.70648
Przyblizona wartosc calki (metoda Simpsona): 1.70697
Przyblizona wartosc calki (metoda Monte Carlo): 1.66955
```

(wyniki test 3.2)

**Test 3.2**

Funkcja: $f(x) = \sin(\mathrm{x})$

Przedział: $[0; \frac{3}{4}\pi]$ czyli xp = 0, xk = ; $\frac{3}{4}\pi$

Ilość podprzedziałów: n = 400

Wynik otrzymany przez program:

```
Przyblizona wartosc calki (metoda prostokatow): 1.70905
Przyblizona wartosc calki (metoda trapezow): 1.70696
Przyblizona wartosc calki (metoda Simpsona): 1.70697
Przyblizona wartosc calki (metoda Monte Carlo): 1.7357
```

(wyniki test 3.3)

## Analiza wyników Test 3

Wynik policzony przez wolfram:

$$\int_0^{\frac{3\pi}{4}} \sin(x)\,dx = 1 + \frac{1}{\sqrt{2}} \approx 1.7071$$

| Ilość poddziałów | Metoda kwadratów | Metoda trapezów | Metoda Simpsona | Metoda Monte Carlo | Wynik właściwy |
|---|---|---|---|---|---|
| 4 | **1.86562** | 1.65733 | **1.70816** | 1.61967 | |
| 40 | 1.72762 | **1.70648** | **1.70697** | 1.66955 | 1.707106 |
| 400 | 1.70905 | **1.70696** | **1.70697** | 1.73571 | |

## Podsumowanie wyników

Na podstawie przedstawionych wyników, ponownie potwierdza się, że metoda Simpsona jest najbardziej skuteczną metodą całkowania spośród wszystkich prezentowanych. We wszystkich trzech testach błąd tej metody jest najmniejszy.

Metoda trapezów plasuje się na drugim miejscu pod względem skuteczności.

Metoda kwadratów, choć prosta, może być mniej dokładna, szczególnie dla funkcji o dużej zmienności w krótkich przedziałach, co widać w niektórych przypadkach, gdzie jej wyniki są odstające od wartości właściwej.

Wraz ze wzrostem liczby podziałów wszystkie metody numeryczne zbliżają się do wartości właściwej, co pokazuje, że zwiększenie ilości podziałów zazwyczaj poprawia dokładność estymacji całki.

Ogólnie, wybór odpowiedniej metody całkowania zależy od charakterystyki funkcji i wymagań dotyczących dokładności. Metoda Monte Carlo jest często preferowaną metodą ze względu na jej elastyczność i możliwość uzyskania dokładnych wyników dla szerokiej gamy funkcji, ale niestety potrzebuję, aby być dokładna bardzo dużą liczbę przedziałów.

## Wnioski

Metody numerycznego całkowania, takie jak metoda prostokątów, metoda trapezów, metoda Simpsona i metoda Monte Carlo, są użytecznymi narzędziami do przybliżonego obliczania wartości całek, szczególnie gdy funkcje nie mają analitycznych rozwiązań. Każda z tych metod ma swoje plusy i minusy. Metoda prostokątów i metoda trapezów są łatwe do zrozumienia i implementacji, ale mogą wymagać wielu podziałów, by osiągnąć dokładność, zwłaszcza dla funkcji nieregularnych.

Metoda Simpsona, która opiera się na przybliżaniu funkcji parabolami, może być dokładniejsza dla funkcji gładkich, ponieważ uwzględnia krzywiznę funkcji w przybliżeniu. Metoda Monte Carlo, z kolei, korzysta z losowego próbkowania z przedziału całkowania i może być przydatna dla funkcji trudnych do analizy, ale wymaga większej liczby próbek, aby uzyskać precyzyjne wyniki, szczególnie dla funkcji o dużej zmienności.

Podczas testów każda z tych metod wykazała swoje mocne i słabe strony. Metoda Simpsona często dawała najbardziej dokładne wyniki, szczególnie dla funkcji gładkich, podczas gdy metoda Monte Carlo była bardziej elastyczna, ale potrzebowała więcej próbek dla dokładnych wyników.

Wybór odpowiedniej metody numerycznego całkowania zależy od charakterystyki funkcji oraz pożądanego poziomu dokładności. Dla funkcji prostych i gładkich, metody prostokątów, trapezów lub Simpsona mogą być wystarczające, podczas gdy dla bardziej skomplikowanych funkcji, metoda Monte Carlo może być bardziej odpowiednia.

Podsumowując, stosowanie różnych metod numerycznego całkowania pozwala na elastyczne podejście do rozwiązywania problemów związanych z całkowaniem, umożliwiając wybór najlepszej metody dla konkretnego przypadku.