

1. Treść zadania

1.1. Zadanie pierwsze

Wiadomo, że

$$\int_0^1 \frac{4}{1+x^2} dx = \pi$$

Powyższą równość można wykorzystać do obliczenia przybliżonej wartości π po- przez całkowanie numeryczne.

Obliczę wartość powyższej całki, korzystając ze złożonych kwadratur otwartej prostokątów (ang. mid-point rule), trapezów i Simpsona. Na przedziale całkowania rozmieszczę $2^m + 1$ równoodległych węzłów. W kolejnych próbach m wzrasta o 1, tzn. między każde dwa sąsiednie węzły dodawany jest nowy węzeł, a ich zagęszczenie zwiększa się dwukrotnie. Przyjmiję zakres wartości m od 1 do 25.

Dla każdej metody narysuj wykres wartości bezwzględnej błędu względnego w zależności od liczby ewaluacji funkcji podcałkowej, $n + 1$ (gdzie $n = 1/h$, z krokiem h). Wyniki przedstaw na wspólnym wykresie, używając skali logarytmicznej na obu osiach.

Czy istnieje pewna wartość, poniżej której zmniejszanie kroku h nie zmniejsza już błędu kwadratury? Porównaj wartość h_{\min} , odpowiadającą minimum wartości bezwzględnej błędu względnego, z wartością wyznaczoną w laboratorium 1.

Dla każdej z użytych metod porównaj empiryczny rząd zbieżności z rzęd zbieżności przewidywanym przez teorię. Aby wyniki miały sens, do obliczenia rzędu empirycznego użyj wartości h z zakresu, w którym błąd metody przeważa nad błędem numerycznym.

1.2. Zadanie drugie

Obliczę wartość całki

$$\int_0^1 \frac{4}{1+x^2} dx$$

metodą Gaussa-Legendre'a. Narysuję wykres wartości bezwzględnej błędu względnego w zależności od liczby ewaluacji funkcji podcałkowej, $n + 1$. Przyjmę na tyle duży zakres n , aby wykryć, kiedy błąd numeryczny zaczyna przeważać nad błędem metody.

2. Rozwiązanie zadań

2.1. Zadanie pierwsze

2.1.1. Funkcja podcałkowa

```
def f(x):  
    return 4 / (1 + x ** 2)
```

2.1.2. Dokładna wartość całki

```
exact_value = np.pi
```

2.1.3. Zakres wartości m

```
m_values = np.arange(1, 26)
```

2.1.4. Lista przechowująca liczby węzłów dla kolejnych wartości m

```
num_nodes_list = []
```

2.1.5. Wyniki całkowania

```
integral_mid_list = []  
integral_trap_list = []  
integral_simp_list = []
```

2.1.6. Przechowywanie błędów

```
errors_mid_list = []  
errors_trap_list = []  
errors_simp_list = []
```

2.1.7. Pętla po wartościach m w celu obliczenia błędów poszczególnych metod

```
✓ for m in m_values:
    # Obliczenie liczby węzłów
    num_nodes = 2 ** m + 1
    num_nodes_list.append(num_nodes)

    # Rozmieszczenie równoodległych węzłów
    x = np.linspace(0, 1, num_nodes)

    # Złożona kwadratura otwarta prostokątów (metoda mid-point)
    dx = (1 - 0) / num_nodes
    x_mid = (x[:-1] + x[1:]) / 2
    integral_mid = np.sum(f(x_mid) * dx)
    integral_mid_list.append(integral_mid)

    # Złożona kwadratura trapezów
    integral_trap = trapz(f(x), x)
    integral_trap_list.append(integral_trap)

    # Złożona kwadratura Simpsona
    integral_simp =.simps(f(x), x)
    integral_simp_list.append(integral_simp)

    errors_mid_list.append(abs((np.pi - integral_mid_list[m-1]) / np.pi))
    errors_trap_list.append(abs((np.pi - integral_trap_list[m-1]) / np.pi))
    errors_simp_list.append(abs((np.pi - integral_simp_list[m-1]) / np.pi))
```

2.1.8. Obliczenie wartości poniżej której zmniejszenie kroku h nie zmniejsza już błędu kwadratury dla poszczególnych metod

2.1.8.1. Dla metody trapezów:

```
hmin_index = np.argmin(errors_mid_list)
hmin_trap = 1 / num_nodes_list[hmin_index]
print("Wartość hmin:", hmin_trap)
```

2.1.8.2. Dla metody Simpsona:

```
hmin_index = np.argmin(errors_simp_list)
hmin_simp = 1 / num_nodes_list[hmin_index]
print("Wartość hmin:", hmin_simp)
```

2.1.8.3. Dla metody prostokątów:

```
hmin_index = np.argmin(errors_mid_list)
hmin_midpoint = 1 / num_nodes_list[hmin_index]
print("Wartość hmin:", hmin_midpoint)
```

2.1.9. Funkcja do obliczenia rzędu zbieżności

```
def calculate_convergence_order(errors, hs):
    p_values = []

    for i in range(len(errors) - 1):
        if errors[i] == 0 or errors[i+1] == 0:
            continue
        p = np.abs(np.log(errors[i+1] / errors[i]) / np.log(hs[i+1] / hs[i]))
        p_values.append(p)

    return p_values
```

2.1.10. Obliczanie rzędu zbieżności

```
hs = np.logspace(-15, -1, 70)

p_values_trapezoidal = calculate_convergence_order(errors_trap_list, hs)
p_values_simp = calculate_convergence_order(errors_simp_list, hs)
p_values_mid = calculate_convergence_order(errors_mid_list, hs)

print("Rząd zbieżności dla metody trapezów: ", np.mean(p_values_trapezoidal))
print("Rząd zbieżności dla metody Simpsona: ", np.mean(p_values_simp))
print("Rząd zbieżności dla metody mid-point: ", np.mean(p_values_mid))
```

2.1.11. Dobieranie zakresu wartości n dla metody Gaussa-Legendre'a

```
n_values = np.arange(1, 100)
```

2.1.12. Lista przechowująca wartości błędu względnego dla każdej liczby węzłów

```
errors_gauss_list = []
```

2.1.13. Metoda Gaussa Legendre'a

```
def gauss_legendre_integration(n):
    nodes, weights = roots_legendre(n)

    x = 0.5 * (nodes + 1)
    w = 0.5 * weights

    integral_value = np.sum(w * f(x))

    return integral_value
```

2.1.14. Obliczenie wartości błędu względnego dla różnych liczby węzłów

```
for m in m_values:
    # Obliczenie wartości całki metoda Gaussa-Legendre'a
    integral_value = gauss_legendre_integration(m)

    # Obliczenie błędu względnego
    error = np.abs((exact_value - integral_value) / exact_value)
    errors_gauss_list.append(error)
```

2.1.15. Obliczenie rzędu zbieżności oraz h_{\min} dla metody Gaussa-Lagrange'a

```
p_values_gauss = calculate_convergence_order(errors_gauss_list, hs)
print("Rząd zbieżności dla metody Gaussa: ", np.mean(p_values_gauss))

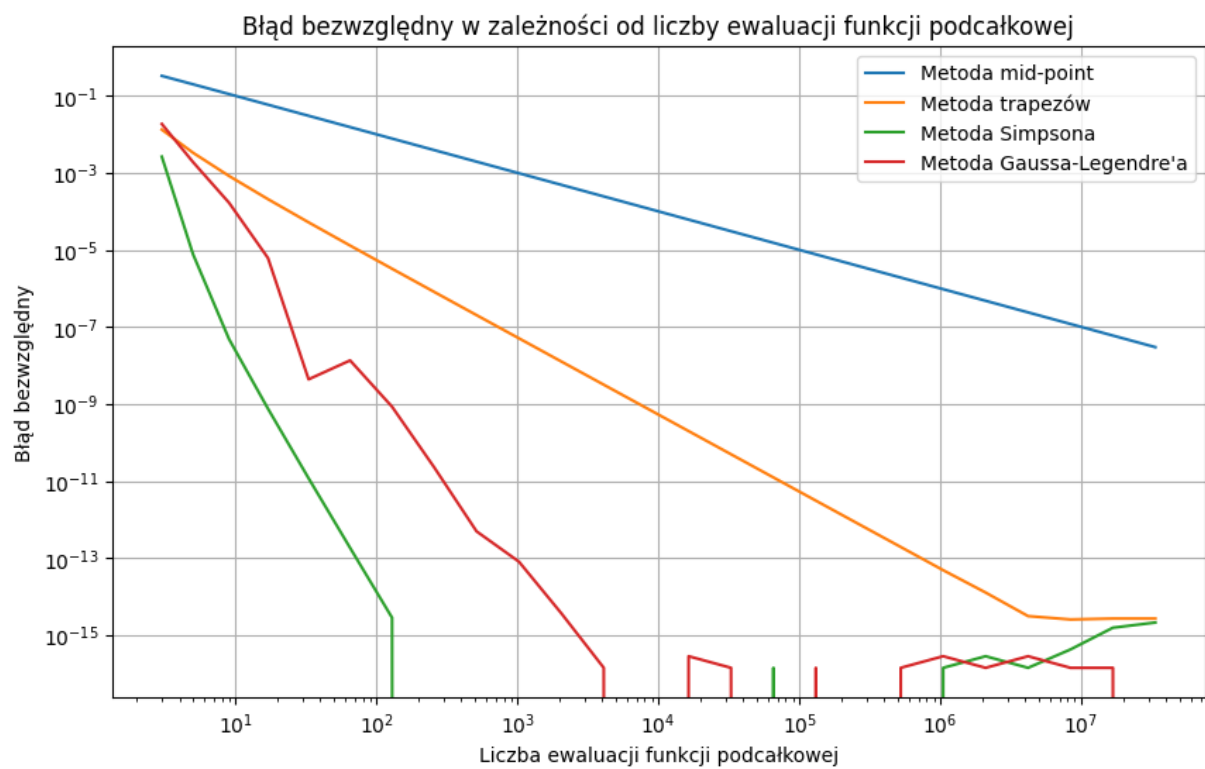
hmin_index = np.argmin(errors_gauss_list)
hmin_midpoint = 1 / num_nodes_list[hmin_index]
print("Wartość hmin:", hmin_midpoint)
```

2.1.16. Rysowanie wykresu

```
plt.figure(figsize=(10, 6))
plt.semilogy(num_nodes_list, errors_mid_list, label='Metoda mid-point')
plt.semilogy(num_nodes_list, errors_trap_list, label='Metoda trapezów')
plt.semilogy(num_nodes_list, errors_simp_list, label='Metoda Simpsona')
plt.semilogy(num_nodes_list, errors_gauss_list, label='Metoda Gaussa-Legendre\'a')
plt.title('Błąd bezwzględny w zależności od liczby ewaluacji funkcji podcałkowej')
plt.xlabel('Liczba ewaluacji funkcji podcałkowej')
plt.ylabel('Błąd bezwzględny')
plt.xscale('log')
plt.yscale('log')
plt.legend()
plt.grid(True)
plt.show()
```

3. Wykresy

3.1. Wykres błędu względnego w zależności od liczby ewaluacji dla metody trapezów, prostokątów, Simpsona, Gaussa-Legendre'a



Wykres 1. Błąd względny w zależności od liczby ewaluacji dla wszystkich metod

4. Tabele

4.1. Tabela błędów względnych metody trapezów

Wartość m	Wartość błędu względnego metody trapezów
1	0.0132
2	0.0033
3	0.0008
4	0.0002
5	$5.1808 * 10^{-5}$
6	$1.2952 * 10^{-5}$
7	$3.2380 * 10^{-6}$
8	$8.0950 * 10^{-7}$
9	$2.0237 * 10^{-7}$
10	$5.0593 * 10^{-8}$
11	$1.2648 * 10^{-8}$
12	$3.16 * 10^{-9}$
13	$7.90 * 10^{-10}$
14	$1.97 * 10^{-10}$
15	$4.94 * 10^{-11}$
16	$1.235 * 10^{-11}$
17	$3.08 * 10^{-13}$
18	$7.719 * 10^{-13}$
19	$1.936 * 10^{-14}$
20	$4.834 * 10^{-14}$
21	$1.272 * 10^{-15}$
22	$3.109 * 10^{-15}$
23	$2.544 * 10^{-15}$
24	$2.685 * 10^{-15}$
25	$2.685 * 10^{-15}$

Tabela 1. Tabela błędów względnych metody trapezów

4.2. Tabela błędów względnych metody Simpsona

Wartość m	Wartość błędu względnego metody trapezów
1	0.0026
2	$7.647 * 10^{-6}$
3	$4.810 * 10^{-8}$
4	$7.527 * 10^{-10}$
5	$1.17 * 10^{-11}$
6	$1.83 * 10^{-13}$
7	$2.82 * 10^{-15}$
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	$1.41 * 10^{-16}$
17	0
18	0
19	0
20	$1.41 * 10^{-16}$
21	$2.82 * 10^{-16}$
22	$1.41 * 10^{-16}$
23	$4.24 * 10^{-15}$
24	$1.41 * 10^{-16}$
25	$4.24 * 10^{-16}$

Tabela 2. Tabela błędów względnych metody Simpsona

4.3. Tabela błędów względnych metody prostokątów

Wartość m	Wartość błędu względnego metody trapezów
1	0.328
2	0.198
3	0.110
4	0.0587
5	0.0302
6	0.0153
7	0.007
8	0.0038
9	0.0019
10	0.0009
11	0.0004
12	0.0002
13	0.0001
14	$6.1031 * 10^{-5}$
15	$3.0516 * 10^{-5}$
16	$1.5258 * 10^{-5}$
17	$7.6293 * 10^{-6}$
18	$3.8146 * 10^{-6}$
19	$1.907 * 10^{-6}$
20	$9.5367 * 10^{-7}$
21	$4.7683 * 10^{-7}$
22	$2.384 * 10^{-7}$
23	$1.192 * 10^{-7}$
24	$5.9604 * 10^{-8}$
25	$2.980 * 10^{-8}$

Tabela 2. Tabela błędów względnych metody prostokątów

4.4. Tabela błędów względnych metody Gaussa-Legrange'a

Wartość m	Wartość błędu względnego metody trapezów
1	0.018
2	0.0018
3	0.00016
4	$6.1279 * 10^{-6}$
5	$4.3624 * 10^{-9}$
6	$1.349 * 10^{-8}$
7	$8.4796 * 10^{-10}$
8	$2.2496 * 10^{-11}$
9	$4.9319 * 10^{-13}$
10	$8.0574 * 10^{-14}$
11	$3.675 * 10^{-15}$
12	$1.4135 * 10^{-16}$
13	0
14	$2.8271 * 10^{-16}$
15	$1.4135 * 10^{-16}$
16	0
17	$1.413 * 10^{-16}$
18	0
19	$1.413 * 10^{-16}$
20	$2.8271 * 10^{-16}$
21	$1.4135 * 10^{-16}$
22	$2.82 * 10^{-16}$
23	$1.41 * 10^{-16}$
24	$1.413 * 10^{-16}$
25	0

Tabela 2. Tabela błędów względnych metody Gaussa-Legrange'a

4.5. Tabela rzędu zbieżności dla metod

Metoda	Rząd zbieżności
Trapezów	2.61
Simpsona	6.159
Prostokątów	1.44
Gauss-Lengrange'a	4.81

Tabela 3. Tabela rzędu zbieżności dla wszystkich metod

4.6. Tabela wartości, poniżej której zmniejszanie kroku h nie zmniejsza już błędu kwadratury dla metody trapezów oraz Simpsona

Metoda	H_min
Trapezów	$1.19 * 10^{-7}$
Simpsona	0.003
Prostokątów	$2,98 * 10^{-8}$
Gauss-Lengrange'a	0.001

Tabela 4. Tabela wartości h_min dla wszystkich metod

5. Wnioski

Empiryczne rzędy zbieżności jakie zostały obliczone w tabeli nr. 3 są bardzo bliskie teoretycznym wartościom, potwierdza to teoretyczne założenia dotyczące rzędu zbieżności tym metodą.

Rzędy te nie są idealnymi wartościami ale różnią się o małe wartości liczbowe może to wynikać z niedokładności obliczeń numerycznych lub innych czynników wpływających na dokładność wyniku.

Zarówno metoda Simpsona jak i Gaussa-Lagrange'a są skutecznymi metodami całkowania numerycznego. Empiryczne rzędy zbieżności dla obu metod są zgodne z teoretycznymi oczekiwaniami, co potwierdza ich poprawność i skuteczność. Minimalne wartości kroku h_{\min} dla obu metod są na akceptowalnym poziomie, co oznacza, że metody te są w stanie osiągnąć wysoką dokładność wyników dla dostatecznie małych wartości kroku h .

Wykres wartości bezwzględnej błędów względnego w zależności od liczby ewaluacji funkcji podcałkowej pokazuje, że błąd względny maleje wraz ze wzrostem liczby węzłów, co sugeruje, że metoda Gaussa-Legendre'a jest skuteczną metodą całkowania numerycznego, a tak samo dobrą jest metoda Simpsona.

6. Bibliografia

Wykład MOWNiT - prowadzony przez dr. Inż. K. Rycerz

Prezentacje – dr. Inż. M. Kuta

7. Dodatkowe informacje

Rozwiązanie obu zadań znajduje się odpowiednio w pliku ex1_ex2.ipynb.