

# MOwNiT laboratorium 5. - Sprawozdanie

Michał Szczurek  
Informatyka, WIEiT

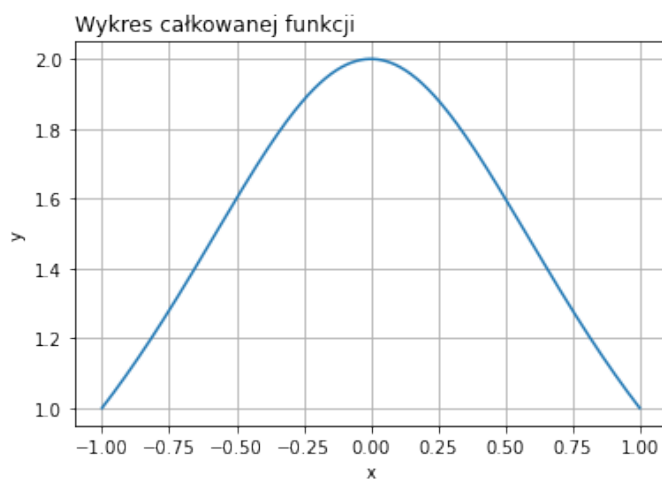
Grupa poniedziałek 12:50

## 1 Wstęp

Celem laboratorium było przetestowanie działania czterech różnych metod całkowania numerycznego:

- metody prostokątów
- metody trapezów
- metody Simpsona (wersji używającej parabol)
- metody Gaussa-Legendre'a

Metody te porównano poprzez obliczenie całki  $\int_{-1}^1 \frac{2}{1+x^2} dx$  i porównanie wyniku do wartości rzeczywistej w zależności od liczby ewaluacji funkcji podcałkowej  $n + 1$ . Skorzystano z faktu, że dokładna wartość wyrażenia wynosi  $\pi$ .

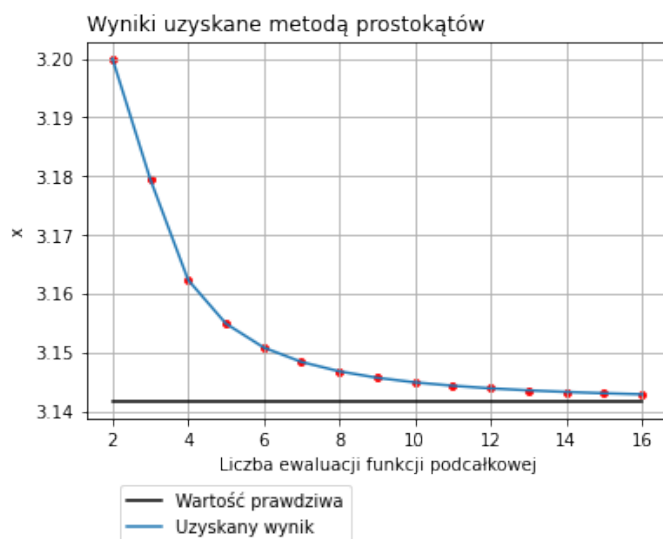


## 2 Całkowanie metodą prostokątów

Poniżej znajduje się kod funkcji wykonującej całkowanie metodą prostokątów ewaluując funkcję  $n+1$  razy, gdzie  $n$  to parametr wejściowy funkcji.

```
def midpoint(func, a, b, n):  
    res = 0  
    n = n+1 # we want n+1 evals, in this method there is (b-a)/h evals  
    h = (b-a)/n  
    for i in range(1, n+1):  
        p = func.subs(x, a + h/2 + (i-1)*h )  
        res += p*h  
  
    return res
```

Poniżej znajduje się wykres otrzymanych wyników dla funkcji, dla  $n+1$  z zakresu od 1 do 16.  $n+1 = 1$  pominięto, ponieważ wartość znacząco odstaje one od reszty wyników zaciemniając wykres. Wartości powyżej 16 nie są już rozróżnialne z racji, że różnica między wynikami zmniejsza się.



Dla tych wyników sporządzono również wykres błędów względnych:

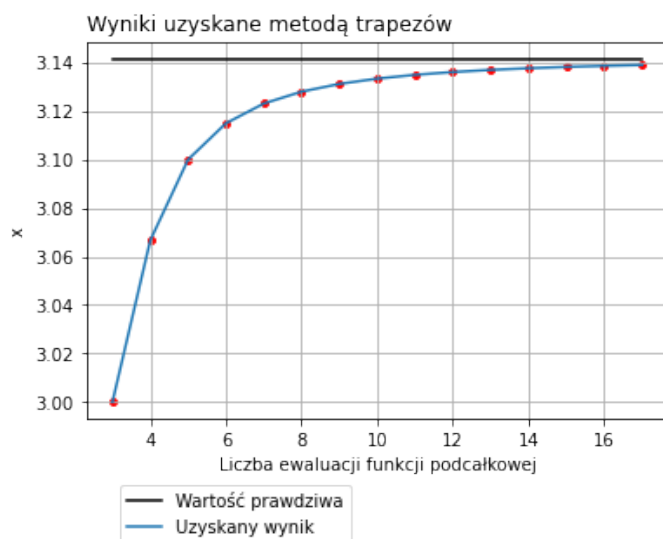


### 3 Całkowanie metodą trapezów

Poniżej znajduje się kod funkcji wykonującej całkowanie metodą trapezów ewaluując funkcję  $n+1$  razy, gdzie  $n$  to parametr wejściowy funkcji.

```
def trapezoid(func, a, b, n):  
    res = 0  
    h = (b-a)/n  
    for i in range(1, n+1):  
        p1 = func.subs(x, a + (i-1)*h)  
        p2 = func.subs(x, a + i*h)  
        res += (p1 + p2)*h/2  
  
    return res
```

Wykres wykonano dla liczby ewaluacji zaczynając od 3 z racji, że dla 2 wartość była wyraźnie odstająca i obniżała jakość wykresu.



Wykres błędów względnych wyglądał następująco:



## 4 Całkowanie metodą Simpsona

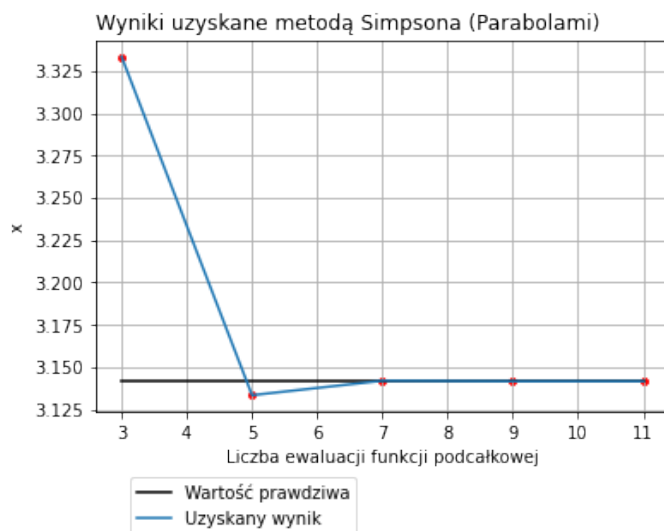
Inną zaimplementowaną metodą całkowania była metoda Simpsona, korzystająca z przybliżenia parabolami i zależności:

$$\int_{x_0}^{x_2} f(x)dx \approx \frac{h}{3}(y_0 + 4y_1 + y_2)$$

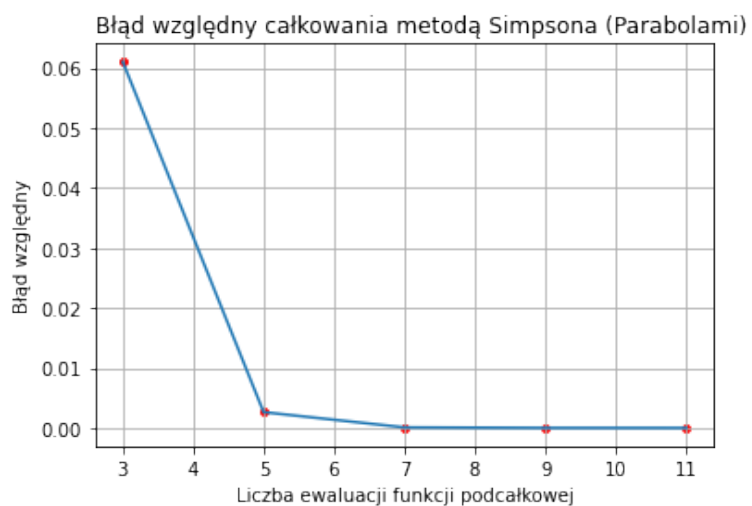
gdzie  $y_i = f(x_i)$ , a  $x_2 - x_1 = x_1 - x_0 = h$ . Metoda ta działa tylko dla nieparzystej liczby  $n+1$  zaczynając od 3.

```
def simpson(func, a, b, n):  
    res = 0  
    h = (b-a)/n  
    for i in range(0, int(n/2)):  
        l = func.subs(x, a + 2*h*i)  
        m = func.subs(x, a + 2*h*i + h)  
        r = func.subs(x, a + 2*h*i + 2*h)  
        res += l + 4*m + r  
    res *= h/3  
  
    return res
```

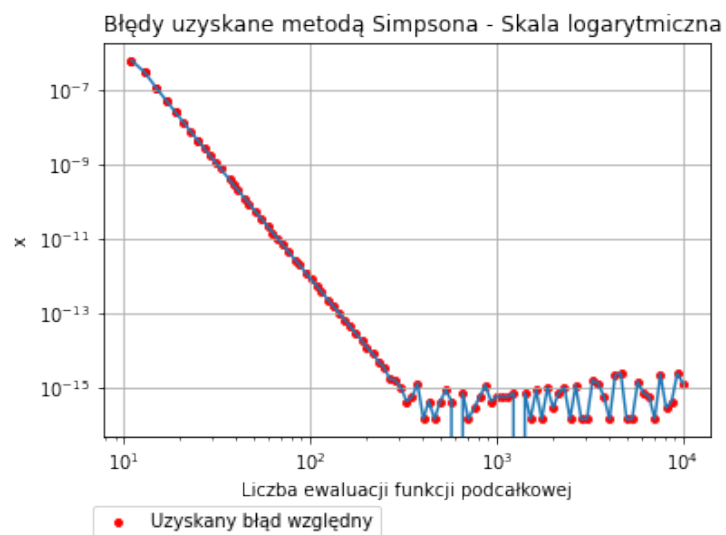
Wobec tego, że metoda działa tylko dla nieparzystych  $n+1$ , wobec czego kolejne wyniki bardzo szybko przybliżają się do wartości dokładnej - ciężko było sporządzić wykres na którym wiele punktów było by wyraźnie rozróżnialne.



Błąd względy wyglądał w następujący sposób:



Sporządziłem, również wykres błędu względnego z osiami logarytmicznymi:



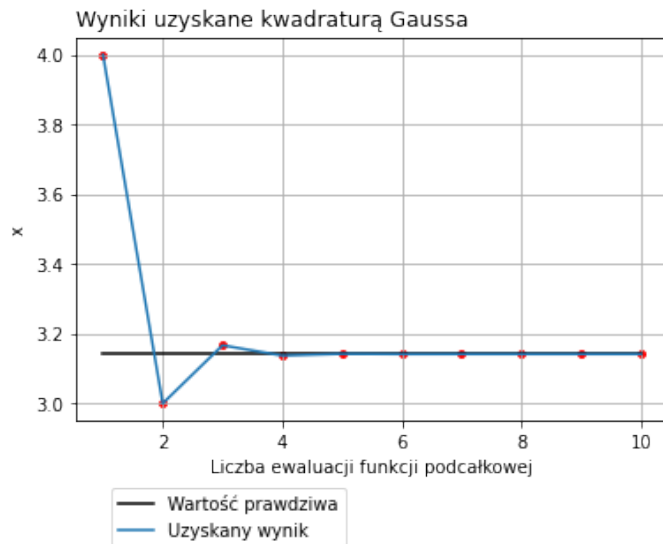
## 5 Całkowanie metodą Gaussa-Legendre'a

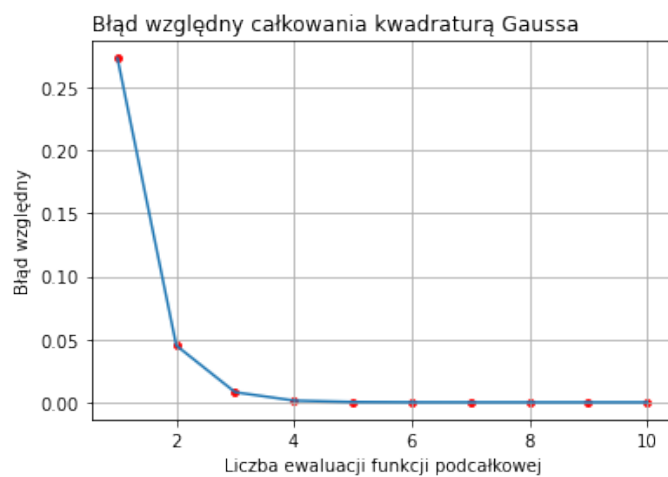
Ostatnią z zaimplementowanych metod była metoda Gaussa-Legendre'a. W celu użycia jej wykorzystano funkcję `legendre_poly` z biblioteki `sympy`, która pozwoliła na obliczenie wielomianu Legendre'a stopnia  $n+1$ , jego pochodnej i pierwiastków. Funkcję ewaluowano w tych właśnie pierwiastkach stosując przy tym funkcję wagową

$$w(x) = \frac{2}{(1 - x_i^2)(P'_n(x_i))^2}$$

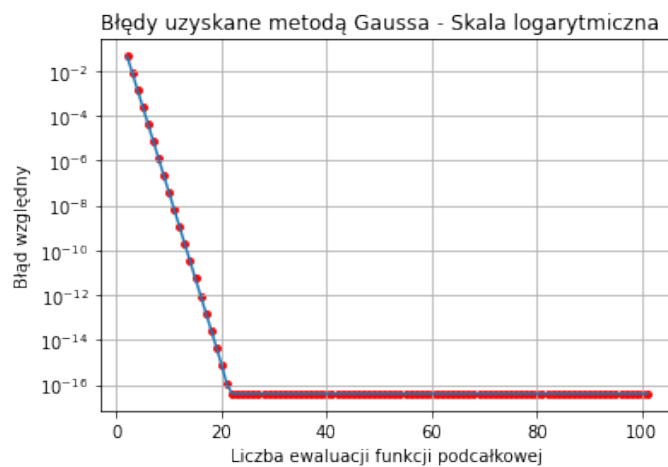
```
def gauss(f, n): # version for interval (-1,1)
    poly = legendre_poly(n+1, x, polys=True)
    poly_d = diff(poly, x)
    roots = poly.real_roots()
    res = 0
    for root in roots:
        if type(root) != Mul and root != 0:
            x_i = root.eval_rational()
        else:
            x_i = root
        w = 2/((1 - x_i**2) * poly_d.subs(x, x_i)**2)
        res += w * f.subs(x, x_i)
    return N(res)
```

Niezależnie od wyboru liczby punktów, w który ewaluowana ma być funkcja podcałkowa, wyniki bardzo szybko zbiegały do wartości rzeczywistej - ciężko było sporządzić czytelne wykresy wartości i błędów, wobec czego wykresy zrobiono dla liczby ewaluacji zaczynając od 1:





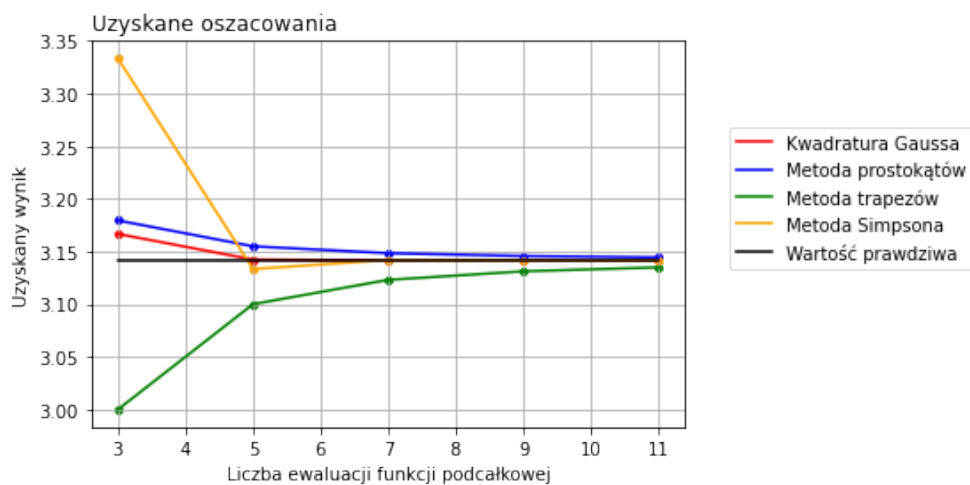
Sporządziłem, również wykres błędu względnego z logarytmiczną osią Y:



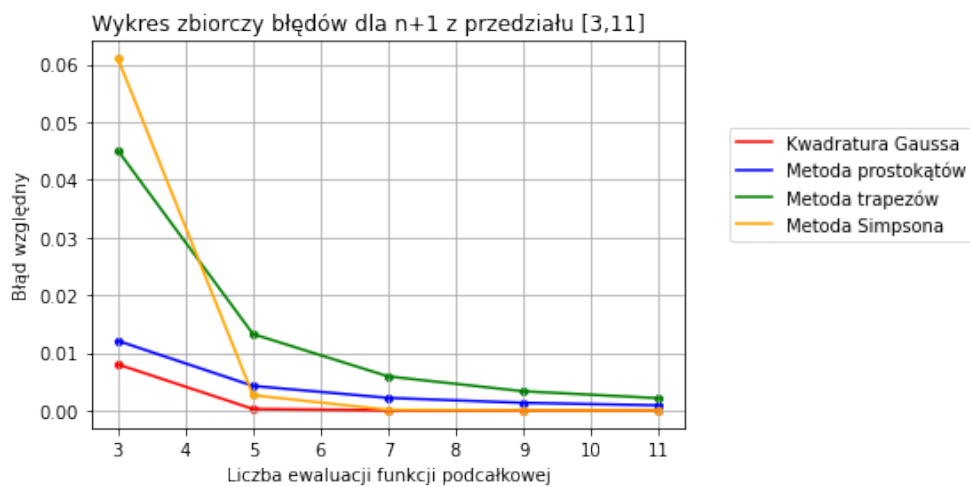


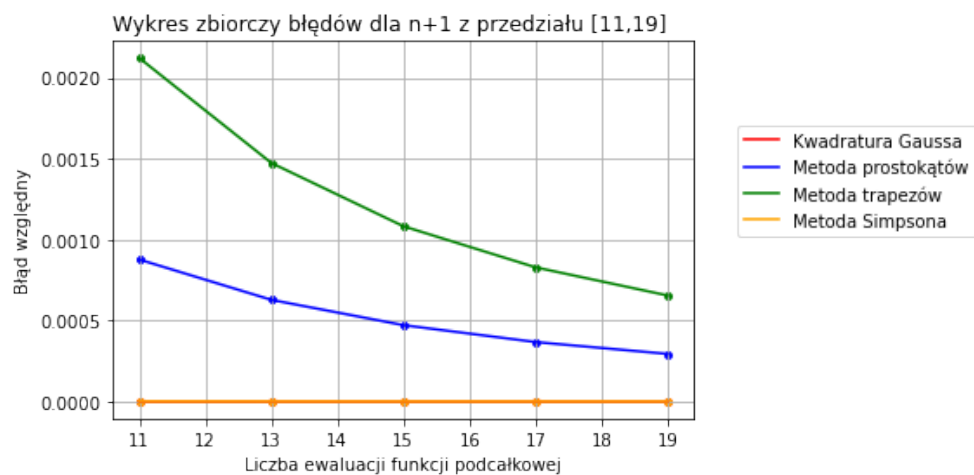
## 6 Wykresy porównujące wyniki

Poniżej znajdują się wykresy porównujące wartości uzyskanych wyników dla nieparzystej liczby ewaluacji funkcji podcałkowej (tak, by metoda Simpsona miała sens):

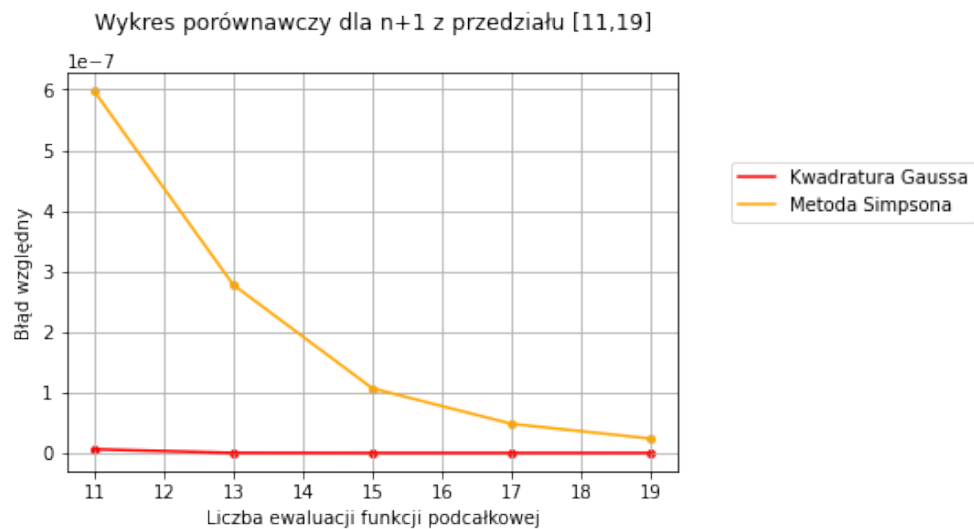


Sporządzono również wykresy porównujące błąd względny dla dwóch różnych zakresów liczb ewaluacji funkcji:





Na powyższym wykresie błąd uzyskany w metodzie Simpsona pokrywa się z tym uzyskanym w metodzie Gaussa-Legendre'a. Wobec tego sporządzono wykres porównujący tylko te dwie metody:



## 7 Wnioski

- dla  $n \geq 5$  metody w kolejności od najdokładniejszej do najmniej dokładnej to: kwadratura Gaussa, metoda Simpsona, metoda prostokątów i metoda trapezów.
- Metoda Simpsona początkowo sprawowała się najgorzej - wynika to z faktu, że do konstrukcji paraboli potrzeba aż 3 punktów w przeciwieństwie do trapezu (2) lub prostokąta (1).
- Metoda Gaussa-Legendre'a była zawsze najlepsza dla badanych przypadków.
- Metoda trapezów dawała zawsze mniejszy wynik od rzeczywistego, na skutek wklęsłości funkcji - bok trapezu znajdował się zawsze pod funkcją całkowaną.
- Metoda prostokątów zawsze dawała wynik większy od dokładnego, również jest to konsekwencją wklęsłości punktu. Aby zrozumieć ten fakt, można podzielić graficznie prostokąty na trójkąty. Analogiczne rozumowanie (tylko dla funkcji wypukłej) znajduje się na stronie <https://www.shmoop.com/definite-integrals/midpoint-estimate.html>
- Wartości uzyskane metodą Simpsona i kwadraturą Gaussa były naprzemiennie większe i mniejsze od oczekiwanego wyniku.
- Wadą metody Gaussa-Legendre'a jest wyższy poziom skomplikowania jak i dłuższy czas obliczania wyniku.
- Od pewnego momentu błąd aproksymacji metodą Simpsona nie zmniejszał się, a oscylował wokół wartości  $10^{-14}$ .
- Od pewnego momentu błąd aproksymacji metodą Gaussa był stały i w przybliżeniu równy  $10^{-16}$ . Jest to wynikiem skończonej dokładności pierwiastków wielomianu Legendre'a jak i liczby  $\pi$