MOWNIT - Sprawozdanie 7
Kwadratury Gaussa-Legendre’a

Mikołaj Wróblewski

1. Policzenie całki za pomocą metody Gaussa-Legendre’a jest dobrze aproksymowane przez wielomiany na przedziale [-1,1], razem ze związanymi wielomianami ortogonalnymi zwanymi wielomianami Legendre’a. Punkty aproksymacji oraz wagi z nimi związane dla wielomianów o stopniach 2-5 przedstawiają się następująco:

Number of points, n	Points, x_i	Approximately, x_i	Weights, w_i	Approximately, w_i
1	0	0	2	2
2	$\pm\frac{1}{\sqrt{3}}$	±0.57735	1	1
3	0	0	$\frac{8}{9}$	0.888889
	$\pm\sqrt{\frac{3}{5}}$	±0.774597	$\frac{5}{9}$	0.555556
4	$\pm\sqrt{\frac{3}{7}-\frac{2}{7}\sqrt{\frac{6}{5}}}$	±0.339981	$\frac{18+\sqrt{30}}{36}$	0.652145
	$\pm\sqrt{\frac{3}{7}+\frac{2}{7}\sqrt{\frac{6}{5}}}$	±0.861136	$\frac{18-\sqrt{30}}{36}$	0.347855
5	0	0	$\frac{128}{225}$	0.568889
	$\pm\frac{1}{3}\sqrt{5-2\sqrt{\frac{10}{7}}}$	±0.538469	$\frac{322+13\sqrt{70}}{900}$	0.478629
	$\pm\frac{1}{3}\sqrt{5+2\sqrt{\frac{10}{7}}}$	±0.90618	$\frac{322-13\sqrt{70}}{900}$	0.236927

Użyłem oczywiście wartości aproksymowanych.

Zaimplementowana przeze mnie metoda została rozszerzona dla dowolnego przedziału całkowania [a,b].
Poniżej wyniki testów dla funkcji, których całki mieliśmy wyliczyć, w przedziale [-1,1], wraz z błędami, oraz porównaniem błędów dla podstawowych kwadratur (przyjmując liczbę kroków równą 100000):

```
3 * x^3 - 1
error for rect_rule: 6.0000000021709354e-05
error for trapezoid rule: 3.863576125695545e-14
error for simpson's rule: 1.3333333341636333e-05
stopien wielomianu: 2
wynik: -2.0     error 0.0
stopien wielomianu: 3
wynik: -2.000001        error 1.000000000139778e-06
stopien wielomianu: 4
wynik: -2.0     error 0.0
stopien wielomianu: 5
wynik: -2.000001        error 1.000000000139778e-06
2 * x^2
error for rect_rule: 2.6665913921419815e-10
error for trapezoid rule: 1.3332757120565475e-10
error for simpson's rule: 2.6666133329689146e-05
stopien wielomianu: 2
wynik: 1.33333209       error 1.2433333329564533e-06
stopien wielomianu: 3
wynik: 1.3333355386875778       error 2.2053542447775243e-06
stopien wielomianu: 4
wynik: 1.3333328970764418       error 4.362568912785747e-07
stopien wielomianu: 5
wynik: 1.333333718270177        error 3.849368439556855e-07
4 * sin(x)
error for rect_rule: 6.731767878433713e-05
error for trapezoid rule: 7.026036653520751e-16
error for simpson's rule: 5.912899799416968e-16
stopien wielomianu: 2
wynik: 0.0      error 0.0
stopien wielomianu: 3
wynik: 0.0      error 0.0
stopien wielomianu: 4
wynik: 0.0      error 0.0
stopien wielomianu: 5
wynik: 0.0      error 0.0
```

Wyniki zamieściłem również w pliku tekstowym, dołączonym do archiwum ze sprawozdaniem.
Co ciekawe błąd dla pierwszej funkcji jest jednakowy w przypadku użycia wielomianów parzystego stopnia, również pomiędzy wielomianami nieparzystych stopni nie ma żadnej różnicy.
Widzimy natomiast, że jeżeli chodzi o porównanie z metodami podstawowymi, lepiej spisała się metoda trapezów - oczywiście w porównaniu z użyciem wielomianów nieparzystych stopni, wielomiany parzystych stopni dały idealny wręcz wynik.

Co do funkcji kwadratowej metoda Gaussa-Legendre'a spisała się nieco gorzej, możemy jednak zaobserwować zmniejszającą się wartość błędu, w zależności od stopnia wielomianu - im wyższy stopień, tym błąd jest miejszy.

Dla sinusoidy natomiast metody podstawowe spisały się gorzej - co prawda dla metody trapezów oraz simpsona błąd był niewielki, ale nadał wystąpił - natomiast metoda Gaussa-Legendre'a spisała się niezawodnie.

Następnie wykonałem testy dla kilku przedziałów innych niż przedział [-1,1].

2. Testy dla przedziałów innych niż [-1,1]

Na początku przyjąłem przedział [0,20]

```
3 * x^3 - 1
error for rect_rule: 2.3999879994953517
error for trapezoid rule: 6.0006859712302685e-06
error for simpson's rule: 1.5998426666483283
stopien wielomianu: 2
wynik: 119979.94404999999       error 0.05595000000903383
stopien wielomianu: 3
wynik: 119980.12923094098       error 0.12923094097641297
stopien wielomianu: 4
wynik: 119979.98036843985       error 0.019631560149719007
stopien wielomianu: 5
wynik: 119980.04731215796       error 0.04731215795618482
2 * x^2
error for rect_rule: 0.07999973332698573
error for trapezoid rule: 1.333346517640166e-07
error for simpson's rule: 0.05333279999467777
stopien wielomianu: 2
wynik: 5333.33209       error 0.0012433333331500762
stopien wielomianu: 3
wynik: 5333.337538687579        error 0.004205354245641502
stopien wielomianu: 4
wynik: 5333.332897076441        error 0.00043625689158943715
stopien wielomianu: 5
wynik: 5333.335718270177        error 0.0023849368435548968
4 * sin(x)
error for rect_rule: 0.0003651859925200007
error for trapezoid rule: 3.946130533449832e-09
error for simpson's rule: 0.00024346785144313543
stopien wielomianu: 2
wynik: -37.98999910239953       error 40.35767085514596
stopien wielomianu: 3
wynik: -21.949484912368675      error 24.317156665115107
stopien wielomianu: 4
wynik: 37.84242151626225        error 35.47474976351582
stopien wielomianu: 5
wynik: -15.713034464682401      error 18.080706217428833
```

Widzimy, że nasza funkcja radzi sobie dużo gorzej - w niektórych przypadkach błąd rzędu 1e-3 byłby nie dopuszczalny - dla sinusoidy natomiast wyniki w ogóle nie odzwierciedlają rzeczywistości. Warto byłoby również przetestować znacznie szerszy przedział całkowania, co również poczyniłem - [10,1000]

```
3 * x^3 - 1
error for rect_rule: 14849911.658203125
error for trapezoid rule: 36.7464599609375
error for simpson's rule: 9899862.892700195
stopien wielomianu: 2
wynik: 749999648815.5543        error 342694.44567871094
stopien wielomianu: 3
wynik: 750000790611.6206        error 799101.6206054688
stopien wielomianu: 4
wynik: 749999871266.45  error 120243.55004882812
stopien wielomianu: 5
wynik: 750000288858.0557        error 297348.0556640625
2 * x^2
error for rect_rule: 9898.977653384209
error for trapezoid rule: 0.01617288589477539
error for simpson's rule: 6600.595314502716
stopien wielomianu: 2
wynik: 666665849.1993638        error 150.80063617229462
stopien wielomianu: 3
wynik: 666666519.9563773        error 519.9563772678375
stopien wielomianu: 4
wynik: 666665947.0875468        error 52.912453174591064
stopien wielomianu: 5
wynik: 666666299.1627293        error 299.16272926330566
4 * sin(x)
error for rect_rule: 0.027098047431827865
error for trapezoid rule: 2.2892760860315775e-05
error for simpson's rule: 0.0036421600396820963
stopien wielomianu: 2
wynik: -2817.607185046103       error 2812.0013826246345
stopien wielomianu: 3
wynik: 2812.883259537131        error 2818.4890619585995
stopien wielomianu: 4
wynik: 931.3378765388574        error 936.943678960326
stopien wielomianu: 5
wynik: -906.0642012208041       error 900.4583987993354
```

Jak widzimy nasza metoda nie radzi sobie wcale, generuje bardzo duże błędy, oczywiście ilość kroków przy metodach podstawowych również ma znaczenie, dlatego przy nich również obserwujemy znaczne błędy - liczba podprzedziałów równa 100000 zdaje się być zdecydowanie za mała.

3. Podsumowanie

Metoda Gaussa-Legendre'a jest zdecydowanie dobra dla przedziału [-1,1]. Jednoznacznie możemy stwierdzić, że całki użytych do pomiarów funkcji są liczone całkiem dobrze. Rzec również trzeba, że wystarczy nam tylko kilka iteracji do policzenia całki z przyzwoitą dokładnością, zatem metoda ta definitywnie dobrze sprawdzi się w aplikacjach w których to czas - nie dokładność - gra główną rolę. Co do szerszych przedziałów wyniki jednoznacznie ukazują słabości tej metody - błędy są zdecydowanie za duże, abyśmy mogli wykorzystać metodę Gaussa-Legendre'a do przedziałów innych aniżeli [-1,1].