Stanisław Wilczyński*

Pracownia z analizy numerycznej
Sprawozdanie do zadania **P.1.12**

Wrocław, 2 grudnia 2015

Spis treści

1. Wstęp

Obliczanie całek jest problemem nie tylko teoretycznym. Jest one ważne ze względu na swoje zastosowania w praktyce w dziedzinach takich jak fizyka,chemia, budownictwo, czy analiza ryzyka finansowego. Obliczenie jednej czy dwóch prostych całek nie jest problem dla przeciętnych studentów pierwszego roku dowolnych studiów ścisłych, jednak w przypadku poważniejszych obliczeń wymagana jest pomoc komputera. Niestety, jednak przy każdym użyciu takiego sprzętu musimy liczyć się z możliwym błędem związanym np. ze zjawiskiem utraty cyfr znaczących. Powszechnie znane metody Simpsona czy trapezów są dokładne, ale wymagają obliczania wartości całkowanej funkcji w wielu punktach. W niektórych wypadkach, przy odpowiedniej postaci funkcji podcałkowej możliwe jest wykorzytsanie innego sposobu opartego na rekurencji. W tym sprawozdaniu zajmiemy się całką $\int_0^1 t^n e^t \mathrm{d}t$. Zaprezentujemy dwie metody do obliczania wartości tej całki dla kolejnych $n \in \mathbb{N}$.

2. Matematyczne podstawy

Niech (y_n) będzie ciągiem zdefiniowanym następująco:

$$y_n = \int_0^1 t^n e^t \mathrm{d}t.$$

Najpierw sprawdzimy że ten ciąg jest malejący:

$$\begin{aligned} y_n - y_{n-1} &= \int_0^1 t^n e^t \mathrm{d}t - \int_0^1 t^{n-1} e^t \mathrm{d}t = \int_0^1 t^{n-1}(t-1) e^t \mathrm{d}t \leqslant 0, \\ &\quad \text{bo } (t-1) \leqslant 0 \text{ dla każdego } t \in (0,1). \end{aligned}$$

Teraz pokażemy, że (y_n) zbiega do 0. Z twierdzenia o wartości średniej ([**?**, strona 102]):

* *E-mail*: `opos1@onet.eu`

$$1\int_0^1 t^n \mathrm{d}t \leqslant y_n \leqslant e\int_0^1 t^n \mathrm{d}t. \tag{1}$$

Otrzymujemy więc:

$$\frac{1}{n+1} \leqslant y_n \leqslant \frac{e}{n+1}. \tag{2}$$

Zauważmy, że

$$\lim_{n\to\infty} \frac{1}{n+1} = \lim_{n\to\infty} \frac{e}{n+1} = 0.$$

więc z twierdzenia o trzech ciągach([?, strona 6]) i nierówności (??) mamy:

$$\lim_{n\to\infty} y_n = 0.$$

Stosując całkowanie przez części znajdujemy rekurencyjny związek między y_n i y_{n-1}:

$$y_n = \int_0^1 t^n e^t \mathrm{d}t = t^n e^t|_{t=0}^{t=1} - \int_0^1 nt^{n-1}e^t \mathrm{d}t = e - n\int_0^1 t^{n-1}e^t \mathrm{d}t = e - ny_{n-1}. \tag{3}$$

Obliczamy jeszcze

$$y_0 = \int_0^1 e^t \mathrm{d}t = e - 1 = 1.718281828459045.... \tag{4}$$

3. Metoda I - naiwna

3.1. Opis i działanie metody I

W tej metodzie wykorzystamy związek rekurencyjny (??) do obliczania kolejnych wartości, tzn. mając dane y_{n-1} wyznaczamy y_n. Za pomocą załączonego programu *program.jl* policzyliśmy pierwsze 21 wyrazów naszego ciągu(w arytmetyce z podwójną precyzją). Niech $\tilde{y}_n$ oznacza obliczoną w ten sposób wartość n-tego wyrazu ciągu. Otrzymane wyniki prezentujemy w tabelce (w tej i każdej następnej tabelce rzeczywisty wynik $y_n$jest podawany do 16 miejsca po przecinku)[1]:

n	$\tilde{y}_n$	y_n	liczba tych samych cyfr znaczących w y_n i $\tilde{y}_n$
0	1.7182818284590452	1.7182818284590452	16
2	0.7182818284590451	0.7182818284590452	15
4	0.4645364561314058	0.4645364561314071	14
6	0.3446845416469490	0.3446845416469873	13
8	0.2743615330158317	0.2743615330179760	11
10	0.2280015152934535	0.2280015154864418	9
12	0.1950999056863769	0.1950999311608206	7
14	0.1705190649530128	0.1705237013017674	4
16	0.1503481618374036	0.1514608855385011	2
18	-0.204253561558256	0.1362398909775906	0
20	-129.2637081328594	0.1238038307625699	0

Zauważamy, że liczba takich samych cyfr znaczących w liczbach y_n i $\tilde{y}_n$ maleje o 1 przy prawie każdej iteracji. Oznacza to, że wyniki $\tilde{y}_n$ dla $n \geqslant 18$ nie mają nawet jednej takiej samej cyfry jak rzeczywiste wartości y_n. Co więcej ciąg $\tilde{y}_n$ nie jest nawet nieujemny tak jak y_n. Skąd bierze się tak ogromna niedokładność obliczeń?

[1] Wartości y_n zostały obliczone za pomocą pakietu Wolfram Mathematica 10

3.2. Analiza metody I

Niech d_n oznacza błąd bezwzględny, tzn. $d_n = |\tilde{y}_n - y_n|$. Zgodnie z definicją zawartą w [?, strona 48] mówimy, że algorytm jest numerycznie niestabilny, jeśli małe błędy popełnione na początku obliczeń powodują wypaczanie wyniku w kolejnych krokach algorytmu. Formalnie, aby sprawdzić czy algorytm jest niestabilny numerycznie należy sprawdzić błąd względny, czyli w naszym wypadku $\frac{d_n}{y_n}$. Przeprowadzimy analizę błędu bezwzględnego. Zauważmy, że błąd d_2 jest rzędu precyzji naszej arytmetyki DOUBLE, tzn. $d_2 \approx 10^{-16}$. Co więcej:

$$d_n = |\tilde{y}_n - y_n| = |(e - n\tilde{y}_{n-1}) - (e - ny_{n-1})| = |n(\tilde{y}_{n-1} - y_{n-1})| = nd_{n-1}. \tag{5}$$

Powyższy rachunek daje nam tylko przybliżoną wartość błędu, gdyż pomijamy tu niedokładności wynikające z wykonywania operacji arytmetycznych. Mimo wszystko, korzystając z powyższej równości możemy obliczyć przybliżony wzór na d_n:

$$d_n = nd_{n-1} = n(n-1)d_{n-2} = ... = \frac{1}{2}n!d_2. \tag{6}$$

Korzystając z tego wzoru obliczamy $d_{20} = \frac{1}{2}20!d_2 \approx \frac{1}{2}2,4 \times 10^{18} \times 10^{-16} \approx 120$ i zauważamy, że dla 20 wyrazu naszego ciągu obliczona wartość nie jest nawet bliska wartości rzeczywistej. Ten wniosek wynika z tego, że błąd względny wynosi aż $\frac{d_{20}}{y_{20}} \approx 1044,13$, co oznacza,że wartość d_{20} jest blisko 1000 razy większa niż rzeczywista wartość y_{20}. Oczywiście, mając na uwadze wzór (??) wnioskujemy, że zastosowany tutaj algorytm wyliczania y_n jest numerycznie niestabiny.

4. Metoda II - ulepszona

4.1. Opis metody II

Druga metoda jest oparta na algorytmie Millera ([?, strony 23-28]). Zauważmy, że korzystjąc z nierówności (??) możemy stwierdzić, że ciąg (y_n) jest wolno zbieżny, a co za tym idzie, dla dużych N, $y_N \approx e - Ny_N$. Z tego związku możemy policzyć, że $y_N \approx \frac{e}{N+1}$. Stosując oznaczenia z poprzedniego rozdziału kładziemy:

$$\tilde{y}_N = \frac{e}{N+1}.$$

Wtedy $d_N \leqslant \frac{e-1}{N+1}$ (nierówność (??)) Oczywiście, dzięki związkowi rekurencyjnemu(??) możemy się cofać, tzn. obliczyć y_{n-1} w zależności od y_n. Przekształcając równanie (??) otrzymujemy:

$$y_{n-1} = \frac{e - y_n}{n}. \tag{7}$$

Za pomocą powyższej równości, wykorzystując załączony program *program.jl* obliczamy wartości wyrazów ciągu $\tilde{y}_n$ zaczynając od N=20. Wyniki prezentujemy w poniższej tabelce:

N	$\tilde{y}_N$	y_N	liczba tych samych cyfr znaczących w y_N i $\tilde{y}_N$
20	0.1294419918313831	0.1238038307625699	2
18	0.1362547282435611	0.1362398909775906	4
16	0.1514609340262984	0.1514608855385011	6
14	0.1705237015037999	0.1705237013017674	9
12	0.1950999311619307	0.1950999311608206	11
10	0.2280015154864502	0.2280015154864418	13
8	0.2743615330179761	0.2743615330179760	15
6	0.3446845416469873	0.3446845416469873	16
4	0.4645364561314070	0.4645364561314071	16
2	0.7182818284590452	0.7182818284590452	16
0	1.7182818284590452	1.7182818284590452	16

Porównując tabelki dla metody I i II zauważamy, że II jest dokładniejsza o około 5 cyfr. Ponieważ w metodzie II błąd przy obliczaniu pierwszej wartości zależy od tego jakie N wybraliśmy, stwierdzamy, że dla $N \geqslant 20$ dokładność będzie jeszcze lepsza, a dokładność metody I ulega pogorszeniu wraz ze wzrostem N. Skąd bierze się tak znaczna różnica wyniku dla obu metod?

4.2. Analiza metody II

Podobnie jak w metodzie I przeprowadzimy analizę błędu bezwzględnego. Już w poprzednim rozdziale zauważyliśmy, że jeśli rozpoczynamy działanie naszej metody od N, to

$$d_N \leqslant \frac{e-1}{N+1}.$$

Skoro mamy wzór rekurencyjny na wcześniejsze wyrazy ciągu (**??**)„ możemy również obliczyć przybliżone błędy bezwzględne (ponownie ignorujemy błąd wynikający z wykonywania operacji arytmetycznych):

$$d_{N-1} = |\tilde{y}_{N-1} - y_{N-1}| = \left|\frac{e-\tilde{y}_N}{N} - \frac{e-y_N}{N}\right| = \left|\frac{\tilde{y}_N - y_N}{N}\right| = \frac{d_N}{N}. \tag{8}$$

Dalej korzystając z powyższej równości otrzymujemy:

$$d_{N-k} = \frac{(N-k)!}{N!} d_N. \tag{9}$$

Gdy $N = 20$, $d_{20} \approx 3 \times 10^{-3}$
Oznacza to, że już $d_8 = \frac{8!}{20!} d_N \approx 1,6 \times 10^{-14} \times 6 \times 10^{-3} \approx 1 \times 10^{-16}$. Jeśli zaglądniemy do tabelki jest to nawet dokładnie błąd, który otrzymaliśmy przy obliczaniu $\tilde{y}_8$.

5. Podsumowanie

Z przeprowadzoenj analizy wynika, że metoda I nie nadaje się do obliczania dokładnych wartości wyrazów ciągu (y_n). Jednakże, jak łatwo zauważyć, metoda II pozwala nam obliczać każdy wyraz ciągu (y_n) z dokładnością do 16. miejsca po przecinku. Korzystając ze wzoru (**??**)

otrzymujemy:

$$d_n = \frac{n!}{(n+k)!} d_{n+k}, \ \forall k \in N. \tag{10}$$

W takim razie, by zachodziła nierówność $d_n < 10^{-16}$ wystarczy znaleźć takie k, dla którego

$$\frac{n!}{(n+k)!} d_{n+k} \leqslant \frac{n!}{(n+k)!} \times \frac{e-1}{n+1} < 10^{-16}.$$

Oczywiście takie k można znaleźć dla każdego n. Co więcej, im większe n tym mniejsze k. Przykładowo dla $n = 100$ $k = 9$, a dla $n = 1000$ $k = 6$.

Podsumowując, aby za pomocą metody II obliczyć y_n z dużą dokładnością, wystarczy rozpocząć obliczenia od wyrazu ciągu o indeksie o kilka(-naście) większym. Ponadto, aby zwiększyć dodatkowo dokładność metody II do większej liczb cyfr po przecinku, wystarczy wybrać dla danego n większe niż wcześniej k w taki sposób, aby $\frac{n!}{(n+k)!} \times \frac{e-1}{n+1} < 10^{-p}$, gdzie p jest liczbą cyfr po dziesiętnych, które chcemy wyznaczyć.

Literatura

[1] R. Szwarc, Skrypt do wykładu z analizy matematycznej I `http://www.math.uni.wroc.pl/~szwarc/pdf/AnalizaISIM1.pdf`.

[2] D.Kincaid, W.Cheney, Numerical Analysis. Mathematics of scientific computing, Brooks/Cole Publishing Company,Pacific Groove California, 1991.

[3] J.Wimp, Computation with recurrence relations, Boston:Pitman Advanced Publ. Program, 1984.