

Analiza numeryczna (M) - Pracownia 1

Zadanie P1.8

Analiza i wyliczanie przybliżonych wartości elementów ciągu

Jakub Kuciński, prowadzący Witold Karczewski

Wrocław, Październik 30, 2019

Spis treści

1	Wprowadzenie	1
2	Twierdzenia wykorzystane w sprawozdaniu	2
3	Własności ciągu $\{y_n\}$	3
3.1	Monotoniczność	3
3.2	Ograniczenia górne i dolne na wyrazy ciągu	3
3.3	Zbieżność	3
3.4	Zależność rekurencyjna wyrazów ciągu	4
4	I metoda: wyliczanie kolejnych elementów ciągu przy użyciu zależności rekurencyjnej	4
5	II metoda: wyliczanie poprzednich elementów ciągu przy użyciu zależności rekurencyjnej	5
6	Źródła różnicy dokładności metod I i II	8
6.1	Niedokładność metody I	8
6.2	Dokładność metody II	9
7	Podsumowanie	10

1 Wprowadzenie

Obliczanie wartości wyrazów ciągu jest jednym z podstawowych zagadnień metod numerycznych. Można spotkać się z problemem obliczania wyrazów ciągu o elementach jawnej postaci lub też ciągu o elementach spełniających pewną zależność rekurencyjną. W każdym przypadku wybór metody zależy ściśle od rodzaju ciągu, z którym mamy do czynienia, jego własności oraz wzoru, którym jest opisany. Celem tego sprawozdania jest zbadanie własności ciągu $y_n = \int_0^1 t^n e^t dt$ oraz próba jak najdokładniejszego wyliczenia wartości jego pierwszych 20 wyrazów, a także stwierdzenie źródła niedokładności i dokładności zastosowanych metod.

2 Twierdzenia wykorzystane w sprawozdaniu

Twierdzenie 1. Jeżeli funkcje f i g są całkowne na przedziale $[a, b]$ to funkcje $f(x) \pm g(x)$ też są całkowne oraz

$$\int_a^b f(x)dx \pm \int_a^b g(x)dx = \int_a^b (f(x) \pm g(x))dx \quad (1)$$

Dowód. Przedstawiony w [1], strona 135. □

Twierdzenie 2. Jeżeli funkcje f i g są całkowne na przedziale $[a, b]$ i dla wszystkich $x \in [a, b]$ mamy $f(x) \leq g(x)$ to

$$\int_a^b f(x)dx \leq \int_a^b g(x)dx \quad (2)$$

Jeśli dodatkowo $f(x) < g(x)$ poza skończoną liczbą argumentów z przedziału $[a, b]$ to

$$\int_a^b f(x)dx < \int_a^b g(x)dx \quad (3)$$

Dowód. Przedstawiony w [1], strona 136. □

Twierdzenie 3. Zasadnicze twierdzenie rachunku różniczkowego i całkowego.

Jeśli funkcja $f(x)$ jest całkowna na $[a, b]$ oraz $F(x)$ jest funkcją pierwotną do $f(x)$, to

$$\int_a^b f(x)dx = F(b) - F(a) = F(x) \Big|_a^b \quad (4)$$

Dowód. Przedstawiony w [2], strona 113. □

Twierdzenie 4. Całkowanie przez części.

Jeżeli funkcje f i g są ciągle oraz f' i g' są całkowne na przedziale $[a, b]$, to

$$\int_a^b f'(x)g(x)dx = f(x)g(x) \Big|_a^b - \int_a^b f(x)g'(x)dx \quad (5)$$

Dowód. Wiemy, że $(f(x)g(x))' = f'(x)g(x) + f(x)g'(x)$. Z zasadniczego twierdzenia rachunku różniczkowego i całkowego (3) otrzymujemy:

$$f(x)g(x) \Big|_a^b = \int_a^b (f'(x)g(x) + f(x)g'(x))dx \stackrel{(1)}{=} \int_a^b f'(x)g(x)dx + \int_a^b f(x)g'(x)dx \quad (6)$$

Po odpowiednim uporządkowaniu elementów dostajemy tezę. □

Twierdzenie 5. Twierdzenie o wartości średniej.

Jeśli funkcje f i g są całkowne na $[a, b]$, $g(x) \geq 0$ dla $x \in [a, b]$ oraz $m \leq f(x) \leq M$ dla $x \in [a, b]$ to

$$\int_a^b f(x)g(x)dx = \lambda \int_a^b g(x)dx \quad (7)$$

gdzie $\lambda \in [m, M]$.

Dowód. Przedstawiony w [2], strona 118. □

Twierdzenie 6. Twierdzenie o 3 ciągach.

Jeśli ciągi a_n, b_n, c_n spełniają $a_n \leq b_n \leq c_n$ oraz $\lim_{n \rightarrow \infty} a_n = \lim_{n \rightarrow \infty} b_n = g$ to

$$\lim_{n \rightarrow \infty} c_n = g \quad (8)$$

Dowód. Przedstawiony w [1], strona 38. □

3 Własności ciągu $\{y_n\}$

3.1 Monotoniczność

W celu zbadania monotoniczności ciągu $y_n = \int_0^1 t^n e^t dt$ przeanalizujemy różnicę dwóch kolejnych wyrazów tego ciągu:

$$y_n - y_{n+1} = \int_0^1 t^n e^t dt - \int_0^1 t^{n+1} e^t dt \stackrel{(1)}{=} \int_0^1 (t^n e^t - t^{n+1} e^t) dt = \int_0^1 t^n e^t (1-t) dt \quad (9)$$

Zauważmy, że wyrażenie podcałkowe $t^n e^t (1-t)$ dla $t \in (0, 1)$ przyjmuje wartości dodatnie oraz dla $t = 1$ i $t = 0$ jest równe 0. Stąd z twierdzenia (2) otrzymujemy:

$$0 = \int_0^1 0 \cdot dt < \int_0^1 t^n e^t (1-t) dt = y_n - y_{n+1} \quad (10)$$

Zatem $y_n - y_{n+1} > 0$, czyli ciąg $\{y_n\}$ jest ściśle malejący.

3.2 Ograniczenia górne i dolne na wyrazy ciągu

Niech $f(t) = e^t$ oraz $g(t) = t^n$. Wtedy f i g są całkowalne na $[a, b]$, $g(t) \geq 0$ dla $t \in [0, 1]$ oraz $1 \leq f(t) \leq e$ dla $t \in [0, 1]$, bo $f(t)$ ściśle rosnąca i ciągła na $[0, 1]$ ($f(0) = 1$ oraz $f(1) = e$). Zatem z twierdzenia o wartości średniej (5) otrzymujemy:

$$1 \int_0^1 g(t) dt \leq \int_0^1 f(t)g(t) dt \leq e \int_0^1 g(t) dt \quad (11)$$

$$\int_0^1 t^n dt \leq \int_0^1 f(t)g(t) dt \leq e \int_0^1 t^n dt \quad (12)$$

$$\frac{t^{n+1}}{n+1} \Big|_0^1 \leq \int_0^1 f(t)g(t) dt \leq e \frac{t^{n+1}}{n+1} \Big|_0^1 \quad (13)$$

$$\frac{1}{n+1} \leq \int_0^1 f(t)g(t) dt \leq \frac{e}{n+1} \quad (14)$$

Otrzymaliśmy więc ograniczenie na wyrazy ciągu $\{y_n\}$

$$\frac{1}{n+1} \leq y_n \leq \frac{e}{n+1} \quad (15)$$

3.3 Zbieżność

Pokażemy zbieżność ciągu $\{y_n\}$ do zera. Przypomnijmy udowodnione w poprzednim podrozdziale nierówności:

$$\frac{1}{n+1} \leq y_n \leq \frac{e}{n+1} \quad (16)$$

Zauważmy, że

$$\lim_{n \rightarrow \infty} \frac{1}{n+1} = \lim_{n \rightarrow \infty} \frac{e}{n+1} = 0 \quad (17)$$

Zatem z twierdzenia o trzech ciągach dostajemy

$$\lim_{n \rightarrow \infty} y_n = 0 \quad (18)$$

3.4 Zależność rekurencyjna wyrazów ciągu

Pokażemy, że między kolejnymi elementami ciągu $\{y_n\}$ zachodzi następujący związek rekurencyjny:

$$y_{n+1} = e - (n+1)y_n \quad (19)$$

W tym celu przekształcimy wzór na y_{n+1} przy użyciu twierdzenia o całkowaniu przez części (4):

$$\begin{aligned} y_{n+1} &= \int_0^1 t^{n+1} e^t dt = \int_0^1 t^{n+1} (e^t)' dt \stackrel{(4)}{=} t^{n+1} e^t \Big|_0^1 - \int_0^1 (t^{n+1})' e^t dt = \\ &= 1^{n+1} e^1 - 0^{n+1} e^0 - \int_0^1 (n+1) t^n e^t dt = e - (n+1) \int_0^1 t^n e^t dt = e - (n+1) y_n \end{aligned} \quad (20)$$

4 I metoda: wyliczanie kolejnych elementów ciągu przy użyciu zależności rekurencyjnej

Wyliczmy wartość początkową y_0 bezpośrednio ze wzoru ciągu $\{y_n\}$, a następnie korzystając z tego wyniku wyrazy y_1, y_2, \dots, y_{20} .

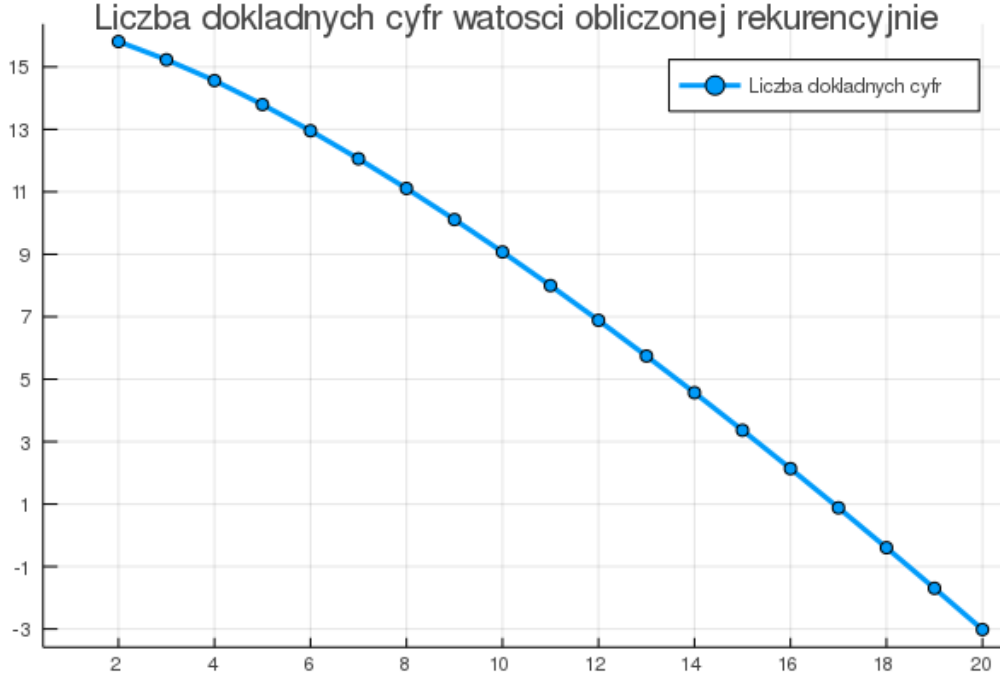
$$y_0 = \int_0^1 t^0 e^t dt = \int_0^1 e^t dt = e^t \Big|_0^1 = e - 1 \quad (21)$$

Przedstawmy uzyskane wartości w porównaniu do wyników podanych przez Wolfram Alpha.

Numer wyrazu	Wyliczone przez program	Podane przez Wolfram Alpha
1	1.0	1.0
2	0.7182818284590451	0.7182818284590452
3	0.5634363430819098	0.5634363430819095
4	0.4645364561314058	0.4645364561314071
5	0.395599547802016	0.3955995478020096
6	0.34468454164694906	0.34468454164698736
7	0.30549003693040166	0.30549003693013366
8	0.27436153301583177	0.2743615330179761
9	0.24902803131655915	0.24902803129726034
10	0.22800151529345358	0.2280015154864418
11	0.21026516023105568	0.21026515810818538
12	0.19509990568637692	0.19509993116082064
13	0.18198305453614516	0.18198272336837698
14	0.17051906495301283	0.17052370130176742
15	0.1604958541638526	0.160426308932534
16	0.15034816183740363	0.15146088553850118
17	0.16236307722318344	0.14344677430452527
18	-0.2042535615582568	0.1362398909775906
19	6.599099498065924	0.12972389988482377
20	-129.26370813285942	0.12380383076256996

Zauważmy, że według obliczonych przez nasz program wartości mamy $y_{17} > y_{16}$, chociaż według udowodnionej wcześniej monotoniczności ciągu $\{y_n\}$ (3.1) powinniśmy dostać $y_{17} < y_{16}$. Podobnie dostaliśmy, że $y_{19} > y_1$. Dostaliśmy również $y_{18}, y_{20} < 0$, co jest sprzeczne z faktem $y_n = \int_0^1 t^n e^t dt > 0$. Na podstawie przeprowadzonych obliczeń można więc przypuszczać, że wraz ze wzrostem numeru wyliczanego elementu rośnie również niedokładność obliczeń.

Sprawdźmy, jak zachowuje się liczba dokładnych cyfr wartości wyrazów otrzymanych przez nasz program, względem wyników podanych przez Wolfram Alpha, które uznamy tutaj za wyniki dokładne.



Na podstawie wykresu można podejrzewać, że liczba dokładnych cyfr dziesiętnych maleje nieliniowo dla kolejnych wyrazów ciągu, przy czym od wyrazu y_{18} otrzymana wartość nie ma już żadnych cyfr dokładnych.

Wniosek 1. Obliczanie następujących po sobie elementów y_0, y_1, \dots, y_{20} ciągu $\{y_n\}$ przy użyciu wzoru rekurencyjnego jest niestabilne, ponieważ liczba dokładnych cyfr szybko zmniejsza się wraz z obliczaniem każdego kolejnego wyrazu.

5 II metoda: wyliczanie poprzednich elementów ciągu przy użyciu zależności rekurencyjnej

Zauważmy, że wobec nierówności

$$\frac{1}{n+1} \leq y_n \leq \frac{e}{n+1} \quad (22)$$

udowodnionej w (3.2) ciąg $\{y_n\}$ jest wolno zbieżny, czyli wyrazy y_n i y_{n-1} są prawie sobie równe. Wtedy korzystając z własności rekurencyjnej $y_n = e - ny_{n-1}$ mamy w przybliżeniu $y_n = e - ny_n$, a stąd

$$y_n = \frac{e}{n+1} \quad (23)$$

Przekształcając wzór rekurencyjny możemy obliczyć poprzednie wyrazy ciągu $\{y_n\}$ korzystając z przybliżonej wartości pewnego dalszego wyrazu tego ciągu.

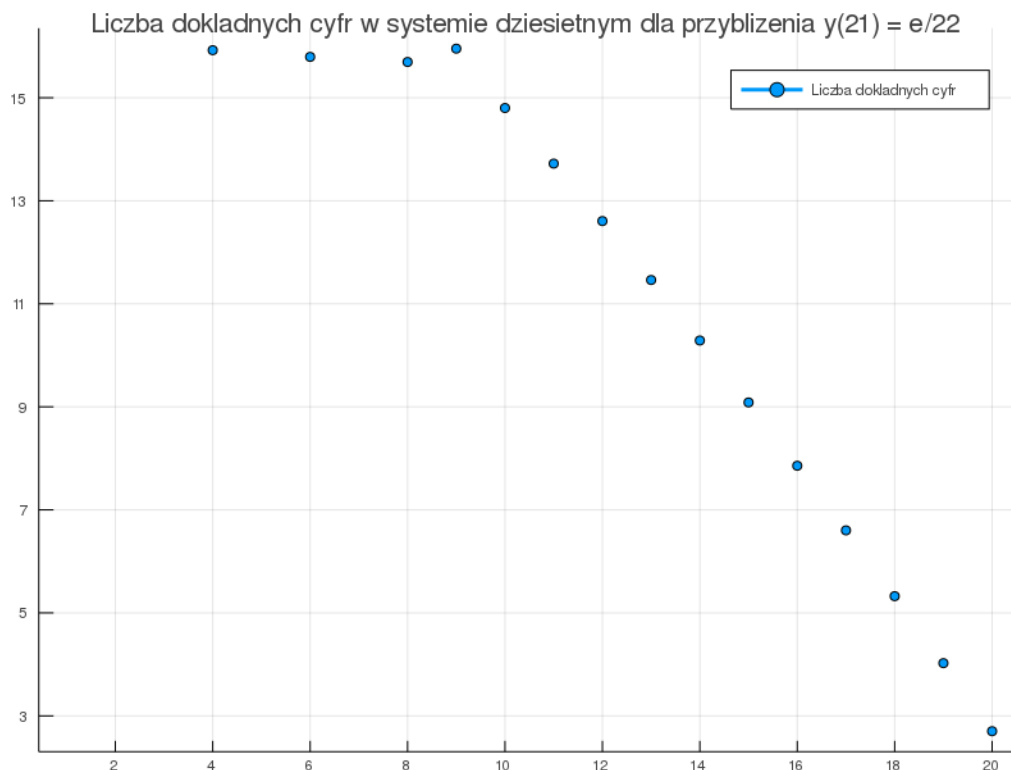
$$y_n = e - ny_{n-1} \quad (24)$$

$$y_{n-1} = \frac{e - y_n}{n} \quad (25)$$

Przedstawmy uzyskane wartości w porównaniu do wyników podanych przez Wolfram Alpha.

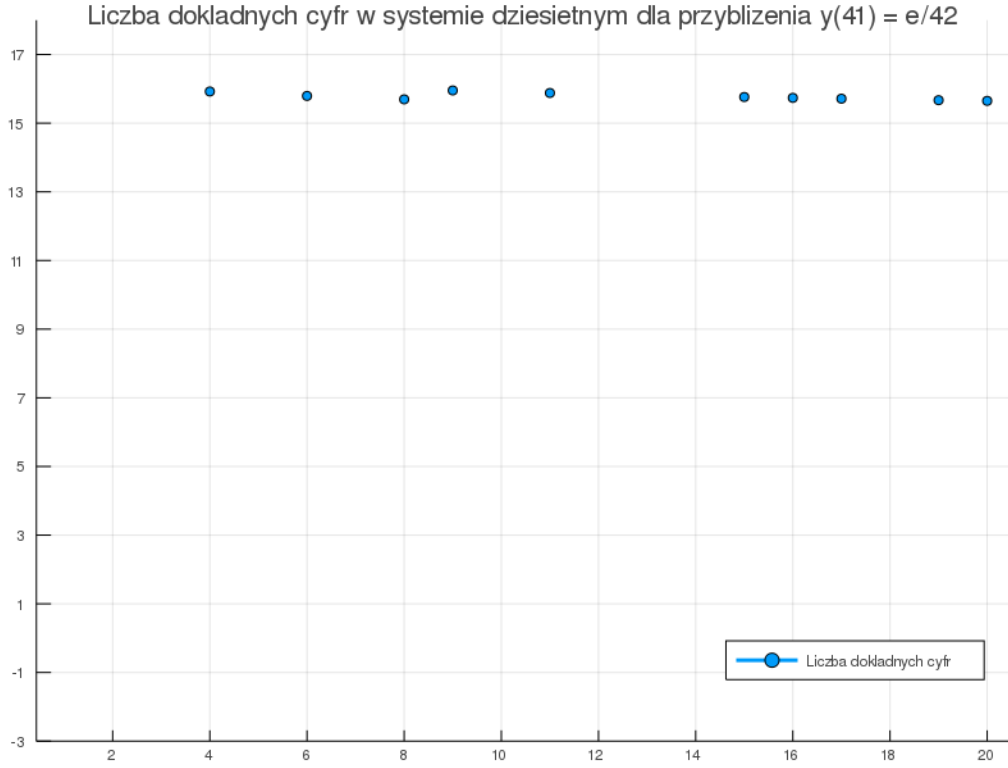
Numer wyrazu	Wyliczone przez program	Podane przez Wolfram Alpha
1	1.0	1.0
2	0.7182818284590452	0.7182818284590452
3	0.5634363430819095	0.5634363430819095
4	0.46453645613140704	0.4645364561314071
5	0.3955995478020096	0.3955995478020096
6	0.34468454164698730	0.34468454164698736
7	0.30549003693013366	0.30549003693013366
8	0.27436153301797606	0.2743615330179761
9	0.24902803129726037	0.24902803129726034
10	0.22800151548644143	0.2280015154864418
11	0.21026515810818938	0.21026515810818538
12	0.19509993116077226	0.19509993116082064
13	0.1819827233690055	0.18198272336837698
14	0.17052370129296804	0.17052370130176742
15	0.16042630906452457	0.160426308932534
16	0.1514608834266519	0.15146088553850118
17	0.14344681020596278	0.14344677430452527
18	0.1362392447517153	0.1362398909775906
19	0.12973617817645441	0.12972389988482377
20	0.12355826492995658	0.12380383076256996

Można zauważyć, że podane przybliżenia są dokładniejsze niż w przypadku wyliczania wartości kolejnych wyrazów ciągu zaczynając od zerowego elementu. Liczbę dokładnych cyfr przedstawia wykres:



Uwaga 1. Liczby całkowite, dla których na wykresie nie ma podanych wartości odpowiadają wyrazom, których wartości zostały policzone dokładnie.

Z danych przedstawionych na wykresie i w tabeli można wysunąć wniosek, że wraz z obliczaniem poprzednich wyrazów, zmniejsza się błąd otrzymywanych wartości. Prawdziwe wydaje się przypuszczenie, że wraz z wyborem dalszego wyrazu jako przybliżenia początkowego, zmniejsza się błąd wyrazów początkowych. W naszym przykładzie wyrazy do y_9 są dokładne/niemal dokładne. Spróbujmy osiągnąć większą dokładność początkowych 20 wyrazów, biorąc za początkowe przybliżenie $y_{41} = \frac{e}{42}$.



Dla początkowego przybliżenia $y_{41} = \frac{e}{42}$ otrzymaliśmy bardzo dokładne wartości (liczba cyfr dokładnych > 15) dla wszystkich pierwszych 20 wyrazów.

6 Źródła różnicy dokładności metod I i II

6.1 Niedokładność metody I

Przyjmy się najpierw metodzie I, w której obliczaliśmy kolejno $y_0, y_1, y_2, \dots, y_{20}$.

Wyraz $y_2 = e - 2$ jest liczbą niewymierną, zatem nie może być przedstawiona w sposób dokładny w arytmetyce Float64. Niech \tilde{y}_2 będzie przybliżoną wartością y_2 zapisaną w arytmetyce Float64. Wtedy $\tilde{y}_2 = y_2 \cdot (1 + \epsilon)$ dla $|\epsilon| < u$, gdzie u to precyzja arytmetyki. Załóżmy ponadto, że podstawowe działania arytmetyczne w arytmetyce Float64 wykonywane są dokładnie. Wtedy:

$$\begin{aligned}\tilde{y}_3 &= e - 3 \cdot \tilde{y}_2 = e - 3 \cdot y_2 \cdot (1 + \epsilon) = e - 3y_2 - 3\epsilon y_2 = y_3 - 3\epsilon y_2 \\ \tilde{y}_4 &= e - 4 \cdot \tilde{y}_3 = e - 4(y_3 - 3\epsilon y_2) = e - 4y_3 + 12\epsilon y_2 = y_4 + 12\epsilon y_2 \\ &\dots\end{aligned}$$

$$\tilde{y}_n = e - n \cdot \tilde{y}_{n-1} = e - n(y_{n-1} \mp \frac{1}{2}(n-1)!\epsilon \cdot y_2) = e - ny_{n-1} \pm \frac{1}{2}n!\epsilon \cdot y_2 = y_n \pm \frac{1}{2}n!\epsilon \cdot y_2$$

Mamy więc oszacowanie błędu bezwzględnego:

$$\Delta y_n = |\tilde{y}_n - y_n| = \frac{1}{2}n!\epsilon \cdot y_2 \quad (26)$$

Widać, że błąd bezwzględny rośnie niezwykle szybko wraz ze wzrostem n . Już dla $n = 10$ mamy:

$$\Delta y_{10} = \frac{1}{2} 10! \epsilon \cdot y_2 = 1.8144 \cdot 10^6 \cdot \epsilon \cdot y_2 \quad (27)$$

a dla $n = 20$:

$$\Delta y_{20} = \frac{1}{2} 20! \epsilon \cdot y_2 \simeq 1.217 \cdot 10^{18} \cdot \epsilon \cdot y_2 \quad (28)$$

Widzimy więc, że już dla małych wartości n błąd bezwzględny Δy_n staje się bardzo duży. W rzeczywistości wielkość błędu może być jeszcze większa, ze względu na niedokładność działań arytmetycznych w arytmetyce Float64 oraz przybliżoną reprezentację liczby e , których błędy również zostaną spotęgowane przez nieustanne przemnażanie przez coraz większe wartości liczby n .

6.2 Dokładność metody II

Spójrzmy teraz na metodę II, w której przybliżamy pewien początkowy wyraz y_N przez $\tilde{y}_N = \frac{e}{N+1}$, a następnie używając wzoru rekurencyjnego

$$y_{n-1} = \frac{e - y_n}{n} \quad (29)$$

obliczamy wartości poprzednich wyrazów ciągu.

W (3.2) udowodniliśmy, że

$$\frac{1}{n+1} \leq y_n \leq \frac{e}{n+1} \quad (30)$$

Zatem $\tilde{y}_N = y_N(1 + \epsilon)$, gdzie $|\epsilon| \leq e - 1$. Korzystając ze wzoru rekurencyjnego dostajemy:

$$\begin{aligned} \tilde{y}_{N-1} &= \frac{e - \tilde{y}_N}{N} = \frac{e - y_N - \epsilon y_N}{N} = \frac{e - y_N}{N} - \frac{\epsilon y_N}{N} = y_{N-1} - \frac{\epsilon \cdot y_N}{N} \\ \tilde{y}_{N-2} &= \frac{e - \tilde{y}_{N-1}}{N-1} = \frac{e - y_{N-1} + \frac{\epsilon y_N}{N}}{N-1} = \frac{e - y_{N-1}}{N-1} + \frac{\epsilon y_N}{N(N-1)} = y_{N-2} + \frac{\epsilon \cdot y_N}{N(N-1)} \\ &\dots \\ \tilde{y}_k &= \frac{e - \tilde{y}_{k+1}}{k+1} = \frac{e - y_{k+1} \mp \frac{\epsilon y_N}{N!/(k+1)!}}{k+1} = \frac{e - y_{k+1}}{k+1} \pm \frac{\epsilon y_N}{N!/k!} = y_k \pm \frac{\epsilon \cdot y_N}{N!/k!} \end{aligned}$$

Mamy więc oszacowanie błędu bezwzględnego:

$$\Delta y_k = |\tilde{y}_k - y_k| = \frac{\epsilon \cdot y_N}{N!/k!} \quad (31)$$

Widzimy więc, że im odleglejszym od wyrazu y_N jest y_k tym mniejszy będzie błąd. Zatem w celu dokładnego wyliczenia pewnej liczby początkowych elementów ciągu $\{y_n\}$ należy wybrać na tyle duże N , by wyrażenie $\frac{\epsilon \cdot y_N}{N!/k!}$ było wystarczająco małe dla każdego y_k , którego wartość chcemy uzyskać. W obliczeniach pominęliśmy błędy wynikające z niedokładności działań arytmetycznych w arytmetyce Float64 oraz niedokładność reprezentacji liczby e . Zauważmy jednak, że również wielkości tych błędów będą pomniejszane z każdym krokiem, dzięki dzieleniu przez $k+1$ podczas obliczania każdego kolejnego wyrazu y_k , podobnie jak zmniejszany jest błąd przybliżenia y_N .

7 Podsumowanie

Na podstawie wykonanych obliczeń numerycznych i analizy teoretycznej problemu doszedłem do wniosku, że w zadaniu obliczania wyrazów ciągu o zależności rekurencyjnej istotna może być kolejność wyliczanych elementów i wybór elementu początkowego. Wybór najdokładniejszego wyrazu startowego i następnie wyliczanie kolejnych wyrazów, bazując na jego wartości, może nie być stabilne. Może mieć to miejsce w sytuacji, gdy wzór ciągu powoduje stopniowe zwiększanie się nawet niewielkiego błędu początkowego przy obliczaniu kolejnych elementów. Wówczas dla obliczania dalszych wyrazów spotęgowany błąd początkowy może okazać się względnie bardzo duży. Optymalniejsza może okazać się metoda, w której jako element początkowy bierzemy nawet względnie niedokładne przybliżenie pewnego wyrazu, ale obliczanie kolejnych (bądź poprzednich) wyrazów powoduje zmniejszanie się błędu początkowego, dzięki czemu obierając odpowiednio daleki element początkowy możemy zredukować duży błąd przybliżenia początkowego do tego stopnia, że przy obliczaniu docelowego wyrazu stanie się nieznaczący.

Literatura

- [1] Maciej Paluszyński, *Analiza matematyczna dla informatyków. Notatki z wykładu*, Wrocław, 5 lutego 2019, <http://www.math.uni.wroc.pl/~mpal/academic/2019/skrypt.pdf>
- [2] Ryszard Szwarc, *Analiza matematyczna ISIM I*, Wrocław, 2013, <http://www.math.uni.wroc.pl/~szwarc/pdf/AnalizaISIM1.pdf>