Analiza Numerczna - zadanie 1.4

Eksperymentalne wyznaczanie liczby e w programie napisanym w języku Julia

Prowadzący: Paweł Woźny* autor: Dawid Więcław

14 grudnia 2018

1 Wstęp

1.1 Co to jest liczba e?

Liczbę e jest stałą wykorzystywaną w matematyce i fizyce. Jest ona podstawą logarytmu naturalnego. W przybliżeniu $e = 2,71828182845$.

1.2 Gdzie pojawia się liczba e?

Liczba e pojawia się w matematyce, gdzie jest obecna w niemal każdej jej dziedzinie. Poza matematyką pojawia się też w:

1. Ekonomii - niech V_0 będzie kapitałem początkowym, V kapitałem aktualnym, m liczbą kapitalizacji w roku, r stopą oprocentowania, a n liczbą lat (okresów). W przypadku kapitalizacji ciągłej kapitał przyszły wyraża się wzorem $\lim_{m\to\infty} V_0(1+\frac{r}{m})^{mn} = V_0\mathbf{e}^{rn}$
2. Biologii - liczebność populacji w warunkach idealnych: $f(t) = m_0 a^{\frac{t}{T}}$, gdzie m_0 to początkowa liczba organizmów, a to liczba organizmów powstających z danego organizmu, T to czas jednego cyklu, a t to czas. Chcąc zbadać aktualne tempo wzrostu populacji należy policzyć pochodną $f'(t) = m_0(\frac{1}{T})a^{t/T}log_{\mathbf{e}}a$, gdzie pojawia się e.
3. Fizyce - Przy rozpadzie radioaktywnym występuje funkcja wykładnicza, w rucu falowym e stanowi podstawę funkcji wykładniczej będącej czynnikiem tłumiącym funkcji wychylanie od czasu: $x(t) = \mathbf{e}^{-\frac{\Gamma t}{2}}(Asin(\omega t) + Bsin(\omega t))$

1.3 Kilka wzorów

1. $\int e^x = e^x + C$
2. $(a^x)' = a^x log_e(a)$
3. $e^{i*\pi} = 1$

1.4 Po co ten eksperyment?

Eksperyment ma na celu zbadanie metod numerycznych (zaczynając od $e \approx (1+\frac{1}{n})^n$) mających na celu uzyskać liczbę e jako daną w pamięci komputera. Badanie również ma na celu porównanie ze sobą tych sposobów, a także znalezienie takiego który byłby równie skuteczny jak ten, którego użyli autorzy stałej e w języku Julia.

* *E-mail*: Pawel.Wozny@ii.uni.wroc.pl

2 Próba z definicji

2.1 Opis

Pierwsze podejście będzie polegało na próbie wyliczenia liczby e korzystając z jej definicji jako granicy $e = lim(1 + 1/n)^n$, czyli jak miało to miejsce w treści zadania. Wykorzystując załączony program o nazwie **program.ipynb** zweryfikowano kolejne wartości funkcji $zGranicy(n) = (1 + \frac{1}{n})^n$, co wydaje się najlepszym numerycznym odwzorowaniem dla granicy:

$$\lim_{n\to\infty}(1+\frac{1}{n})^n$$

dla odpowiednio dużych n.

2.2 Tabela wyników dla precyzji 16 i 32 bitowej

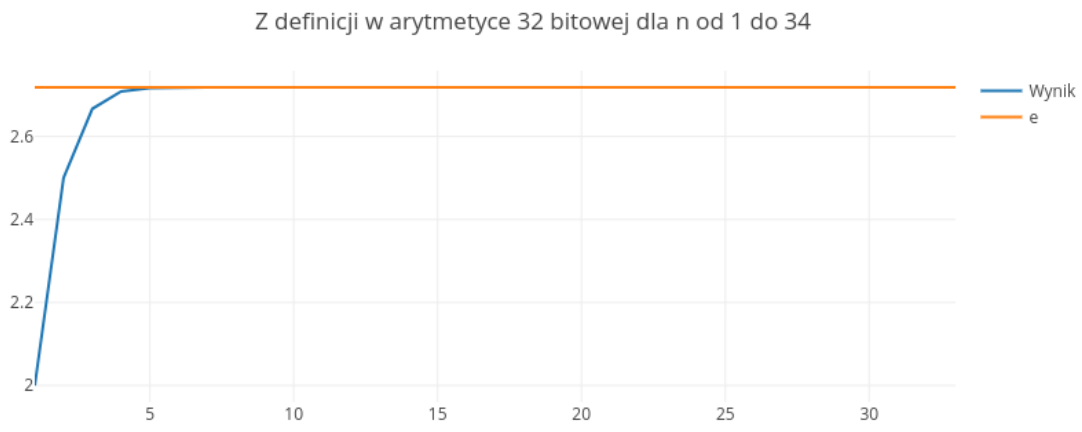
	W precyzji 16 bitowej			W precyzji 32 bitowej		
N	Wynik	bł. bezwzgl.	bł. wzg.	Wynik	bł. bezwzgl.	bł. wzg.
8	**2**.56578	1.5e-01	5.6e-02	**2**.5657845139503479	1.5e-01	5.6e-02
8^2	**2**.69734	2.1e-02	7.7e-03	**2**.6973449525650989	2.1e-02	7.7e-03
8^3	**2.71**563	2.6e-03	9.7e-04	**2.71**56320001689912	2.6e-03	9.7e-04
8^4	**2.71**795	3.1e-04	1.1e-04	**2.71**79500811896659	3.3e-04	1.2e-04
8^5	**2.7182**6	0.0e+00	0.0e+00	**2.7182**403519302940	4.1e-05	1.5e-05
8^6	1.00000	1.7e+00	6.3e-01	**2.7182**766437660460	5.2e-06	1.9e-06
...	...	...	...	...	...	...
8^{10}	1.00000	1.7e+00	6.3e-01	**2.71828182**69357085	1.9e-09	6.9e-10
8^{11}	1.00000	1.7e+00	6.3e-01	1.0000000000000000	1.7e+00	6.3e-01

2.3 Tabela wyników dla precyzji 256 bitowej

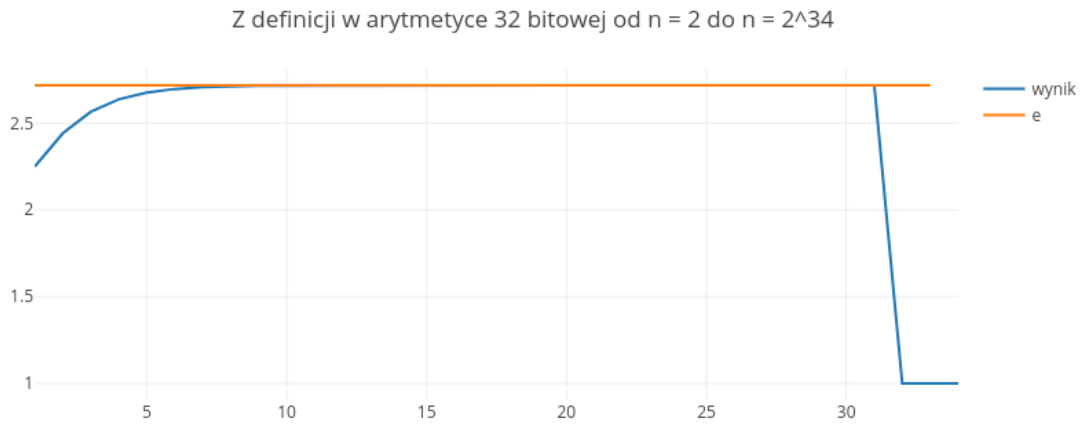
Wyniki dla $e \approx (1 + 1/n)^n$ w precyzji 256 bitowej			
N	Wynik	Błąd bezwzględny	Błąd względny
8	**2**.56578451	1.5e-01	5.6e-02
64	**2**.69734495	2.1e-02	7.7e-03
512	**2.71**563200	2.6e-03	9.7e-04
...	...	...	...
8^{21}	**2.7182818285**	1.5e-19	5.4e-20
8^{22}	1.0000000000	1.7e+00	6.3e-01

Aby porównać wynik eksperymentu z wynikiem dokładnym wykorzystano stałą e zapisaną w bibliotece języka Julia (zakładając, że jest to poprawne przybliżenie). Przy pomocy tej stałej obliczono błędy bezwzględne i względne oraz zaznaczono dokładne cyfry dziesiętne poprzez pogrubienie ich.

2.4 Wykresy



Rysunek 1: Wykres zależności wyniku od n a także porónanie go z wynikiem dokładnym



Rysunek 2: Wykres zależności wyniku od n dla zakresu ukazującego błąd od odpowienio dużych argumentów

2.5 Wnioski

Z wyników tabeli można zauważyć, że różnica maleje dziesięciokrotnie wraz z pomnożeniem liczby n przez 8. Jednakże dla odpowiednio dużych n składnik $\frac{1}{n}$ staje się równy zero przez co $(1+\frac{1}{n})^n = (1+0)^n = 1^n = 1$, co sprawia że metoda ta wymaga bardzo dużo pamięci dla uzyskiwania kolejnych, dokładniejszych przybliżeń, co sprawia, że jest ona nieefektywna.

3 Inne podejście

3.1 Opis

Rozpatrując funkcję $f(x) = e^x$ w $x = 1$ otrzymuje się liczbę e. Aby móc analizować tą funkcję należy doprowadzić ją do postaci, którą da się zaimplementować na komputerze. Zatem rozwijając e^x w szereg Macloraine'a (a mianowicie jego skończony fragment) w $x = 1$ możliwe jest otrzymanie odpowieniego przybliżenia wartości liczby e. Więc $e = \sum_{k=0}^{\infty} \frac{1}{k!}$. Jego skończony fragment zaś należałoby przedstawić w postaci $e \approx \sum_{k=0}^{n} \frac{1}{k!}$, odpowiednio zwiększając n możliwe jest zatem uzyskanie odpowiednio dokładnego wyniku.

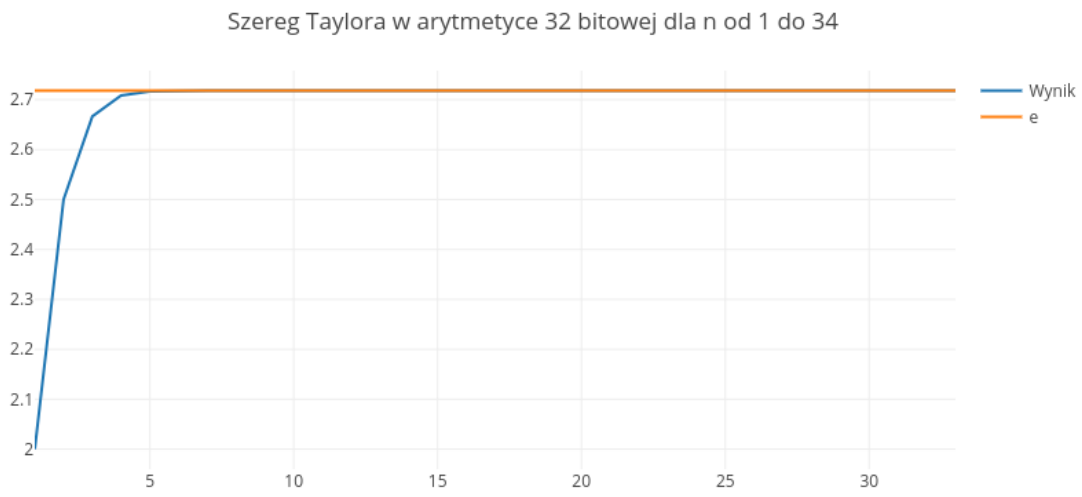
3.2 Tabela wyników dla precyzji 16 i 32 bitowej

	W precyzji 16 bitowej			W precyzji 32 bitowej		
N	Wynik	bł. bezwzgl.	bł. wzg.	Wynik	bł. bezwzgl.	bł. wzg.
1	**2**.00000	7.2e-01	2.6e-01	**2**.000000000	7.2e-01	2.6e-01
2	**2**.50000	2.2e-01	8.0e-02	**2**.500000000	2.2e-01	8.0e-02
3	**2**.66668	5.2e-02	1.9e-02	**2**.666687011	5.2e-02	1.9e-02
4	**2.7**0837	9.9e-03	3.6e-03	**2.7**08374023	9.9e-03	3.6e-03
5	**2.71**667	1.6e-03	5.8e-04	**2.71**6666666	1.6e-03	5.9e-04
6	**2.718**01	2.4e-04	9.0e-05	**2.718**055555	2.3e-04	8.3e-05
7	**2.7182**6	0.0e+00	0.0e+00	**2.7182**53968	2.8e-05	1.0e-05
...	...	...	...	...	...	...
11	**2.7182**6	0.0e+00	0.0e+00	**2.718281826**	2.8e-09	1.0e-09
12	**2.7182**6	0.0e+00	0.0e+00	**2.718281827**	9.3e-10	3.4e-10
...	...	...	...	...	...	...
65	**2.7182**6	0.0e+00	0.0e+00	**2.718281827**	9.3e-10	3.4e-10
66	Inf	-Inf	-Inf	Inf	-Inf	-Inf

3.3 Tabela wyników dla arytemtyki 256 bitowej

$e \approx \sum_{k=0}^{n} \frac{1}{k!}$ w precyzji 256 bitowej			
N	Wynik	Błąd bezwzględny	Błąd względny
1	**2**.0000000000000000	7.2e-01	2.6e-01
2	**2**.5000000000000000	2.2e-01	8.0e-02
3	**2**.6666870117187500	5.2e-02	1.9e-02
4	**2.7**083740234375000	9.9e-03	3.6e-03
...	...	...	...
20	**2.718281828**4590452	2.1e-20	7.5e-21
21	**2.718281828**4590452	2.6e-19	9.4e-20
...	...	...	...
65	**2.718281828**4590452	1.3e-17	4.8e-18
66	Inf	-Inf	-Inf

3.4 Wykres



Rysunek 3: Wykres zależności wyniku przybliżenia $e \approx \sum_{k=0}^{n} \frac{1}{k!}$ od n i zestawienie go z dokładną wartością

3.5 Wnioski

Metoda ta (jak można wywnioskować z tabel) jest znacznie szybsza od uzyskiwania liczby e metodą pierwszą a jej błąd dla zadanego n można oszacować z góry przez $(\frac{1}{n!})$ co zostanie udowodnione.

Twierdzenie 1. *Błąd w n-tej sumie szeregu* $e \approx \sum_{k=0}^{n} \frac{1}{k!}$ *wynosi co najwyżej* $\frac{1}{n!}$

Dowód. N-ta reszta szeregu Macloraine'a w postaci Cauchy'ego wynosi $R_n(1,0) = \frac{e^{\theta}(1-\theta)^n}{n!}$ $\theta \in [0,1]$ badając funkcję $f(\theta) = e^{\theta}(1-\theta)^n$na przedziale do którego należy θ otrzumujemy: $f'(\theta) = e^{\theta}(1-\theta)^{n-1}(1-\theta-n)$ co jest mniejsze od 0 dla $\theta \in (0,1]$, a zatem największa wartość ta funkcja osiagnie dla $\theta = 0$, więc wyniesie ona 1. Czyli różnica w najgorszym przypadku wynosi $R = \frac{1}{n!}$, skąd jest to oszacowanie błędu z góry. □

Jest to najszybsza zbieżność jaką udało się uzyskać w tych rozważaniach, jednak pojawia się problem natury numerycznej tak samo jak w przypadku pierwszym, gdyż dla odpowiednio dużych n liczba $\frac{1}{n!}$ może być równa 0, a także poprzez przekroczenie zakresu, prawdopodobnie $n! = 0$ co daje wynik $e \approx \infty$. Aby zniwelować ten błąd w następnej części następuje poprawa (numeryczna) tego sposobu.

4 Próba ulepszenia

4.1 Opis

Tempo zbieżności metody nr 2 można uznać za zadowalające, gdyż chcąc błąd rzędu 10^{-n} dla odpowiednio dużych n wystarczy mniej niż n iteracji algorytmu, więc należałoby jedynie uporać się z problemem natury numerycznej dzielenia $\frac{1}{n!}$ co skutkuje:
$e = \sum_{n=0}^{\infty} \frac{1}{n!} = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{4!} + \frac{1}{4!} + \frac{1}{5!} ... = 2 + \frac{1}{2}(1 + \frac{1}{3} + \frac{1}{3*4} + \frac{1}{3*4*5!} + ...) = 2 + \frac{1}{2}(1 + \frac{1}{3}(1 + \frac{1}{4} + \frac{1}{4*5!} + ...)) = 2 + \frac{1}{2}(1 + \frac{1}{3}(1 + \frac{1}{4}(1 + \frac{1}{5!}(1 + ...))))$

4.2 Tabela wyników w precyzji 16 bitowej

Wyniki dla $e = 2 + \frac{1}{2}(1 + \frac{1}{3}(1 + \frac{1}{4}(1 + \frac{1}{5!}(1 + ...))))$			
N	Wynik	Błąd bezwzględny	Błąd względny
2	**2**.5000000000000000	2.2e-01	8.0e-02
4	**2.7**083129882812500	9.9e-03	3.7e-03
7	**2.7182**617187500000	0.0e+00	0.0e+00

4.3 Tabela wyników w precyzji 32 bitowej

Wyniki dla $e = 2 + \frac{1}{2}(1 + \frac{1}{3}(1 + \frac{1}{4}(1 + \frac{1}{5!}(1 + ...))))$			
N	Wynik	Błąd bezwzględny	Błąd względny
2	**2**.5000000000000000	2.2e-01	8.0e-02
4	**2.7**083333330228925	9.9e-03	3.7e-03
8	**2.7182**787694036961	3.1e-06	1.1e-06
13	**2.718281828**7983537	0.0e+00	0.0e+00

4.4 Tabela wyników w precyzji 256 bitowej

Wyniki dla $e = 2 + \frac{1}{2}(1 + \frac{1}{3}(1 + \frac{1}{4}(1 + \frac{1}{5!}(1 + ...))))$			
N	Wynik	Błąd bezwzględny	Błąd względny
2	**2**.5000000000000000	2.2e-01	8.0e-02
4	**2.7**083333333333333	9.9e-03	3.7e-03
8	**2.7182**787698412698	3.1e-06	1.1e-06
16	**2.718281828**4590423	3.0e-15	1.1e-15
32	**2.718281828**4590452	1.2e-37	4.4e-38
57	**2.718281828**4590452	0.0e+00	0.0e+00

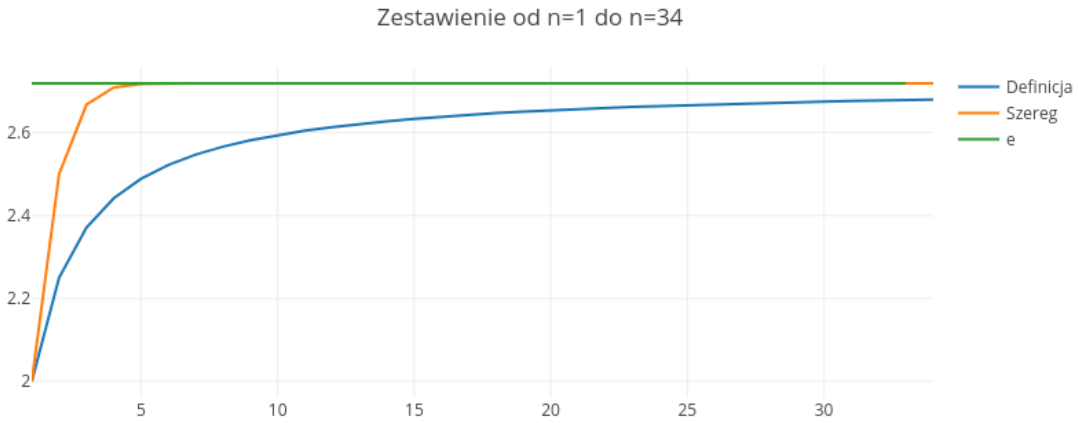
4.5 Wnioski

Tempo zbieżności tej metody jest takie samo jak w przypadku metody nr 2, jednak przez zamiane sumy na ten schemat udało sie wyeliminować dzielenia przez duże liczby (największy mianownik jaki się pojawi w tym schemacie to n) dzięki czemu algorytm przebiega poprzez ciągłę dodawanie 1 i czegoś z przedziału $[\frac{1}{n}, 1.5)$, co wymaga znacznie mniej pamięci niż liczenie $\frac{1}{n!}$.

5 Podsumowanie

Podsumuwując udało się osiągnąć zerowy błąd w precyzji 16, 32 i 256 bitowej oraz błąd wyrażający się od iteracji algorytmu z rozdzialu 4 (i 3) oszacowany z góry przez $\frac{1}{n!}$.

Zestawienie błedów bezwzględnych w precyzji 256 bitowej			
N	$e = (1 + \frac{1}{n})^n$	$e = \sum_{n=0}^{\infty} \frac{1}{n!}$	$e = 2 + \frac{1}{2}(1 + \frac{1}{3}(1 + \frac{1}{4}(...)))$
2	4.7e-01	2.2e-01	2.2e-01
4	2.8e-01	9.9e-03	9.9e-03
8	1.5e-01	3.1e-06	3.1e-06
16	8.0e-02	3.0e-15	3.0e-15
8^5	4.1e-05	Inf	0.0e+00
8^{10}	1.3e-09	Inf	0.0e+00



Rysunek 4: Zestawienie wyników metod poruszanych w tym sprawozdaniu

Porównując wyniki uzyskane w każdej z metod można dojść do wniosku, że metoda poruszona w treści zadania ($e \approx (1 + \frac{1}{n})^n$) jest skrajnie nieefektywnym sposobem uzyskiwania kolejnych przybliżeń liczby e i w rozważanych przypadkach nigdy błąd nie osiągnął zera dla danej precyzji, a także jest zdecydowanie wolniejsza (zmniejsznie błędu 10-cio krotnie na każde przemnożenie n przez 8) niż sposób wyliczenia jej z przekształconego szeregu Macloraine'a ($log_{10}(e - przyblienie) < -n$ dla odpowiednio dużych n. Wyliczenie przybliżenia liczby e przy pomocy ostatniej metody pozwala także na uzyskanie zerowego błędu w każdej rozważanej precyzji.

Literatura

[1] B. Miś, Tajemnicza liczba e i inne sekrety matematyki, Wydawnictwo Naukowo-Techniczne, Warszawa 2008

[2] K. Kuratowski, Rachunek różniczkowy i całkowy, Wydawnictwo naukowe PWN, Warszawa 2018.