Metody obliczeniowe w nauce i technice

Adam Naumiec
Marzec 2023

Laboratorium 2

Arytmetyka komputerowa (cd.)

Spis treści

1. Treść zadań

1. Napisać algorytm do obliczenia funkcji wykładniczej e^x przy pomocy nieskończonych szeregów:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$

   1.1. (1a) Wykonując sumowanie w naturalnej kolejności, jakie kryterium zakończenia obliczeń przyjmiesz?
   1.2. (1b) Proszę przetestować algorytm dla:
   - $x = \pm 1$,
   - $x = \pm 5$,
   - $x = \pm 10$,

   i porównać wyniki z wykonania standardowej funkcji $exp(x)$.
   1.3. (1c) Czy można posłużyć się szeregami w tej postaci do uzyskania dokładnych wyników dla $x < 0$?
   1.4. (1d) Czy możesz zmienić wygląd szeregu lub w jakiś sposób przegrupować składowe, żeby uzyskać dokładniejsze wyniki dla $x < 0$?

2. Dwa matematycznie ekwiwalentne wyrażenia:

$$x^2 - y^2 \text{ oraz } (x - y)(x + y)$$

   2.1. Które z nich może być obliczone dokładnie w arytmetyce zmiennoprzecinkowej? Dlaczego?
   2.2. Dla jakich wartości x i y, względem siebie, istnieje wyraźna różnica w dokładności dwóch wyrażeń?

3. Zakładamy, że rozwiązujemy równanie kwadratowe:

$$ax^2 + bx + c = 0,$$

   gdzie:
   - $a = 1{,}22$;
   - $b = 3{,}34$;
   - $c = 2{,}28$;

   wykorzystując znormalizowany system zmiennoprzecinkowy z:
   - podstawą $\beta = 10$,
   - dokładnością $p = 3$.

   3.1. (a) Ile wyniesie obliczona wartość $\Delta = b^2 - 4ac$?
   3.2. (b) Jaka jest dokładna wartość wyróżnika Δ w rzeczywistej (dokładnej) arytmetyce?
   3.3. (c) Jaki jest względny błąd w obliczonej wartości wyróżnika?

2. Rozwiązania

2.1. Zadanie 1

2.1.1. Algorytm

Algorytm obliczania funkcji wykładniczej za pomocą nieskończonego szeregu napisano w języku Python w wersji 3.11.

```
import math



def e_to_x_power_1(x=1, epsilon=0.000001):
    e_to_x_power = 0
    i = 0

    while True:
        e_to_x_power += (x ** i) / math.factorial(i)
        if abs(x ** (i + 1) / math.factorial(i + 1)) < epsilon:
            break
        i += 1

    return e_to_x_power


def e_to_x_power_2(x=1, n=100):
    e_to_x_power = 0

    for i in range(n):
        e_to_x_power += (x ** i) / math.factorial(i)

    return e_to_x_power


if __name__ == '__main__':
    powers = [1, -1, 5, -5, 10, -10]
    values = [2.71828182845904523536,
              0.36787944117144232159,
              148.41315910257660342111,
              0.00673794699908546709,
              22026.46579480671651695790,
              0.00004539992976248485]

    for power, value in zip(powers, values):
        print("Przybliżona wartość e do potęgi {} wynosi: {}".format(power,
              value))

        e1 = e_to_x_power_1(power)
        print("WERSJA 1. (epsilon) Wyliczona wartość e do potęgi {} wynosi:
              {}".format(power, e1))
        print("WERSJA 1. (epsilon) Błąd względny 1: {}".format(abs(value -
              e1) / value))

        e2 = e_to_x_power_2(power)
        print("WERSJA 2. (n iteracji) Wyliczona wartość e do potęgi {}
              w ynosi: {}".format(power, e2))
        print("WERSJA 2. (n iteracji) Błąd względny 1:
              {}\n".format(abs(value - e2) / value))
```

2.1.2. Zadanie (1a) – kryterium zakończenia obliczeń

Można przyjąć wiele kryteriów zakończenia obliczeń, m.in.:

- Obliczanie ustalonej (np. wybranej przez użytkownika lub proporcjonalnej do wielkości wykładnika) liczby wyrazów szeregu.
- Porównanie wartości bezwzględnej kolejnych składników szeregu z pewną ustaloną mała wartością epsilon i zakończenie algorytmu, gdy kolejne iterowane wartości są na moduł mniejsze od pewnego niewielkiego epsilon (które może zostać wybrane przez użytkownika lub którego wielkość może zostać przyjęta proporcjonalnie do wielkości wykładnika).

Dokonano obliczeń dla obu wersji, a parametru zostały ustalone na początku na konkretne wartości:

- w wersji z ustaloną liczbą obliczanych wyrazów przyjęto:
$$n = 100;$$
- w wersji z epsilonem przyjęto:
$$\varepsilon = 0{,}000001.$$

2.1.3. Zadanie (1b) – test algorytmu

Wartości wyrażenia e^x dla różnych argumentów uzyskane za pomocą algorytmu oraz przybliżone wartości dokładne wraz z błędem względnym (jako separator dziesiętny wykorzystano kropkę zgodnie ze specyfikacją języka Python):

- wersja pierwsza algorytmu (n iteracji):

x	Przybliżona wartość dokładna	Wartość obliczona	Błąd względny
1	2.7182818284590452 3536	2.7182818284590455	1.6337129034990842e-16
-1	0.36787944117144232159	0.36787944117144245	3.017899073375402e-16
5	148.41315910257660342111	148.41315910257657	1.915039717654698e-16
-5	0.00673794699908546709	0.006737946999086907	2.136883067284701e-13
10	22026.46579480671651695790	22026.46579480671	3.3032796463874436e-16
-10	0.0000453999297624 8485	4.5399929433607724e-05	7.244000706807642e-09

Tabela 1. Wartości funkcji exp(x) dla wersji z wybraną liczbą iteracji

- wersja druga algorytmu (epsilon):

x	Przybliżona wartość dokładna	Wartość obliczona	Błąd względny
1	2.7182818284590452 3536	2.7182815255731922	1,1142547828265698e-07
-1	0.36787944117144232159	0.3678791887125221	6,862544952488235e-07
5	148.41315910257660342111	148.41315852164774	3.914267836865017e-09
-5	0.00673794699908546709	0.006738328152479823	5.656817935905362e-05
10	22026.46579480671651695790	22026.465793823776	4.462549121888885e-11
-10	0.0000453999297624 8485	4.5974459989140606e-05	0.012654870385515623

Tabela 2. Wartości funkcji exp(x) dla wersji z epsilon

Wyniki uzyskane w obu wersjach są dosyć satysfakcjonujące, obliczone wartości nie odbiegają bardzo znacząco od wartości rzeczywistych, ale wraz z wzrostem wykładnika (oraz gdy wykładnik malał do bardzo małych wartości) różnica zaczyna się coraz bardziej uwidaczniać.

Efektywność obu algorytmów zależy od dostępnej mocy obliczeniowej i związanymi z tym możliwymi parametrami wykonania algorytmu.

Błędy obliczeniowe są szczególnie zauważalne dla wartości ujemnych. Błąd względny w obu wersjach algorytmu wraz z maleniem ujemnego wykładnika szybko wzrastał co widać szczególnie w drugiej wersji algorytmu (epsilon) z przyjętymi parametrami.

2.1.4. Zadanie (1c) – wyniki dla x < 0

Ten szereg można zastosować do obliczenia wartości funkcji dla $x < 0$, ale dla bardzo małych wartości x obliczenie wartości szeregu może być problematyczne z powodu ograniczonej precyzji arytmetyki bardzo małych liczb w systemach zmiennopozycyjnych. Szereg ten jest wolno zbieżny, co oznacza, że potrzeba wielu wyrazów, aby uzyskać dokładną wartość, szczególnie dla wartości x bliskich zeru. Dlatego dla $x < 0$ lepszym rozwiązaniem jest użycie innych szeregów lub metod numerycznych.

2.1.5. Zadanie (1d) – dokładniejsze wyniki dla x < 0

Szereg:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots,$$

można przekształcić do postaci:

$$e^x = 1 + \left(\frac{x}{1}\right) \cdot \left(1 + \left(\frac{x}{2}\right) \cdot \left(1 + \left(\frac{x}{3}\right) \cdot (1 + \cdots)\right)\right),$$

ta postać szeregu jest bardziej stabilna numerycznie niż poprzednia, szczególnie dla $x < 0$. Można ją zaimplementować rekurencyjnie lub iteracyjnie. Przekształcenie szeregu do szeregu ilorazowego pozwala na efektywne obliczenie wartości szeregu za pomocą algorytmu Hornera z mniejszą liczbą działań arytmetycznych, co pozwala na zminimalizowanie wpływu błędów numerycznych.

2.1.6. Wnioski

Dokładność obliczeń zależy od wielu składowych m.in. algorytmu, kryterium zakończenia obliczeń, postaci wyrażenia matematycznego itp. Przy wykonywaniu obliczeń na komputerze należy zadbać o marginalizację błędów i przyjęcie odpowiednich kryteriów przy implementacji algorytmów. Należy także zwrócić uwagę dla jakich liczb można za pomocą danego algorytmu otrzymać dokładniejsze

wyniki, a dla jakich otrzymane rezultaty mogą być mniej satysfakcjonujące. Warto się także zastanowić, czy zmiana postaci wyrażenia matematycznego nie pozwoli na uzyskanie lepszych wyników.

2.2. Zadanie 2

2.2.1. Zadanie 2.1. Wartość wyrażeń matematycznie ekwiwalentnych w arytmetyce zmiennoprzecinkowej

Matematycznie wyrażenia są ekwiwalentne, ale w arytmetyce zmiennoprzecinkowej dokładniej obliczona będzie wartość wyrażenia:

$$(x - y)(x + y).$$

Dzieje się tak, ponieważ w arytmetyce zmiennoprzecinkowej, obliczenia są przeprowadzane z pewną skończoną precyzją. W przypadku obliczania $x^2 - y^2$ wynik może być zaburzony z powodu błędów zaokrągleń, które występują podczas operacji arytmetycznych na liczbach zmiennoprzecinkowych, w tym przypadku szczególnie potęgowania, te wyrażenie obarczone jest większym błędem numerycznym. Natomiast, w przypadku obliczania $(x - y)(x + y)$, wartość jest obliczana przez wykonanie jedynie trzech działań arytmetycznych: dodawania i odejmowania, które są bardziej stabilne numerycznie i mniej podatne na błędy zaokrągleń, a później mnożenia, dzięki czemu możliwe jest uzyskanie dokładniejszego wyniku.

2.2.2. Zadanie 2.2. Wartości niewiadomych, dla których istnieje wyraźna różnica w dokładności obliczeń

Istnieje wyraźna różnica w dokładności dwóch wyrażeń, gdy wartości x i y są bliskie sobie. W szczególności, jeśli wartości x i y są równe, to różnica między wynikami obliczeń dwóch wyrażeń może być znaczna. Dzieje się tak, ponieważ znaczące cyfry wyniku ulegają redukcji, podczas gdy błąd może pozostać niezmieniony. Wyrażenie $x^2 - y^2$ może również powodować duże błędy względne.

2.2.3. Wnioski

W przypadku obliczeń numerycznych należy zawsze brać pod uwagę potencjalny wpływ błędów numerycznych na wynik. Należy również rozważyć różne sposoby obliczania danego wyrażenia i wybrać ten, który minimalizuje błędy numeryczne.

W przypadku wyrażeń algebraicznych, takich jak $x^2 - y^2$ i $(x - y)(x + y)$, istnieją różne techniki i identyczności algebraiczne, które pozwalają na zmniejszenie błędów numerycznych i zwiększenie dokładności obliczeń. Przykładem takiej identyczności jest właśnie $(x - y)(x + y) = x^2 - y^2$, która pozwala na obliczenie jednego wyrażenia na podstawie drugiego, minimalizując tym samym błędy numeryczne.

2.3. Zadanie 3

2.3.1. Równanie kwadratowe i znormalizowany system zmiennoprzecinkowy

Rozważane równanie kwadratowe ma postać:

$$1{,}22x^2 + 3{,}34x + 2{,}28 = 0.$$

Wzór na wyróżnik tego równania (trójmianu kwadratowego) to:

$$\Delta = b^2 - 4 \cdot a \cdot c.$$

Wartość wyróżnika w arytmetyce zmiennoprzecinkowej oznaczymy jako: $\hat{\Delta}$, natomiast wartość wyróżnika w arytmetyce dokładnej jako: Δ.

2.3.2. Zadanie 3.1. (a) Obliczona wartość wyróżnika w arytmetyce zmiennoprzecinkowej

Przyjętym w zadaniu systemem zmiennoprzecinkowy jest system:

o podstawie $\beta = 10$,
z dokładnością $p = 3$.

Wartość wyróżnika w arytmetyce zmiennoprzecinkowej w tym systemie równa jest:

- $fl(b^2) = fl(3{,}34 \cdot 3{,}34) = fl(11{,}1556) = 11{,}2$,
- $fl(4 \cdot a \cdot c) = fl(fl(4 \cdot a) \cdot c) = fl(fl(4 \cdot 1{,}22) \cdot 2{,}28) = fl(fl(4{,}88) \cdot 2{,}28) = fl(4{,}88 \cdot 2{,}28) = fl(11{,}1264) = 11{,}1$;

$$\hat{\Delta} = fl(fl(b^2) - fl(4 \cdot a \cdot c)) = fl(11{,}2 - 11{,}1) = fl(0{,}1) = 0{,}1.$$

2.3.3. Zadanie 3.2. (b) Obliczona dokładna wartość wyróżnika w dokładnej arytmetyce

Dokładna wartość wyróżnika wynosi:

$$\Delta = 3{,}34^2 - 4 \cdot 1{,}22 \cdot 2{,}28 = 11{,}1556 - 11{,}1264 = 0{,}0292.$$

2.3.4. Zadanie 3.3. (c) Względny błąd w obliczonej wartości wyróżnika

Błąd względny wyraża się wzorem:

$$\delta = \frac{\Delta x}{x} = \frac{|x - \hat{x}|}{x},$$

w tym przypadku wzór ma postać:

$$\delta = \frac{|\Delta - \hat{\Delta}|}{\Delta}.$$

Otrzymujemy, że błąd względny równy jest:

$$\delta = \frac{|0{,}0292 - 0{,}1|}{0{,}0292} = \frac{|-0{,}0708|}{0{,}0292} = \frac{0{,}0702}{0{,}0292} \approx 2{,}424657 \approx 2{,}42.$$

2.3.5. Wnioski

Mimo niewielkich współczynników równania, otrzymane wyniki znacząco różniły się od siebie. Problem, który był „łatwy do rozwiązania na kartce" okazał się dawać bardzo niedokładne wyniki w systemie zmiennoprzecinkowym przez przyjęcie systemu o zbyt małej precyzji. Przykład dobitnie pokazuje, że ustalając precyzję systemu należy mieć na uwadze nie tylko dane wejściowe, a również wartości uzyskiwane podczas dokonywania obliczeń.

3. Bibliografia

1. Wykłady dr inż. Katarzyny Rycerz z przedmiotu *Metody obliczeniowe w nauce i technice* na czwartym semestrze kierunku Informatyka w AGH w Krakowie
2. Wykresy kreślono za pomocą internetowego programu GeoGebra: https://www.geogebra.org/calculator
3. Obliczenia wykonywano za pomocą internetowego programu WolframAlpha: https://www.wolframalpha.com/ oraz programu Microsoft Excel: https://www.microsoft.com/pl-pl/microsoft-365/excel
4. Programy napisane zostały w języku Python w wersji 3.11: https://www.python.org/
5. Wykorzystano bibliotekę NumPy dla języka Python w wersji 1.24: https://numpy.org/doc/stable/index.html
6. https://en.wikipedia.org/wiki/IEEE_754