Obliczenia naukowe

Sprawozdanie

Lista 1

Monika Tworek

Indeks: 229776

Zadanie 1

Wyznaczyć iteracyjnie macheps, eta i MAX, a następnie porównać z funkcjami bibliotecznymi.

	macheps	eps()	eta	nextFloat()
Float16	0.000977	0.000977	6.0e-8	6.0e-8
Float32	1.1920929e-7	1.1920929e-7	1.0e-45	1.0e-45
Float64	2.220446049250313e-16	2.220446049250313e-16	6.0e-8	5.0e-324

	MAX	realmax()
Float16	6.55e4	6.55e4
Float32	3.4028235e38	3.4028235e38
Float64	1.7976931348623157e308	1.7976931348623157e308

Precyzja arytmetyki jest dwukrotnie mniejsza niż macheps, natomiast wartości eta i MIN_{sub} są takie same.

Zadanie 2

Sprawdzenie eksperymentalne słuszności twierdzenia Kahana.

Po wpisaniu w program otrzymujemy następujące wyniki:

Float16: -0.000977
Float32: 1.1920929e-7
Float64: -2.220446049250313e-16

Stwierdzenie Kahana jest słuszne tylko w przypadku Float32. Dla typu Float16 i Float64 różni się znakiem.

Zadanie 3

Sprawdzenie eksperymentalne w arytmetyce Float64, że liczby są równomiernie rozłożone w przedziale [1,2]. Sprawdzenie zależności dla przedziału[$\frac{1}{2}$,1] i [2,4].

Liczby w przedziale [1,2] są równomiernie rozmieszczone, bo gdy do liczby 1, która w reprezentacji Float64 jest przedstawiana jako:
0011111111110000000000000000000000000000000000000000000000000000
dodamy 2^{-52} otrzymujemy:
0011111111110000000000000000000000000000000000000000000000000001

Z tego można wywnioskować, że liczbę zmiennopozycyjną x z przedziału [1,2] można przedstawić za pomocą $x = 1 + k\delta$ w tej arytmetyce, gdzie $k = 1, 2, \ldots, 2^{52} - 1$ i $\delta = 2^{-52}$.

Analogicznie przedstawia się sytuacja dla przedziału [$\frac{1}{2}$,1], jednakże 2 podnosimy do potęgi 53, a dla przedziału [2,4] do potęgi 51.

Wynika z tego, że im większa odległość przedziału [2^m, 2^{m+1}] rozważanego, tym jest mniej liczb w przedziale, ale są one rozłożone co stałą odległość – epsilon maszynowe.

Zadanie4

Znaleźć w arytmetyce Float64 największą i najmniejszą liczbę x z przedziału [1,2] taką, że $x * \frac{1}{x} \neq 1$.

Korzystając z programu, który iteracyjnie co epsilon maszynowy sprawdzał, czy szukana liczba nie spełnia równania okazało się, że najmniejsza liczba x z przedziału [1, 2], która nie spełnia równiania to:

x = 1.000000057228997

a największa ma wartość:

x = 1.9999999850988384.

Oznacza to, że nie w każdym przypadku można precyzyjnie obliczyć odwrotność liczba, której odwrotność może zostać przybliżona.

Zadanie 5

Eksperymentalnie obliczyć iloczyn skalarny dwóch wektorów w różnej kolejności sumowania.

	Suma32	Suma64
a	1.0251881368296672e-10	1.0251881368296672e-10
b	-0.4543457	-1.5643308870494366e-10
c	-0.5	0.0
d	-0.5	0.0

Prawidłowa wartość: $-1.00657107000000*10^{-11}$.

W tym zadaniu można zauważyć, że nie powinno się sumować liczb bardzo bliskich lub bardzo odległych od siebie ze względu na redukcję cyfr znaczących, która ma wpływ na wynik końcowy. Warto pamiętać, że kolejność składników sumowanych ma znaczenie dla precyzji wyniku.

Zadanie 6

Policzyć w arytmetyce Float64 wartości dwóch równoważnych funkcji:

$$f(x) = \sqrt{x^2+1} - 1$$

$$g(x) = \frac{x^2}{\sqrt{x^2+1}+1}$$

f(x)	g(x)
0.0077822185373186414	0.0077822185373187065
0.00012206286282867573	0.00012206286282875901
1.9073468138230965e-6	1.907346813826566e-6
2.9802321943606103e-8	2.9802321943606116e-8
4.656612873077393e-10	4.6566128719931904e-10

W tabeli znajdują się wyniki dla kolejnych potęg 8^{-1}, 8^{-2}, itp. Można zauważyć, że funkcja f zwraca mniej dokładny wynik, a funkcja g jest bardziej precyzyjna. Dzieje się tak dlatego, że odejmujemy dwie bardzo bliskie sobie liczby, co dzięki poprzedniemu zadaniu zostało udowodnione, że nie otrzymuje się wtedy dokładnego wyniku, czyli precyzja jest bardzo mała, bo występuje redukcja cyfr znaczących.

Zadanie 7

Obliczyć przybliżoną wartość f'(x) dla f(x)=sin x + cos 3x

h	f'(x)	error
1	2.0179892252685967	1.9010469435800585
2	1.8704413979316472	1.753499116243109
3	1.1077870952342974	0.9908448135457593
4	0.6232412792975817	0.5062989976090435
5	0.3704000662035192	0.253457784514981
6	0.24344307439754687	0.1265007927090087
7	0.18009756330732785	0.0631552816187897
8	0.1484913953710958	0.03154911368255764
9	0.1327091142805159	0.015766832591977753
25	0.11694252118468285	2.394961446938737e-7
26	0.116942398250103	1.1656156484463054e-7
27	0.11694233864545822	5.6956920069239914e-8
44	0.1162109375	0.0007313441885381522
45	0.1171875	0.0002452183114618478
46	0.11328125	0.003661031688538152
47	0.109375	0.007567281688538152
48	0.109375	0.007567281688538152
49	0.09375	0.023192281688538152
50	0.125	0.008057718311461848
51	0.0	0.11694228168853815
52	0.0	0.11694228168853815
53	-0.5	0.6169422816885382
54	0.0	0.11694228168853815

Poprawny wynik: $f'(x) = \cos(x) - 3\sin(3x)$, a więc $f'(1) \approx 0.116942281$

Jak widać zmniejszanie wartości h przestaje od pewnego momentu przybliżać wynik, wręcz przeciwnie – zaczyna generować coraz większy błąd. Jest to wynik operacji na bliskich sobie liczbach w arytmetyce zmiennopozycyjnej, których precyzja jest bardzo mała.