1 Zadanie 1

Opis problemu: W zadaniu należało wyznaczyć następujące wartości:

- *Macheps* — najmniejsza liczba macheps > 0 taka, że $\mathrm{fl}(1.0 + \mathrm{macheps}) > 1.0$ i $\mathrm{fl}(1.0 + \mathrm{macheps}) = 1 + \mathrm{macheps}$
- *eta* — najmniejsza dodatnia liczba zmiennoprzecinkowa (najmniejsza liczba nieznormalizowana — $\mathrm{MIN_{sub}}$)
- liczba MAX — największa liczba nieznormalizowana — $\mathrm{MAX_{sub}}$

Oraz odpowiedzieć na pytania:

- Jaki związek ma liczba *macheps* z *precyzją arytmetyki* (oznaczaną na wykładzie przez ϵ)?
- Jaki związek ma liczba *eta* z liczbą $\mathrm{MIN_{sub}}$?
- Co zwracają funkcje floatmin(Float32) i floatmin(Float64) i jaki jest związek zwracanych wartości z liczbą $\mathrm{MIN_{nor}}$?

Wyniki:

1.1 *Macheps*

1.1.1 Float64

eps(Float64) = 2.220446049250313e-16
Mój wynik: 2.220446049250313e-16

1.1.2 Float32

eps(Float32) = 1.1920929e-07
Mój wynik: 1.1920929e-07

1.1.3 Float16

eps(Float16) = 0.000977
Mój wynik: 0.0009765625

1.2 *eta*

1.2.1 Float64

nextfloat(Float64(0.0)) = 5.0e-324
Mój wynik: 5.0e-324

1.2.2 Float32

nextfloat(Float32(0.0)) = 1.0e-45
Mój wynik: 1.0e-45

1.2.3 Float16

nextfloat(Float16(0.0)) = 6.0e-8
Mój wynik: 6.0e-8

1.3 MAX

1.3.1 Float64

floatmax(Float64(0.0)) = 1.7976931348623157e308
Mój wynik: 1.7976931348623157e308

1.3.2 Float32

floatmax(Float32(0.0)) = 3.4028235e38
Mój wynik: 3.4028235e38

1.3.3 Float16

floatmax(Float16(0.0)) = 6.55e4
Mój wynik: 6.55e4

1.4 Wartości z float.h:

1.4.1 FLOAT

FLT_MAX = 3.4028234664e+38
FLT_MIN = 1.1754943508e-38
FLT_EPSILON = 1.1920928955e-07

1.4.2 DOUBLE

DBL_MAX = 1.79769313486231570815e+308
DBL_MIN = 2.22507385850720138309e-308
DBL_EPSILON = 2.22044604925031308085e-16

1.4.3 LONG DOUBLE

LDBL_MAX = 1.18973149535723176502126385303 1e+4932
LDBL_MIN = 3.36210314311209350626267781732 2e-4932
LDBL_EPSILON = 1.08420217248550443400745280087 0e-19

1.5 Odpowiedzi na pytania:

1. Jaki związek ma liczba *macheps* z *precyzją arytmetyki* (oznaczaną na wykładzie przez ϵ)?
   Precyzja arytmetyki $\epsilon = 0.5\beta^{1-t}$(macheps=$\beta^{1-t}$). Wynika to z tego, że błąd musi spełniać założenie $|\delta| \leq \epsilon$ i $\mathrm{fl}(1.0 + \delta) = 1.0$. Oznacza to, że nie może zostać zaokrąglony w górę, ponieważ wtedy będzie już następną liczbą.

2. Jaki związek ma liczba *eta* z liczbą $\mathrm{MIN}_{\mathrm{sub}}$?
   $Min_{sub} = m_{min} * \beta^{c_{min}} = \beta^{1-t} * \beta^{c_{min}} = \beta^{c_{min}+(1-t)}$
   $MIN_{nor} = 1 * \beta^{c_{min}}$
   Liczba $eta = Min_{sub}$

3. Co zwracają funkcje floatmin(Float32) i floatmin(Float64) i jaki jest związek zwracanych wartości z liczbą $\mathrm{MIN}_{\mathrm{nor}}$?
   Floatmin jest MIN_{nor} dla danych typów

1.6 Wnioski

Wyniki uzyskane w zadaniu są zgodne z wartościami podanymi w języku. Zauważamy, że eps() oznacza najmniejszą różnicę między liczbami w zakresie mantysy, a nextfloat() zwraca najmniejszą możliwą następną liczbę.

2 Zadanie 2

Opis problemu: W zadanie trzeba było sprawdzić, czy wynik równania
$3(\frac{4}{3} - 1) - 1$ zwraca wartości epsilonu maszynowego

Wyniki:

2.1 Float64

eps(Float64) = 2.220446049250313e-16
Wynik równania: -2.220446049250313e-16

2.2 Float32

eps(Float32) = 1.1920929e-07
Wynik równania: 1.1920929e-07

2.3 Float16

eps(Float16) = 0.000977
Wynik równania: -0.000977

2.4 Wnioski

Wyniki uzyskane w zadaniu są zgodne z wartościami eps() dla danych typów. Wynika to z faktu, że w obliczeniach występują błędy zaokrągleń, które powodują, że wynik jest równy eps() lub jego przeciwieństwu.

3 Zadanie 3

Opis problemu: W zadaniu należało zbadać odległości między liczbami w arytmetyce zmiennoprzecinkowej w różnych przedziałach. Należało sprawdzić, czy liczba 2^{-52} jest taką odległością.

Wyniki:

3.1 Przedział [1,2]

2^{-52} jest odległością między kolejnymi liczbami w tym przedziale. Możemy stwierdzić to na podstawie binarnego zapisu tych liczb.
1: 0011111111110000000000000000000000000000000000000000000000000000
2^{-52}: 0011110010110000000000000000000000000000000000000000000000000000
1+2^{-52}: 0011111111110000000000000000000000000000000000000000000000000001
Różnią się one na ostatnim bicie, co oznacza, że 2^{-52} jest odległością między tymi liczbami.

3.2 Przedział [2,4]

W tym przedziale odległość między kolejnymi liczbami wynosi 2^{-51}. Możemy stwierdzić to na podstawie binarnego zapisu tych liczb.
2: 0100000000000000000000000000000000000000000000000000000000000000
2^{-52}: 0011110010110000000000000000000000000000000000000000000000000000
2+2^{-52}: 0100000000000000000000000000000000000000000000000000000000000000
Jak widać dodanie 2^{-52} do liczby 2 nie zmienia jej wartości, co oznacza, że 2^{-52} mniejsze od odległości między tymi liczbami. Natomiast dodając 2^{-51} do liczby 2 otrzymujemy otrzymujemy następną liczbę w tym przedziale.

3.3 Przedział [$\frac{1}{2}$,1]

$\frac{1}{2}$: 0011111111100000000000000000000000000000000000000000000000000000
2^{-52}: 0011110010110000000000000000000000000000000000000000000000000000
$\frac{1}{2}$+2^{-52}: 0011111111100000000000000000000000000000000000000000000000000010

Różnią się one na przedostatnim bicie, co oznacza, że 2^{-52} jest dwukrotnie większa od odległości między tymi liczbami.

3.4 Wnioski

W zadaniu udało się potwierdzić, że odległości między liczbami w arytmetyce zmiennoprzecinkowej zależą od przedziału, w którym się znajdują. W przedziale [1,2] odległość ta wynosi 2^{-52}, w przedziale [2,4] wynosi 2^{-51}, a w przedziale $[\frac{1}{2},1]$ wynosi 2^{-53}.

4 Zadanie 4

Opis problemu:
W zadaniu należało znaleźć taką liczbę zmiennoprzecinkową $x \in (1, 2)$, dla której $x * (\frac{1}{x}) \neq 1$. Należało również znaleźć najmniejszą taką liczbę.

4.1 Wynik

Liczba 1.000000057228997 jest najmniejsza liczbą spełniającą warunki zadania.

4.2 Wnioski

W x odległość między liczbami wynosi 2-52, ale w $\frac{1}{x}$ to już jest 2-53 (Ponieważ $\frac{1}{x} \in (\frac{1}{x}, 1)$). Wychodzi na to, że musi to być liczba, która przy dzieleniu jedynki zostanie zaokrąglona w górę oraz później przy mnożeniu przez siebie nie zostanie zaokrąglona w dół.

5 Zadanie 5

Opis problemu:
W zadaniu należało zaimplementować funkcje obliczające iloczyn skalarny dwóch wektorów.

Wyniki:

5.1 Float64

a) Wynik: 1.0251881368296672e-10

b) Wynik: -1.5643308870494366e-10

c) Wynik: 0.0

d) Wynik: 0.0

5.2 Float32

a) Wynik: -0.4999443

b) Wynik: -0.4543457

c) Wynik: -0.5

d) Wynik: -0.5

5.3 Wnioski

W zadaniu nie udało się zbytnio zaobserwować wpływu redukcji liczb znaczących na wynik, ponieważ przypadki, gdzie dane były posortowane dały nam takie same wyniki. Zauważyć na pewno można jednak, że kolejność dodawania ma znaczenie, ponieważ dodawanie "od przodu" i "od tyłu" dały różne wyniki. Kierując się informacjami z wykładu wiemy, że redukcja cyfr znaczących występuje w momencie dodawania liczb skrajnie mniejszych do większych. Natomiast w przypadku odejmowania redukcja ta nachodzi przy liczbach bardzo do siebie zbliżonych.
Poprawny wynik: $-1.00657107000000 * 10^{-11}$
Najbliżej poprawnego wyniku była metoda b) dla Float64, czyli dodawanie "od tyłu".

6 Zadanie 6

Opis problemu:
W zadniu należało zaimplementować dwie funkcje i porównać ich wyniki dla różnych wartości n.
Funkcje:

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

6.1 Wyniki

f(0.125)= 0.0077822185373186414
g(0.125)= 0.0077822185373187065
f(0.015625)= 0.00012206286282867573
g(0.015625)= 0.00012206286282875901
f(0.001953125)= 1.9073468138230965e-6
g(0.001953125)= 1.907346813826566e-6
f(0.000244140625)= 2.9802321943606103e-8
g(0.000244140625)= 2.9802321943606116e-8
f(3.0517578125e-5)= 4.656612873077393e-10
g(3.0517578125e-5)= 4.6566128719931904e-10

f(3.814697265625e-6)= 7.275957614183426e-12
g(3.814697265625e-6)= 7.275957614156956e-12
f(4.76837158203125e-7)= 1.1368683772161603e-13
g(4.76837158203125e-7)= 1.1368683772160957e-13
f(5.960464477539063e-8)= 1.7763568394002505e-15
g(5.960464477539063e-8)= 1.7763568394002489e-15
f(7.450580596923828e-9)= 0.0
g(7.450580596923828e-9)= 2.7755575615628914e-17
f(9.313225746154785e-10)= 0.0
g(9.313225746154785e-10)= 4.336808689942018e-19

6.2 Wnioski

Wyniki funkcji f i g są zbliżone dla większych wartości n, ale wraz ze zmniejszaniem się n, różnice między nimi rosną. Dla bardzo małych wartości n, funkcja f zwraca 0, podczas gdy funkcja g nadal zwraca wartości bliskie zeru, ale niezerowe. Wynika to z faktu, że w odejmowaniu w funkcji f następuje redukacja cufr znaczących. Wynika to z faktu, że $\sqrt{x^2+1}$ wraz ze wzrostem x zbliża się do 1, co powoduje, że różnica między nimi staje się bardzo mała i prowadzi do utraty precyzji.

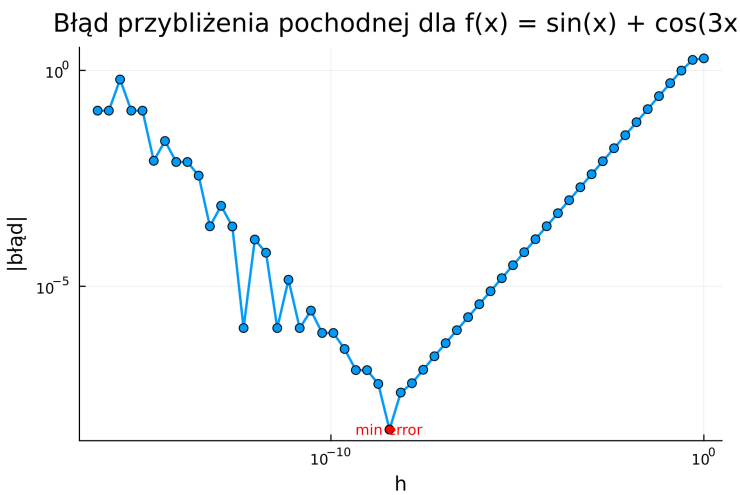
7 Zadanie 7

Opis problemu:
W zadaniu należało zaimplementować funkcję obliczającą wartość pochodnej funkcji $f(x) = \sin(x) + \cos(3x)$ w punkcie $x_0 = 1$ za pomocą wzoru $f`(x_0) = \frac{f(x_0+h)-f(x_0)}{h}$ oraz wzory pochodnej $f`(x) = \cos(x) - 3\sin(3x)$

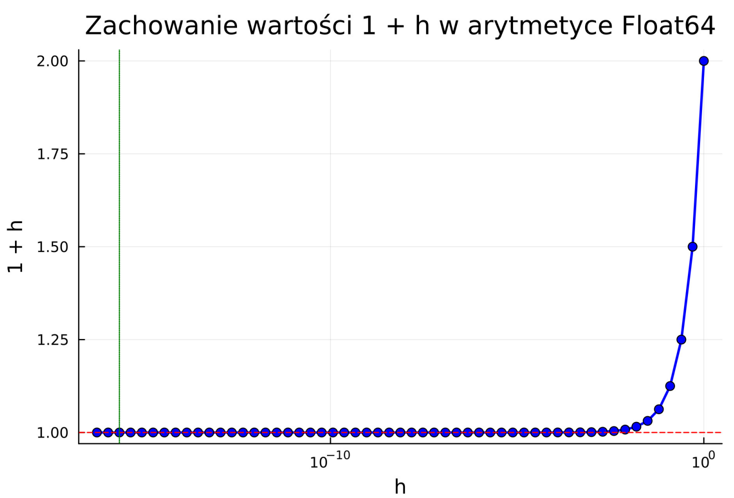
7.1 Wyniki

7.2 Wnioski

W zadaniu udało się zaobserwować, jak wartość h wpływa na błąd przybliżenia pochodnej. Dla dużych wartości h błąd jest duży, ponieważ przybliżenie jest niedokładne. Wraz ze zmniejsjaniem się h, błąd maleje, aż do pewnego momentu, gdzie zaczyna rosnąć ponownie. Dzieje się tak, ponieważ dla bardzo małych wartości h, różnica $f(x_0+h)-f(x_0)$ staje się bardzo mała i prowadzi do redukcji cyfr znaczących. Dodatkowo, na drugim wykresie widać, że dla bardzo małych wartości h, wartość $1+h$ jest równa 1, co również wskazuje na utratę precyzji.



Rysunek 1: Błąd przybliżenia pochodnej



Rysunek 2: Wartość 1 + h