Arytmetyka Komputerowa

Metody Obliczeniowe w Nauce i Technice

Paweł Fornagiel | 15 Marzec 2025

Analiza przypadku

Numeryczne przybliżenie wartości pochodnej w punkcie

Przybliżoną wartość pochodnej funkcji $f(x)$ w punkcie x można obliczyć ze wzoru:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h} \tag{1}$$

Wykorzystać ten wzór do obliczenia pochodnej funkcji $f(x) = \sin(x) + \cos(3x)$ w punkcie $x = 1$ dla $h = 2^i$ ($i = 0, 1, 2, \ldots, 40$). Wykonać obliczenia dla różnej precyzji zmiennych i zwrócić uwagę na typ argumentów i wyników dla funkcji bibliotecznych wykorzystywanych w obliczeniach.

- Jak wytłumaczyć, że od pewnego momentu zmniejszenie wartości h nie poprawia przybliżenia wartości pochodnej?
- Jak zachowują się wartości $1 + h$?
- Obliczone przybliżenia pochodnej porównać z dokładną wartością pochodnej. Obliczenia wykonać dla zmiennych typu `float`, `double`, `long double`.

Dane techniczne

Zadanie zostało przeprowadzone z użyciem narzędzi o następujących parametrach:

- Komputer HP EliteBook 840 G6:
  - System operacyjny: Windows 11 x64
  - Procesor Intel(R) Core(TM) i5-8365U CPU 1.60GHz 1.90 GHz
  - Pamięć RAM: 8GB
- Obliczenia
  - C++, g++ (Rev3, Built by MSYS2 project) 13.2.0
    - Biblioteki:
      - `<iostream>`, `<fstream>`, `<cfloat>`, `<cmath>`, `<sstream>`,
      - `<functional>`, `<type_traits>`,
- Wizualizacja:
  - Python 3.12.0
  - Biblioteki:
    - `numpy 1.26.1`, `matplotlib 3.8.0`, `pandas 2.1.1`

Specyfikacja liczb zmiennoprzecinkowych w języku C++

- `float`:
  - Rozmiar: 4 bajty
  - Precyzja (cyfry dziesiętne): 6
  - Wartość minimalna*: `1.17549e-38`
  - Wartość maksymalna*: `3.40282e+38`
  - Epsilon maszynowy: `1.19209e-07`
- `double`:
  - Rozmiar: 8 bajty
  - Precyzja (cyfry dziesiętne): 15
  - Wartość minimalna*: `2.22507e-308`
  - Wartość maksymalna*: `1.79769e+308`
  - Epsilon maszynowy: `2.22045e-16`
- `long double`:
  - Rozmiar: 16 bajty
  - Precyzja (cyfry dziesiętne): 18
  - Wartość minimalna*: `3.3621e-4932`
  - Wartość maksymalna*: `1.18973e+4932`
  - Epsilon maszynowy: `1.0842e-19`

*Zgodnie z https://en.cppreference.com/w/c/types/limits

Program wyznaczający wartość pochodnej

Program wyznaczający pochodną w języku C++

```
template <typename T>
  T f(T x){
    return sin(x) + cos(3*x);
  }

  template<typename T>
  T derivative(T x, T h, std::function<T(T)> f){
    return (f(x + h) - f(x))/h;
  }

  template <typename T>
  T real_derivative(T x){
    return cos(x) - 3*sin(3*x);
  }
```

Analiza poprawności wartości pochodnej

Wartości pochodnej

i	Wartość precyzji `float`	Wartość precyzji `double`	Wartość precyzji `long double`
0	2.017989158630371	2.0179892252685967	2.0179892252685967
1	1.8704414367675781	1.8704413979316472	1.8704413979316477
2	1.1077871322631836	1.1077870952342974	1.1077870952342976
3	0.6232414245605469	0.6232412792975816	0.6232412792975818
4	0.37040042877197266	0.3704000662035191	0.3704000662035182
5	0.24344444274902344	0.2434430743975468	0.2434430743975488
6	0.1800994873046875	0.1800975633073278	0.1800975633073301
7	0.14849090576171875	0.1484913953710957	0.1484913953711071
8	0.132720947265625	0.1327091142805159	0.1327091142805097
9	0.124847412109375	0.1248236929407084	0.1248236929407228
10	0.12091064453125	0.1208824768110616	0.1208824768110877
11	0.1190185546875	0.1189122504688384	0.1189122504689302
12	0.117919921875	0.1179272337390102	0.1179272337391594
13	0.11767578125	0.1174347496107657	0.1174347496109122
14	0.1181640625	0.1171885136209311	0.1171885136217349
15	0.1171875	0.1170653971457795	0.1170653971478579
16	0.12109375	0.1170038392883725	0.1170038392913426
17	0.125	0.1169730604597134	0.1169730604582213
18	0.125	0.1169576710672117	0.1169576710654496
19	0.125	0.1169499763636849	0.1169499763750252
20	0.125	0.1169461290119215	0.1169461290312483
21	0.25	0.1169442052487283	0.1169442053598004
22	0.25	0.1169432429596781	0.116943243524247
23	0.5	0.1169427623972296	0.1169427626064134
24	0.0	0.1169425211846828	0.1169425221478377
25	0.0	0.1169423982501029	0.1169424019190046

i	Wartość precyzji `float`	Wartość precyzji `double`	Wartość precyzji `long double`
26	0.0	0.1169423386454582	0.1169423418032238
27	0.0	0.1169423162937164	0.1169423117462429
28	0.0	0.116942286491394	0.1169422967213904
29	0.0	0.1169422268867492	0.1169422891980502
30	0.0	0.1169421672821044	0.1169422854436561
31	0.0	0.1169421672821044	0.1169422834645956
32	0.0	0.1169419288635253	0.1169422825332731
33	0.0	0.1169414520263671	0.1169422823004424
34	0.0	0.1169414520263671	0.1169422818347811
35	0.0	0.1169395446777343	0.1169422827661037
36	0.0	0.116943359375	0.1169422827661037
37	0.0	0.1169281005859375	0.116942286491394
38	0.0	0.116943359375	0.116942286491394
39	0.0	0.11688232421875	0.116942286491394
40	0.0	0.1168212890625	0.116942286491394

Tabela 1: Wartości pochodnej obliczonej numerycznie z użyciem różnych precyzji reprezentacji liczb zmiennoprzecinkowych.

i - wartość bezwzględna wykładnika w wyrażeniu $h = 2^{-i}$ użytego w *Równanie (1)*

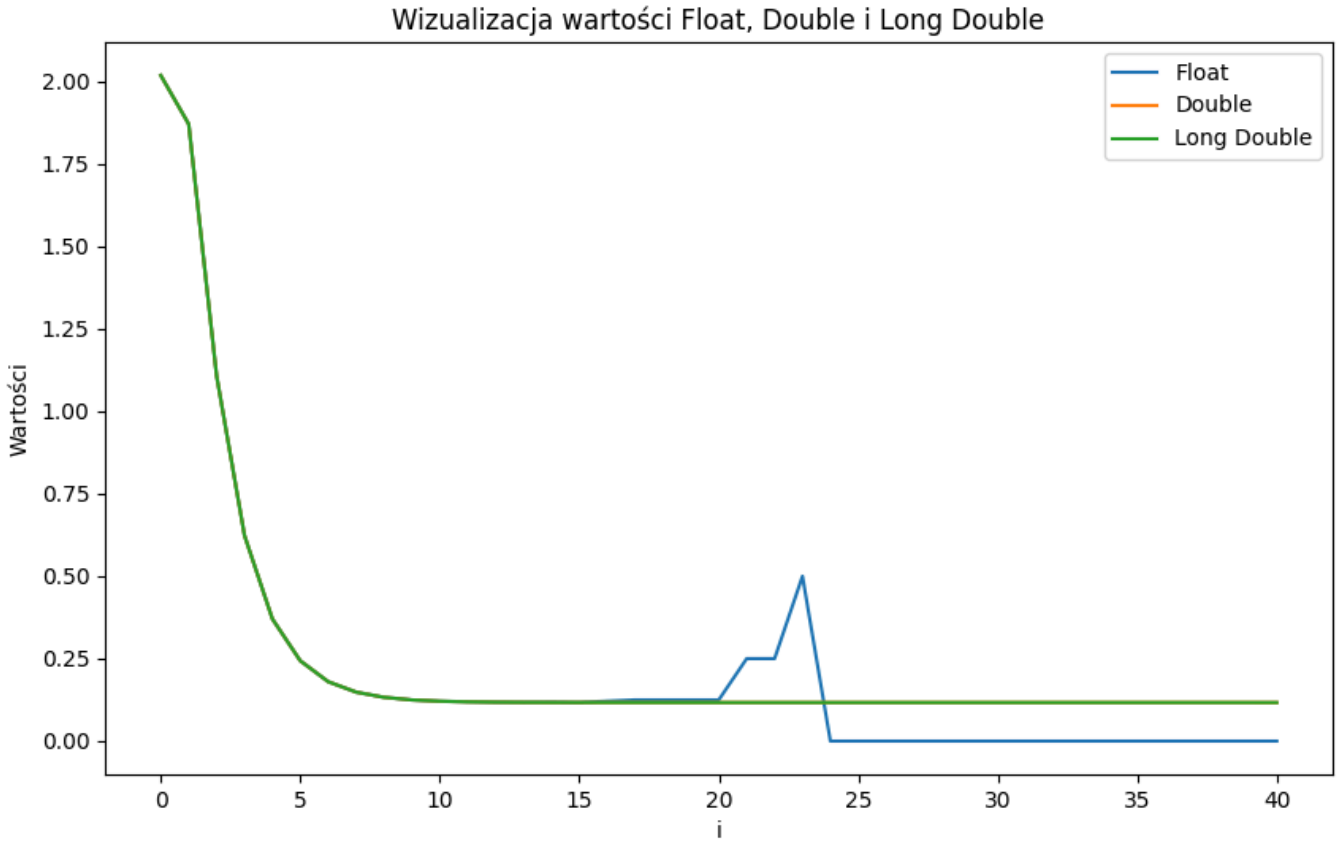
Prawdziwa* wartość pochodnej w punkcie $x = 1$:

$$f'(1) = \cos(1) - 3\sin(3) \approx$$

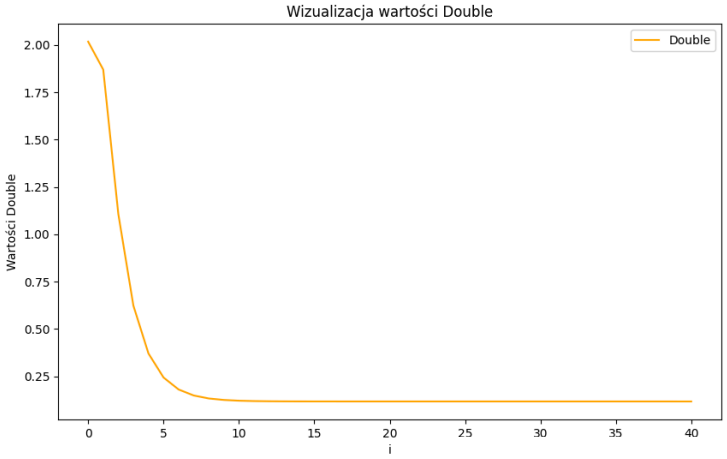
$$\approx 0.11694228168853805109870219901864576419151062786 11...$$

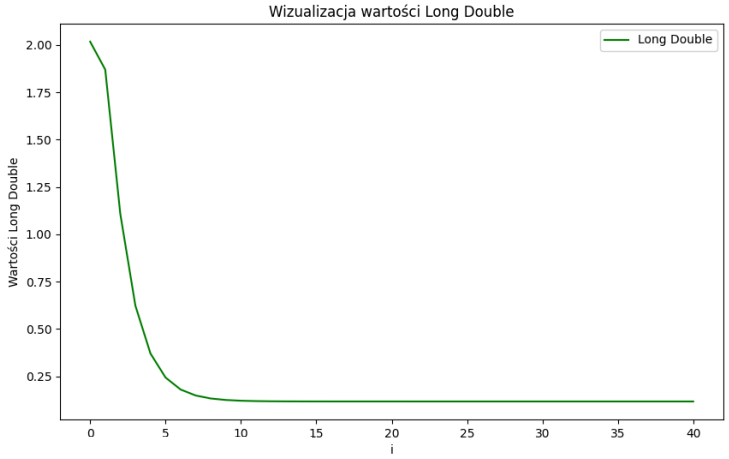
**Obliczone z pomocą WolframAlpha [1]*

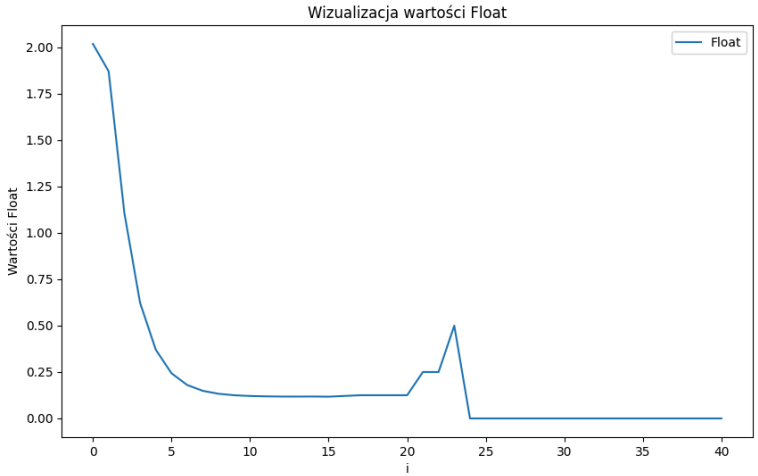
Wizualizacja otrzymanych wartości pochodnej



Wizualizacja otrzymanych wartości pochodnej







Błedy obliczeń względem wartości prawdziwej

i	Błąd względny precyzji `float`	Błąd względny precyzji `double`	Błąd względny precyzji `long double`
0	1.9010468721389768	1.9010469435800583	1.9010469435800583
1	1.753499150276184	1.753499116243109	1.7534991162431093
2	0.9908448457717896	0.9908448135457592	0.9908448135457596
3	0.5062991380691528	0.5062989976090435	0.5062989976090438
4	0.2534581422805786	0.253457784514981	0.2534577845149802
5	0.1265021562576293	0.1265007927090087	0.1265007927090108
6	0.0631572008132934	0.0631552816187896	0.063155281618792
7	0.0315486192703247	0.0315491136825576	0.0315491136825691
8	0.0157786607742309	0.0157668325919777	0.0157668325919716
9	0.0079051256179809	0.0078814112521703	0.0078814112521847
10	0.0039683580398559	0.0039401951225235	0.0039401951225497
11	0.0020762681961059	0.0019699687803003	0.0019699687803922
12	0.0009776353836059	0.0009849520504721	0.0009849520506214
13	0.0007334947586059	0.0004924679222275	0.0004924679223742
14	0.0012217760086059	0.000246231932393	0.0002462319331969
15	0.0002452135086059	0.0001231154572414	0.0001231154593198
16	0.0041514635086059	6.155759983439424e-05	6.155760280456398e-05
17	0.0080577135086059	3.077877117529937e-05	3.0778769683260725e-05
18	0.0080577135086059	1.5389378673624776e-05	1.5389376911579893e-05
19	0.0080577135086059	7.694675146829866e-06	7.69468648719303e-06
20	0.0080577135086059	3.8473233834324105e-06	3.847342710295925e-06
21	0.1330577135086059	1.9235601902423127e-06	1.9236712623838686e-06
22	0.1330577135086059	9.612711400208698e-07	9.61835708958097e-07
23	0.3830577135086059	4.807086915192826e-07	4.809178754017923e-07
24	0.116942286491394	2.394961446938737e-07	2.404592996841531e-07

i	Błąd względny precyzji `float`	Błąd względny precyzji `double`	Błąd względny precyzji `long double`
25	0.116942286491394	1.1656156484463053e-07	1.2023046657268438e-07
26	0.116942286491394	5.6956920069239914e-08	6.011468577489737e-08
27	0.116942286491394	3.460517827846843e-08	3.005770487070564e-08
28	0.116942286491394	4.802855890773117e-09	1.5032852397416866e-08
29	0.116942286491394	5.4801788884617515e-08	7.509512224351202e-09
30	0.116942286491394	1.1440643366000813e-07	3.755118095432556e-09
31	0.116942286491394	1.1440643366000813e-07	1.776057624374664e-09
32	0.116942286491394	3.528250127615707e-07	8.447350497591856e-10
33	0.116942286491394	8.296621709646957e-07	6.11904406105316e-10
34	0.116942286491394	8.296621709646957e-07	1.462431187975767e-10
35	0.116942286491394	2.737010803777196e-06	1.077565693413055e-09
36	0.116942286491394	1.0776864618478044e-06	1.077565693413055e-09
37	0.116942286491394	1.4181102600652196e-05	4.8028559918749685e-09
38	0.116942286491394	1.0776864618478044e-06	4.8028559918749685e-09
39	0.116942286491394	5.99574697881522e-05	4.8028559918749685e-09
40	0.116942286491394	0.0001209926260381	4.8028559918749685e-09

Tabela 2: Wartości błędów pochodnej obliczonej numerycznie względem prawdziwej* wartości pochodnej.

i - wartość bezwzględna wykładnika w wyrażeniu $h = 2^{-i}$ użytego w *Równanie (1)*

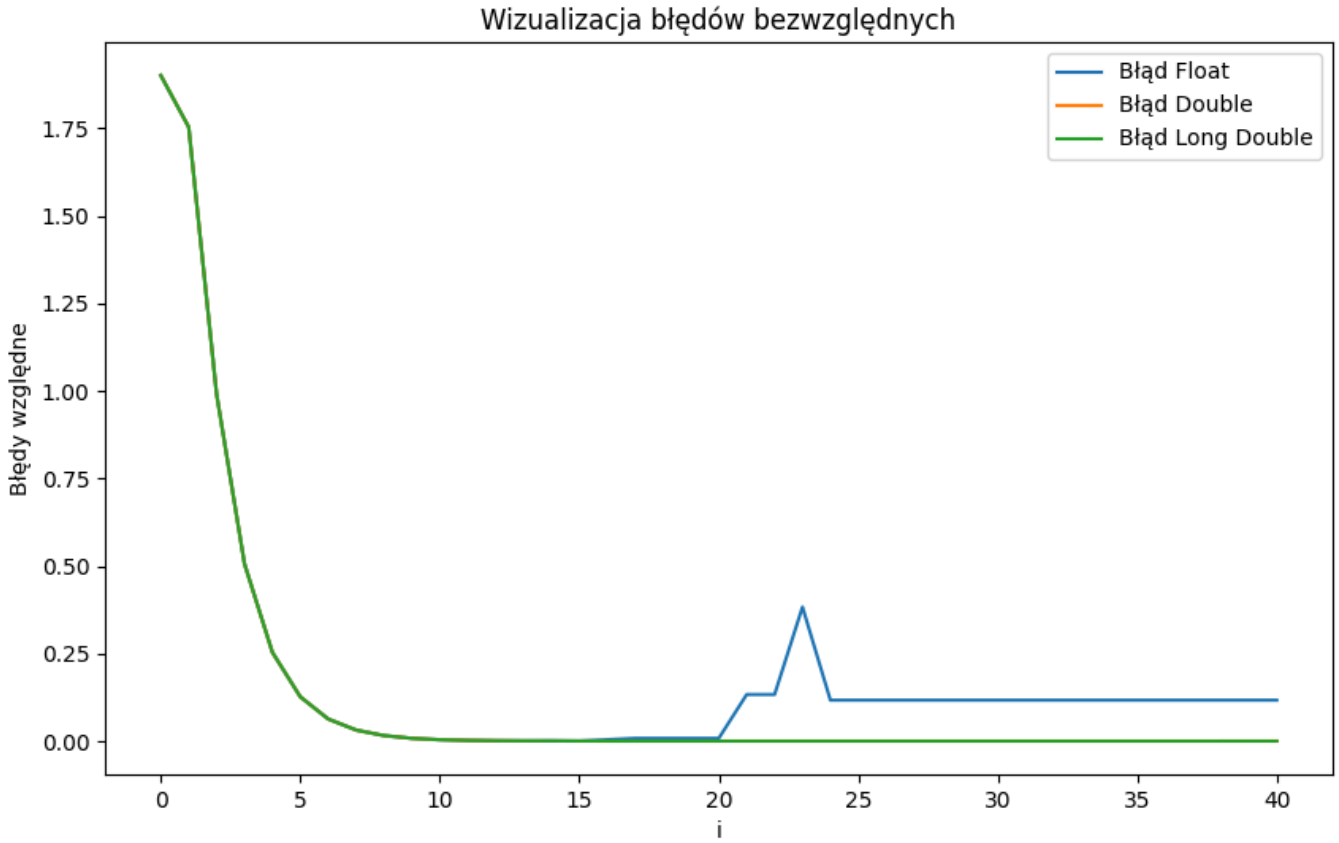
Prawdziwa* wartość pochodnej w punkcie $x = 1$:

$$f'(1) = \cos(1) - 3\sin(3) \approx$$

$$\approx 0.11694228168853805109870219901864576419151062786 11...$$

**Obliczone za pomocą programu WolframAlpha [1]*

Wizualizacja błędów obliczeń względem wartości prawdziwej



Problem z arytmetyką zmiennoprzecinkową

Jaki jest problem podczas obliczania pochodnej?

Weźmy wyrażenie $f(1)^*$:

- Wynik otrzymany programem WolframAlpha: $x = -0.14852151179254895 06$
- Wynik otrzymany z precyzją `double` w C++ : $x' = -0.14852151168363803 00$

$$|x - x'| \approx 1.089109205990704731 0 \times 10^{-}10$$

Dla wyrażenia $f(1) - f(1 + 2^{-30}))^*$:

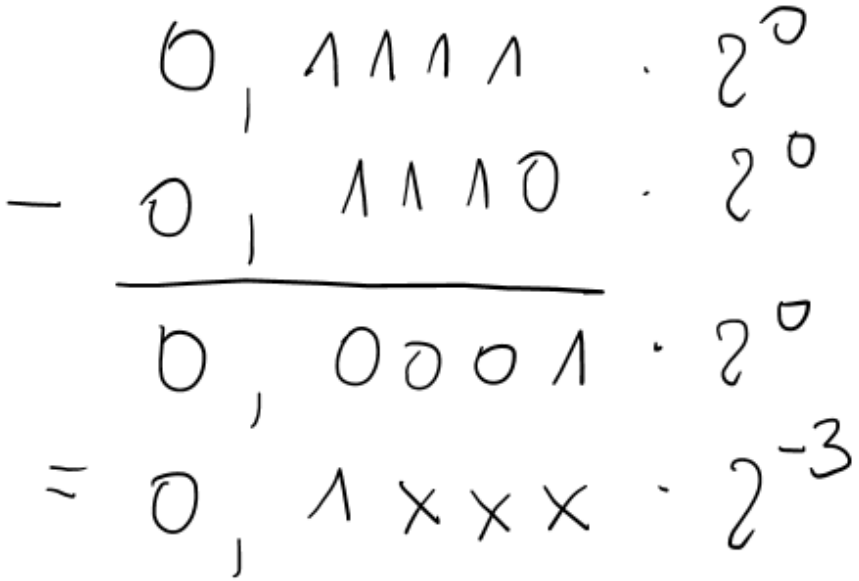
- Wynik otrzymany programem WolframAlpha: $y = -1.089109903627151745 1 \times 10^{-10}$
- Wynik otrzymany z precyzją `double` w C++ : $y' = -1.089108803142835313 3 \times 10^{-10}$

$$|y - y'| \approx -1.1004843164319 \times 10^{-16}$$

Błąd bezwzględny wartości liczby jest wielokrotnie większy od różnicy wartości dwóch liczb „bliskich” według ich wartości bezwzględnej. Obliczenia stają się niedokładne podczas wykonywania operacji odejmowania.

*Zgodnie z oznaczeniami w Równanie (1)

Problem z arytmetyką zmiennoprzecinkową



Wizualizacja `Catastrophic Cancellation`

Źródło: dr inż. Katarzyna Rycerz, wykład z przedmiotu Metody Obliczeniowe w Nauce i Technice

Wnioski

Dlaczego od pewnego momentu zmniejszanie wartości h h nie poprawia przybliżenia?

Gdy h staje się zbyt małe, błąd zaokrągleń związany z odejmowaniem bliskich sobie wartości $f(x+h)$ i $f(x)$ zaczyna dominować nad błędem metody (błąd obcięcia). Różnica $f(x+h) - f(x)$ traci precyzję w reprezentacji zmiennoprzecinkowej, a dzielenie przez bardzo małe h wzmacnia ten błąd.

W efekcie przybliżenie staje się mniej dokładne, mimo teoretycznej poprawy metody.

Analiza wartości wyrażenia $h + 1$

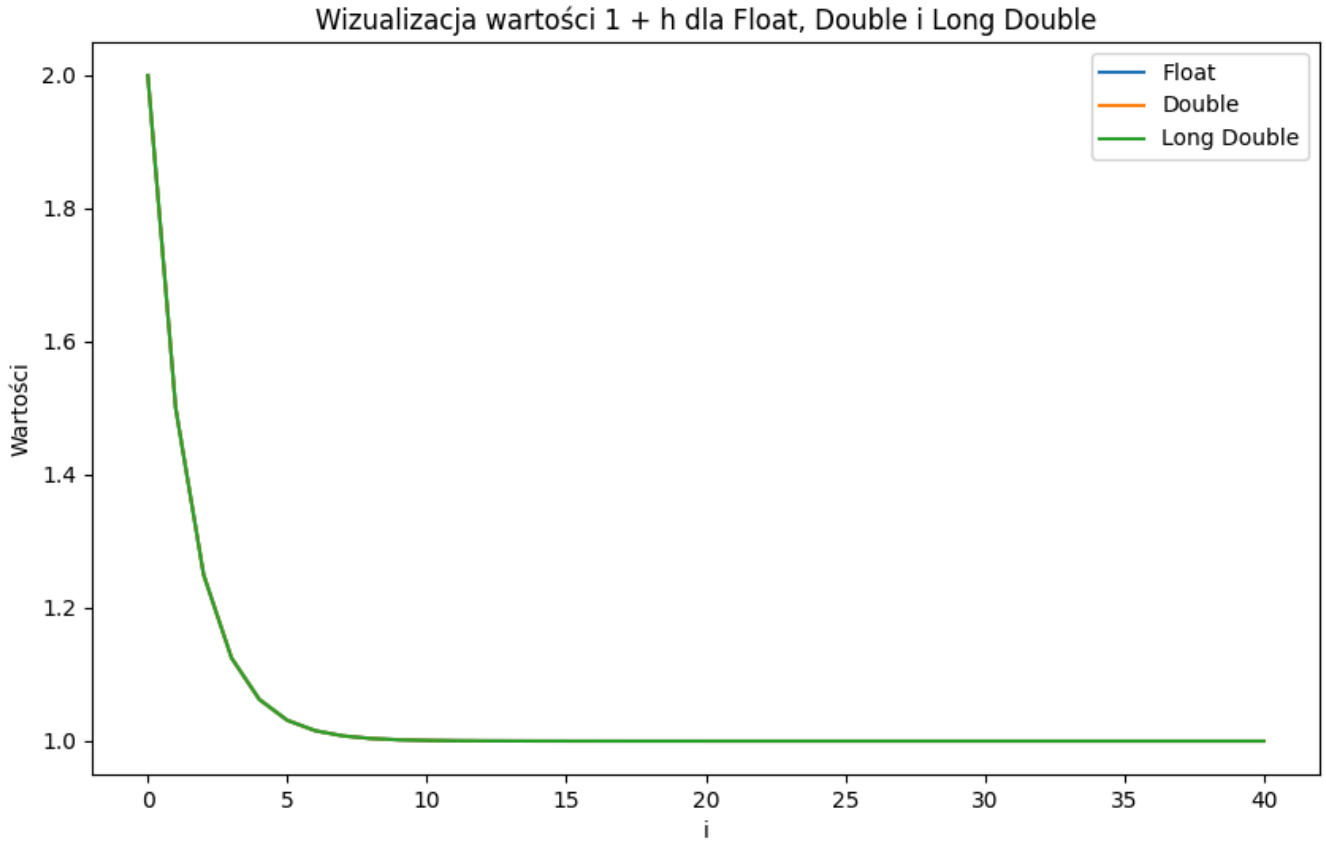
Wartości $1 + h$

i	Wartość precyzji `float`	Wartość precyzji `double`	Wartość precyzji `long double`
0.0	2.0	2.0	2.0
1	1.5	1.5	1.5
2	1.25	1.25	1.25
3	1.125	1.125	1.125
4	1.0625	1.0625	1.0625
5	1.03125	1.03125	1.03125
6	1.015625	1.015625	1.015625
7	1.0078125	1.0078125	1.0078125
8	1.00390625	1.00390625	1.00390625
9	1.001953125	1.001953125	1.001953125
10	1.0009765625	1.0009765625	1.0009765625
11	1.00048828125	1.00048828125	1.00048828125
12	1.000244140625	1.000244140625	1.000244140625
13	1.0001220703125	1.0001220703125	1.0001220703125
14	1.00006103515625	1.00006103515625	1.00006103515625
15	1.000030517578125	1.000030517578125	1.000030517578125
16	1.0000152587890625	1.0000152587890625	1.0000152587890625
17	1.0000076293945312	1.0000076293945312	1.0000076293945312
18	1.0000038146972656	1.0000038146972656	1.0000038146972656
19	1.0000019073486328	1.0000019073486328	1.0000019073486328
20	1.0000009536743164	1.0000009536743164	1.0000009536743164
21	1.0000004768371582	1.0000004768371582	1.0000004768371582
22	1.000000238418579	1.000000238418579	1.000000238418579
23	1.0000001192092896	1.0000001192092896	1.0000001192092896
24	1.0	1.0000000596046448	1.0000000596046448
25	1.0	1.0000000298023224	1.0000000298023224
26	1.0	1.0000000149011612	1.0000000149011612
27	1.0	1.0000000074505804	1.0000000074505804
28	1.0	1.0000000037252903	1.0000000037252903
29	1.0	1.0000000018626451	1.0000000018626451
30	1.0	1.0000000009313224	1.0000000009313224
31	1.0	1.0000000004656613	1.0000000004656613
32	1.0	1.0000000002328306	1.0000000002328306
33	1.0	1.000000000116415	1.000000000116415
34	1.0	1.0000000000582077	1.0000000000582077
35	1.0	1.0000000000291038	1.0000000000291038
36	1.0	1.000000000014552	1.000000000014552
37	1.0	1.000000000007276	1.000000000007276
38	1.0	1.000000000003638	1.000000000003638
39	1.0	1.0000000000018188	1.0000000000018188
40	1.0	1.0000000000009095	1.0000000000009095

Tabela 3: Wartości wyrażenia $1 + h$ dla $h = 2^{-i}$ dla różnej precyzji reprezentacji liczb zmiennoprzecinkowych.

i - wartość bezwzględna wykładnika w wyrażeniu $h = 2^{-i}$ użytego w *Równanie (1)*

Wizualizacja wartości $1 + h$



Wartości błędu $1 + h$ precyzji `float` względem precyzji `double`

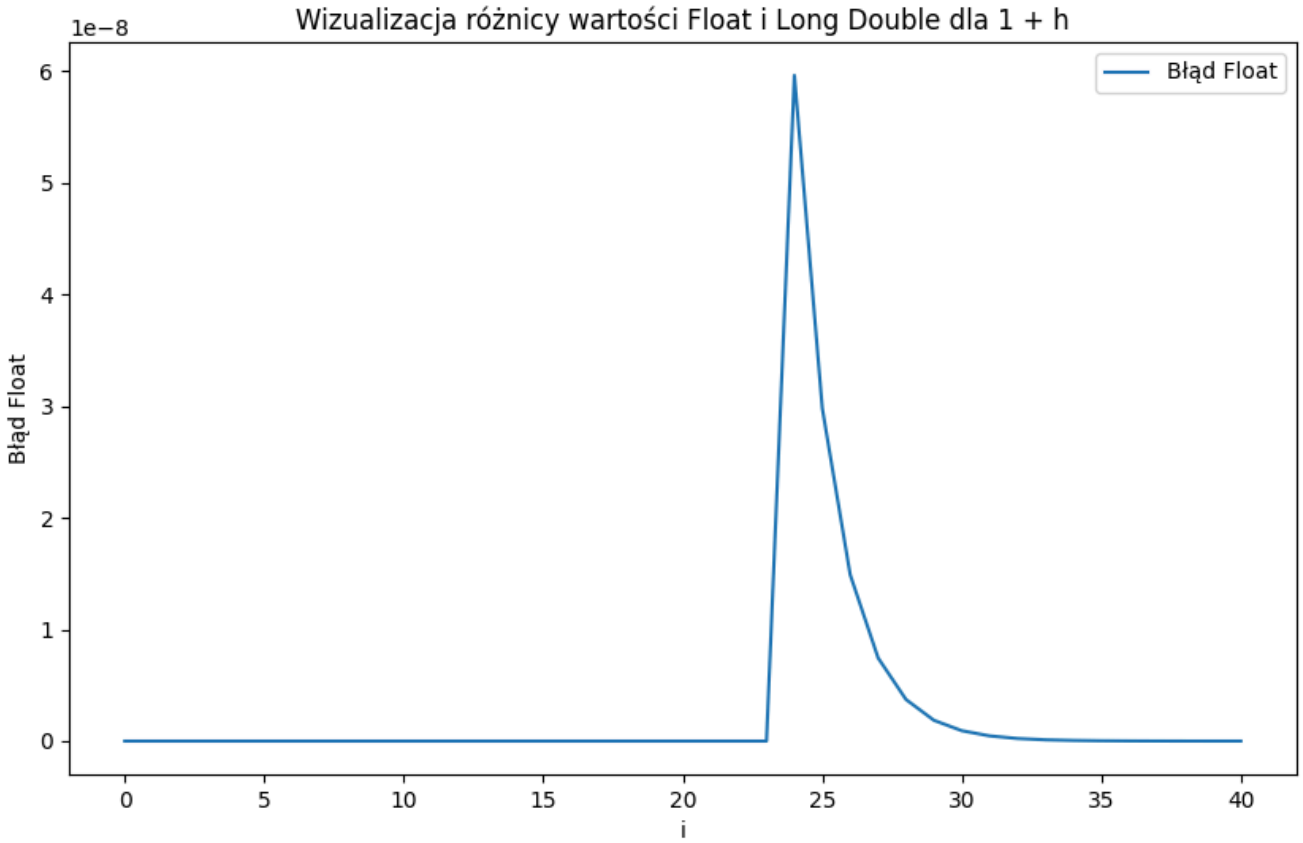
i	Błąd prezycji `float` względem `double long`
0	0.0
1	0.0
2	0.0
3	0.0
4	0.0
5	0.0
6	0.0
7	0.0
8	0.0
9	0.0
10	0.0
11	0.0
12	0.0
13	0.0
14	0.0
15	0.0
16	0.0
17	0.0
18	0.0
19	0.0
20	0.0
21	0.0
22	0.0
23	0.0
24	5.960464477539064e-08
25	2.9802322387695312e-08

i	Błąd prezycji `float` względem `double long`
26	1.4901161193847656e-08
27	7.4505805969238265e-09
28	3.7252902984619132e-09
29	1.8626451492309566e-09
30	9.313225746154783e-10
31	4.656612873077393e-10
32	2.3283064365386958e-10
33	1.1641532182693479e-10
34	5.820766091346741e-11
35	2.9103830456733704e-11
36	1.4551915228366852e-11
37	7.275957614183428e-12
38	3.637978807091713e-12
39	1.8189894035458565e-12
40	9.094947017729282e-13

Tabela 3: Wartości wyrażenia $1 + h$ dla $h = 2^{-i}$ dla różnej precyzji reprezentacji liczb zmiennoprzecinkowych.

i - wartość bezwzględna wykładnika w wyrażeniu $h = 2^{-i}$ użytego w *Równanie (1)*

Wizualizacja wartości błędu $1 + h$ precyzji `float` względem `long double`



Wnioski

Jak zachowują się wartości $1 + h$?

Dla h mniejszych niż epsilon maszynowy danego typu danych (np. $h = 2^{-24}$ dla `float`), wartość $1 + h$ jest reprezentowana jako 1. Wynika to z ograniczonej precyzji liczb zmiennoprzecinkowych — dodanie bardzo małego h nie zmienia wartości 1.

Dodatek | Wolframalpha

[1] Cytując dokumentację dotyczącą precyzji obliczeń narzędzia WolframAlpha:

WolframAlpha has the power to do computations in arbitrary precision, eliminating the cumulative error that arises when fixed point, floating point or conventional integer representations are used [...]

[...] In calculations involving arbitrary-precision approximate numbers, the Wolfram Language tracks the propagation of the numerical error. The use of high-precision numbers can yield accurate results where other numerical systems fail.