Metody Obliczeniowe w Nauce i Technice
Natalia Bratek, 10.03.2024
Laboratorium 01

Arytmetyka komputerowa

1. Treści zadań

1.1 Znaleźć "maszynowe epsilon", czyli najmniejszą liczbę a, taką, że a+1>1.

1.2 Rozważamy problem ewaluacji funkcji sin(x), m.in. propagację błędu danych wejściowych, tj. błąd wartości funkcji ze względu na zakłócenie h w argumencie x:

a) Ocenić błąd bezwzględny przy ewaluacji sin(x)

b) Ocenić błąd względny przy ewaluacji sin(x)

c) Ocenić uwarunkowanie dla tego problemu

d) Dla jakich wartości argumentu x problem jest bardzo czuły?

1.3 Funkcja sinus zadana jest nieskończonym ciągiem

sin(x) = x – x^3/3! + x^5/5! – x^7/7! + …

a) Jakie są błędy progresywny i wsteczny, jeśli przybliżamy funkcję sinus biorąc tylko pierwszy człon rozwinięcia, tj. sin(x) ≈ x, dla x = 0.1, 0.5 i 1.0?

b) Jakie są błędy progresywny i wsteczny, jeśli przybliżamy funkcję sinus biorąc pierwsze dwa człony rozwinięcia, tj. sin(x) ≈ x - x^3/6, dla x = 0.1, 0.5 i 1.0?

1.4 Zakładamy, że mamy znormalizowany system zmiennoprzecinkowy z β = 10, p = 3, L = -98.

a) Jaka jest wartość poziomu UFL (underflow) dla takiego systemu?
b) Jeśli x = 6.87 x 10^(-97) i y = 6.81 x 10^(-97), jaki jest wynik operacji x – y?

2. Rozwiązania zadań

2.1 Wyznaczanie maszynowego epsilonu

Maszynowy epsilon to najmniejsza liczba, którą dodając do jedności otrzymujemy wynik różny od jedności.
Funkcja findMachineEpsilon() najpierw ustawia epsilon na wartość 1.0. Ta wartość jest początkowym kandydatem na maszynowe epsilon, który będzie stopniowo

zmniejszany. Funkcja używa pętli while do powtarzania procesu dzielenia wartości epsilon przez 2, dopóki dodanie epsilon do 1 daje wynik różny od 1. W każdej iteracji pętli wartość eps jest dzielona przez 2.0, co stopniowo zmniejsza jej wartość, przybliżając się do momentu, w którym eps + 1.0 będzie równe 1.0 z dokładnością do precyzji. Zanim eps zostanie zaktualizowane do nowej wartości, aktualna wartość eps jest zapisywana w zmiennej epsLast. To zapewnia, że po wyjściu z pętli, epsLast zawiera ostatnią wartość eps, dla której eps + 1.0 było różne od 1.0. Po zakończeniu pętli, funkcja zwraca epsLast, która reprezentuje maszynowe epsilon.

```
def findMachineEpsilon():
    eps = 1.0
    while(eps + 1.0) != 1.0:
        epsLast = eps
        eps = eps / 2.0
    return epsLast
```

```
findMachineEpsilon()
```

```
2.220446049250313e-16
```

Uzyskana wartość maszynowego epsilonu wynosi 2.220446049250313e-16 oraz zgadza się z rzeczywistą wartością maszynowego epsilonu.

2.2 Problem ewaluacji funkcji sin(x)

2.2.1 Ocena błędu bezwzględnego przy ewaluacji sin(x)

$$y = \sin x$$

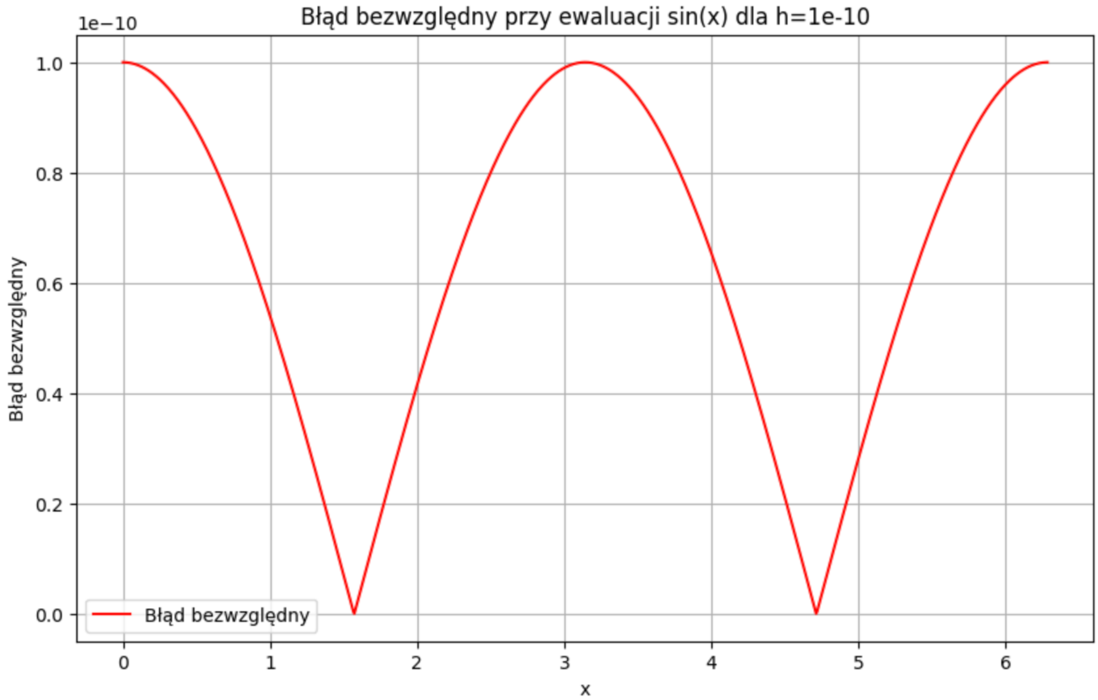
Wartość błędu bezwzględnego można wyrazić za pomocą wzoru, gdzie h to zakłócenie:

$$\Delta y = |\sin(x + h) - sinx)|$$

Błąd bezwzględny Δy funkcji y=sin(x) spowodowany małym zakłóceniem h w argumencie *x* można przybliżać za pomocą pierwszej pochodnej funkcji sin(*x*) względem *x*.

$$\lim_{h \to 0} \frac{\sin(x + h) - \sin(x)}{h} = \cos(x)$$

Oznacza to, że błąd bezwzględny ewaluacji sin(x) spowodowany przez zakłócenie h w argumencie x jest w przybliżeniu równy |h cos(x)|.



Rys.1

Na wykresie przedstawiono błąd bezwzględny przy ewaluacji sin(x) dla zakłócenia h=1e−10 w zakresie od 0 do 2π.

2.2.2 Ocena błędu względnego przy ewaluacji sin(x)

$$y = \sin x$$

Wartość błędu względnego można wyrazić za pomocą wzoru, gdzie h to zakłócenie:

$$\Delta y = |\frac{\sin(x+h) - sinx}{\sin(x)}|$$

Analogicznie jak w poprzednim podpunkcie wzór można oszacować przez:

Błąd względny = $\frac{|h\cos(x)|}{|\sin(x)|}$

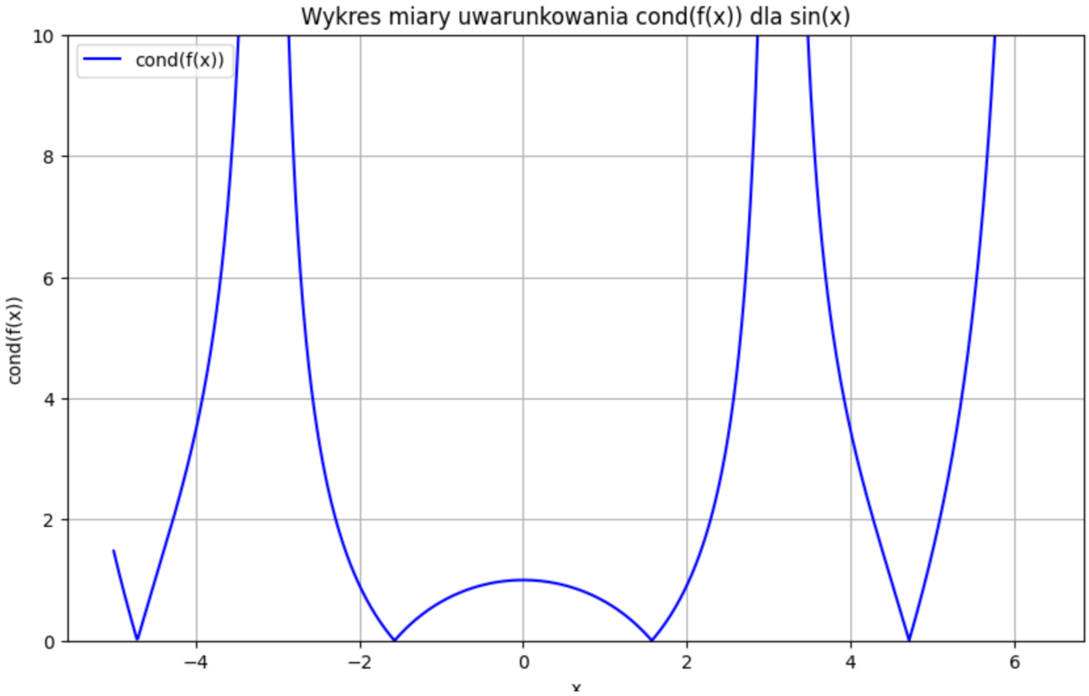
Jest to przybliżenie, które zakłada, że |sin(x)| nie jest równe 0. Dla wartości x, dla których sin(x)=0 (np. x=0, x=π, x=2π), błąd względny nie jest dobrze zdefiniowany, ponieważ w mianowniku występuje 0.

2.2.3 Uwarunkowanie problemu

Wskaźnik uwarunkowania definiuje się jako maksymalny stosunek błędu względnego rozwiązania do błędu względnego danych.

Dla funkcji $f(x)=\sin(x)$, miara uwarunkowania cond w punkcie x przy małym zakłóceniu h może być zdefiniowana jako:

$$cond\big(f(x)\big) = cond(\sin(x)) = \left|\frac{x\, f'(x)}{f(x)}\right| = \left|\frac{x\, cosx}{sinx}\right|$$



Rys.2

2.2.4 Czułość

Czułość problemu ewaluacji funkcji sin(x) to wrażliwość na błędy wejściowe.

Wartości x bliskie wielokrotnościom π, gdzie sin(x)=0, powodują, że mianownik wyrażenia cond(x) staje się bardzo mały, co z kolei może powodować, cond(x) staje się bardzo duże. To sugeruje, że problem jest źle uwarunkowany w tych punktach.
Dla bardzo dużych wartości x, wartość cond(x) również może wzrosnąć, ponieważ x znajduje się w liczniku wyrażenia.
W rzeczywistości największa czułość (najgorsze uwarunkowanie) występuje w okolicach punktów, gdzie wartość sin(x) przechodzi przez zero.

2.3 Przybliżenie funkcji sin szeregiem Taylora

Błąd progresywny to wartość bezwzględna z różnicy wartości rzeczywistej i przybliżonej.

$$\Delta y = |y - \hat{y}|$$

Błąd wsteczny to wartość bezwzględna z różnicy argumentu wstawionego do funkcji i argumentu, dla którego przybliżona wartość funkcji jest wartością rzeczywistą.

$$\Delta x = |\arcsin(\hat{y}) - x] = |\arcsin(x) - x|$$

2.3.1 Uwzględnienie pierwszego członu rozwinięcia

x = 0.1

y = sin(x)
$\hat{y} \approx x$
$\Delta y = |y - \hat{y}|$
Błąd progresywny:
$\Delta y = |\sin(x) - x| = |\sin(0.1) - 0.1| \approx 0.000166583$
Błąd wsteczny:
$\Delta x = |\arcsin(x) - x| \approx 0.000167421$

x = 0.5

y = sin(x)
$\hat{y} \approx x$
$\Delta y = |y - \hat{y}|$
Błąd progresywny:
$\Delta y = |\sin(x) - x| = |\sin(0.5) - 0.5| \approx 0.020574461$
Błąd wsteczny:
$\Delta x = |\arcsin(x) - x| \approx 0.023598776$

x = 1.0

y = sin(x)
$\hat{y} \approx x$
$\Delta y = |y - \hat{y}|$
Błąd progresywny:
$\Delta y = |\sin(x) - x| = |\sin(1.0) - 1.0| \approx 0.158529015$

Błąd wsteczny:
$\Delta x = |\arcsin(x) - x| \approx 0.570796327$

2.3.2 Uwzględnienie pierwszych dwóch członów rozwinięcia

x = 0.1

y = sin(x)
$\hat{y} \approx x - \frac{x^3}{6}$

$\Delta y = |y - \hat{y}|$
Błąd progresywny:
$\Delta y = |\sin(x) - x + \frac{x^3}{6}| = |\sin(0.1) - 0.1 + \frac{0.1^3}{6}| = 0.000000083$

Błąd wsteczny:
$\Delta x = |\arcsin\left(x - \frac{x^3}{6}\right) - x| = |\arcsin\left(0.1 - \frac{0.1^3}{6}\right) - 0.1| \approx$
0.000000084

x = 0.5

y = sin(x)
$\hat{y} = x = 0.5$
$\Delta y = |y - \hat{y}|$
Błąd progresywny:
$\Delta y = |\sin(x) - x + \frac{x^3}{6}| = |\sin(0.5) - 0.5 + \frac{0.5^3}{6}| \approx 0.000258872$

Błąd wsteczny:
$\Delta x = |\arcsin\left(x - \frac{x^3}{6}\right) - x| = |\arcsin\left(0.5 - \frac{0.5^3}{6}\right) - 0.5| \approx$
0.000294959

x = 1.0

y = sin(x)
$\hat{y} = x = 1.0$
$\Delta y = |y - \hat{y}|$
Błąd progresywny:
$\Delta y = |\sin(x) - x + \frac{x^3}{6}| = |\sin(1.0) - 1.0 + \frac{1.0^3}{6}| \approx 0.008137651$
Błąd wsteczny:
$\Delta x = |\arcsin\left(x - \frac{x^3}{6}\right) - x| = |\arcsin\left(1.0 - \frac{1.0^3}{6}\right) - 1.0| \approx$
0.014889217

```python
x = np.array([0.1, 0.5, 1.0])
y =  np.array([0.1, 0.5, 1.0])
np.set_printoptions(suppress=True)
np.set_printoptions(precision=9)
```

```python
def forward(n, f):
    return np.abs(np.sin(n) - f)
```

```python
forward(x, x)
```

```
array([0.000166583, 0.020574461, 0.158529015])
```

```python
forward(y, y - y**3/6.0)
```

```
array([0.000000083, 0.000258872, 0.008137651])
```

```python
def backward(n, f):
    return np.abs(np.arcsin(f) - n)
```

```python
backward(x, x)
```

```
array([0.000167421, 0.023598776, 0.570796327])
```

```python
backward(y, y - y**3/6.0)
```

```
array([0.000000084, 0.000294959, 0.014889217])
```

2.3.3 Wnioski

Można zauważyć, że przy użyciu rozszerzenia funkcji sinus z szeregu Taylora z większą ilością członów otrzymano bardziej dokładne wyniki.

2.4 Znormalizowany system zmiennoprzecinkowy

2.4.1 Poziom UFL

Poziom UFL to najmniejsza liczba dodatnia, jaka może być zapisana w danym systemie.
Mantysa takiej liczby musi być równa 1, a wykładnik możliwie jak najmniejszy.
Dla systemu z podstawą β, precyzją p i minimalnym wykładnikiem L, UFL można obliczyć jako:

$$\text{UFL} = \beta^L$$

Podstawiając wartości dla β = 10 i L = -98 otrzymujemy:

UFL = 10^{-98}

2.4.2 Wynik operacji x - y

x = 6,87 * 10^{-97}
y = 6,81 * 10^{-97}
x – y = (6,87 - 6,81) * 10^{-97} = 6 * 10^{-99} < UFL

Wniosek
Należy zauważyć, że z powodu minimalnego wykładnika L = -98 wynik tej operacji w tym systemie będzie wynosił 0.

3. Bibliografia

https://en.wikipedia.org/wiki/Machine_epsilon
https://pl.wikipedia.org/wiki/Wska%C5%BAnik_uwarunkowania
IEEE 754 – Wikipedia, wolna encyklopedia