# BŁĘDY OBLICZEŃ NUMERYCZNYCH

- **błędy zaokrągleń**

  **skończona liczba cyfr (bitów) w reprezentacji numerycznej**

- **błędy obcięcia**

  **rozwinięcia w szeregi i procesy iteracyjne -**

  **w praktyce muszą być skończone**

- **błędy metody**

  **przybliżone rozwiązania problemu**

**poza tym:**

*** błędy modelu**

*** błędy danych wejściowych**

*** pomyłki**

## Zaokrąglenia

1. **nieskończone reprezentacje dziesiętne**
2. **nieskończone reprezentacje binarne skończonych reprezentacji dziesiętnych, np.:**

$(\ 0.1\ )_{10} = (\ 0.0001100110011001100110011001100\ ...\ )_2$

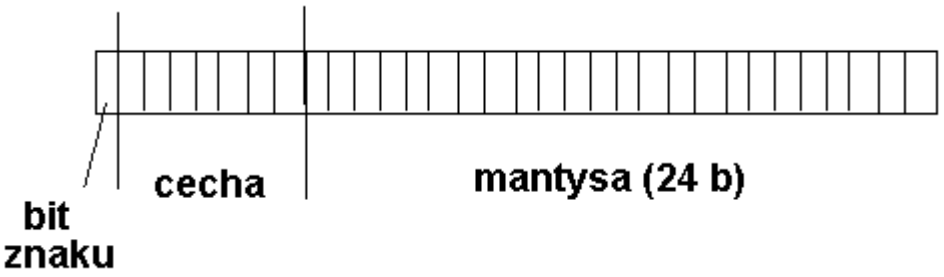
**0.09999999 ...**

$(0.000110011)_2 \rightarrow 0.099609375$

**przykład:**

## zmiennoprzecinkowa reprezentacja w IBM 370
**(najlepsza maszyna obliczeniowa w latach 70-tych)**

## 32-bitowe słowo




## podstawa systemu liczenia - 16

**rozpatrzmy liczbę, której reprezentacja ma postać**

**0 1000010 10110011000001000000000**

## wykładnik podstawy: $[ (1000010)_2 = 66 ] – 64$

**(cecha bez znaku – odejmowanie 64 zapewnia zakres wartości cechy:**

**od -64 do +63**

## wartość (uwzgl. postać znormalizowaną)

$$L = m*p^C$$

$$[(1/2)^1 + (1/2)^3 + (1/2)^4 + (1/2)^7 + (1/2)^8 + (1/2)^{14}]*16^{(66-64)} = 179.015625$$

## minimalnie większa wartość różniąca się 1 bitem

**0 1000010 10110011000001000000001**

**179.015609741210937 5**

## minimalnie mniejsza wartość różniąca się 1 bitem:

**0 1000010 10110011000000111111111**

**179.015640258789062 5**

**porównajmy:**

**179.015609741210937 5**

**179.015625**

**179.015640258789062 5**

**1) oznacza to, że reprezentacja**

**0 1000010 10110011000001000000000**

**odpowiada**
**nieskończonej „ilości” liczb rzeczywistych z połowy powyższego przedziału**

**2) maksymalnie 7 znaczących cyfr 10-tnych**

**przy 24 znaczących bitach reprezent. maszynowej**

**3) największa możliwa liczba**

**0 1111111 11111111111111111111111 $\approx 16^{63} \approx 10^{76}$**

**najmniejsza**

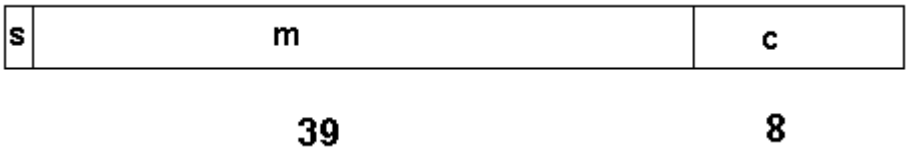
**0 0000000 00010000000000000000000 $= 16^{-65} \approx 10^{-78}$**

**(najmniejsza możliwa mantysa znormalizowana = 1/16 = $16^{-1}$ ,**
**najmniejsza możliwa cecha)**

**rozważmy bardziej przyziemny przykład:**

**reprezentacja REAL w PASCALU (PC – słowo jednobajtowe)**

**(tu arytmetyka jest dwójkowa )**




**cecha bez znaku: wartość wykładnika = c-129 [binarnie] gdy c>0**

**np.: c = 10000000 = $2^7$ = 128 => wartość wykładnika = -1**

**c = 00011001 = 25 => wartość wykładnika = -104**

**...**

**obliczenie wartości:**

- **jeśli c = 00000000 => L = 0**

  **(oznaka najmniejszego możliwego wykładnika = -129, a zatem najmniejsza możliwa liczba, co do wartości bezwzględnej = 0 )**

- **jeśli 0<c<256 => $L = (-1)^s \cdot 2^{(c-129)} \cdot 1.m$**

**(zauważmy, że tu zawsze dodawane jest 1 do mantysy)**

**najmniejsza możliwa liczba ujemna, różna od zera ma zatem postać:**

**100000000000000000000000000000000000000000000001**

**i wynosi $-2^{-128}$ ~ $-2.9*10^{-39}$ (sprawdź kalkulatorem)**

**a dodatnia**

**000000000000000000000000000000000000000000000001**

**ale w PAO będzie wyglądała tak:**

**0000000100000000000000000000000000000000000000**

**...PROGRAM REAL_MIN...(przesuwa 1 co bajt)**

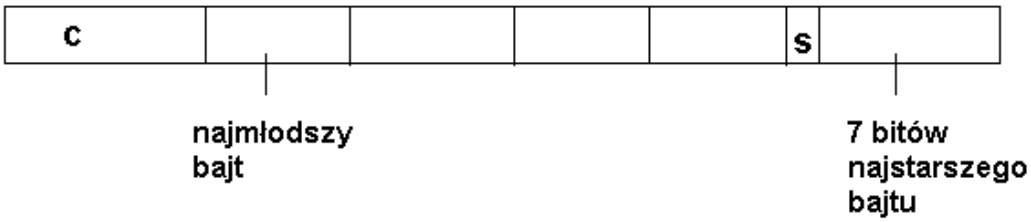
**a największa ujemna**

**1111111111111111111111111111111111111111111111**

**i wynosi ~ $-2*2^{(255-129)} = -2^{127}$ ~ $-1.7 * 10^{38}$**

**w PAO, reprezentacja wygląda inaczej gdyż**

**z pamięci kolejne BAJTY sczytywane są sekwencyjnie (od lewej) i „ustawiane” w rejestrze „poprawnie” – od prawej ... zatem:**




**program REAL_BIN – reprezentacja REAL w PAO (..dla +/- 0.5)**

**program MAX_REAL – znajdowanie bitu znaku**

## utrata dokładności

- **operacja dodawania wymaga wyrównania cech**

**(wykonajmy dodawanie na cyfrach, a nie na bitach i załóżmy że mamy tylko 7 dostępnych cyfr na mantysę)**

**$0.1460071 \times 10^{3}$ ---> $0.0000015 \times 10^{8}$**

**+ $0.2389034 \times 10^{8}$**

**a próbując dodać np. $0.1460071 \times 10^{-1}$ do $0.2389034 \times 10^{8}$**

**w wyniku otrzymamy $0.2389034 \times 10^{8}$**

a co jeśli do **$0.2389034 \times 10^{8}$** mielibyśmy dodać **100 milionów** takich **$0.1460071 \times 10^{-1}$** liczb?

dodając je sukcesywnie, wciąż mielibyśmy w wyniku **$0.2389034 \times 10^{8}$**

ale dodając wpierw do siebie 100 milionów ($10^{8}$) razy **$0.1460071 \times 10^{-1}$**

dostaniemy **$0.1460071 \times 10^{7}$**

a następnie dodając to do **$0.2389034 \times 10^{8}$** dostaniemy **$0.2535041 \times 10^{8}$**

- **odejmowanie liczb bliskich (a - b) * c**

**a = 0.2356073 x $10^6$**

**b = 0.2356102 x $10^6$**

**- 0.2900000 x $10^1$ (bo zawsze postać znormalizowana)**

**c = 0.1112050 x $10^1$**

**wynik**

**0.3224945 x $10^1$**

**|** *śmieci* **|**

**mówimy, że *p** jest zmiennoprzecinkowym przybliżeniem do *p* z dokładnością do *t* cyfr znaczących jeśli *t* jest największą nieujemną liczbą całkowitą taką, że**

$$\frac{|p - p^*|}{|p|} < 5 \times 10^{-t}$$

**Przykład**

**rozwiązanie równania**

$$\frac{d^2 y(x)}{dx^2} = c^2 y(x),$$

*C* **- rzeczywiste**

**np. równanie Schrödingera dla cząstki wewnątrz bariery**

$$y(x) = A\, e^{-CX} + B\, e^{CX}$$

**żądamy rozwiązania, które** $y(0)=1, \quad y'(0)= -C$

⇨ $y(x) = e^{-CX}$

*jeśli chcemy numerycznie znaleźć to rozwiązanie np. w przedziale [0,P],*
*czyli dla wielu arg. x, to musimy nadać wartość szykanego rozwiązania w x=0*
*a następnie „znajdować" numeryczne y(x) w kolejnych „węzłach" dyskretnej siatki zmiennej x w przedziale [0,P]; czyli -*
*zadajemy 1 oraz C w x=0 i numerycznie szukamy wartości Y dla x>0*
**rozwiązując numerycznie na dyskretnej siatce węzłów:**

*1 (jedynka)* **oraz** *C*

**są określone (zaokrąglone) z numeryczną dokładnością** $\varepsilon$

⇨ $y_{\text{numeryczne}} \approx e^{-CX} + \varepsilon e^{CX}$

już w X=0 będzie zawierało domieszkę rozwiązania

$e^{cx}$ (z wagą rzędu $\varepsilon$ - rzędu dokładności numerycznej )

**dla dużych i dodatnich** $X$ **"eksplodujące" rozwiązanie** $e^{CX}$ **zdominuje**

**numeryczne rozwiązanie** $y_{num}$

## błędy obcięcia

**pewne wielkości wyliczane są zawsze**
**z nieskończonych sum, szeregów czy iteracji**

**np:**

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots$$

**przykład unikania utraty dokładności**

$$f(x) = x - \sin(x)$$

**dla x ≈ 0 x ≈ sin(x) (bliskie wartości)**

**==>** $$f(x) = \frac{x^3}{3!} - \frac{x^5}{5!} + \frac{x^7}{7!} - \ldots$$

.....

**numeryczne sumowanie szeregu $1/(n^3)$**

**- w przód i w tył**

**...program... suma_szeregu**

**METODY N UMERYCZNE =**

**sposoby przybliżonego rozwiązywania zagadnień**

- **dyskretyzacja**
- **aproksymacja**
- **propagacja błędów i numeryczna niestabilność**

**np. przykład (prawo rozpadu promieniotwórczego)**

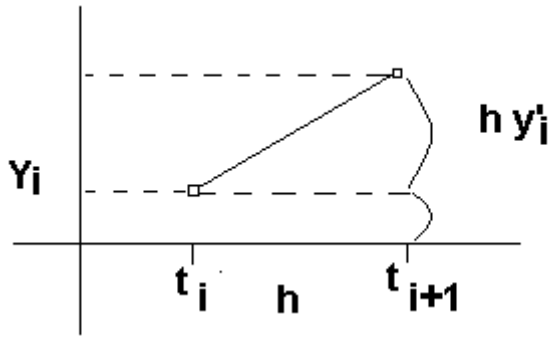
**rozwiązanie równania** $\frac{dY(t)}{dt} = -\lambda Y(t)$

**metodą Eulera (poznamy później)**

*rozwiązaniem ścisłym jest* $y(t) = \exp(-\lambda t)$

*(dla* $\lambda > 0$ *wartości* $Y(t)$ *powinny monotonicznie maleć w czasie)*

**na siatce punktów** $\{ t_i \}, \quad t_{i+1} - t_i = h$




$$Y_{i+1} = Y_i + h\, Y_i' = (1 - \lambda h) Y_i$$

**metoda jest niestabilna**

**jeśli**

$$h > 2 / \lambda$$

**...program EULER_GR ....**

**problemy źle uwarunkowane**

**(przy zadanej dokładności numerycznej)**

**np.:**

**układ równań**

**6.025 x + 109.774 y = 4.12**

**14.054 x + 256.125 y = 10.50**

**ma rozwiązanie:**

**x = -250.15  y = 13.77  ;**

**jeśli dysponujemy dokładnością do 2 miejsc po kropce i wyniki pośrednich operacji zaokrąglamy do 2 miejsc po kropce, to rozwiązaniem jest**

**x = - 44.33  y = 2.47**

**powód:**

**równania ”prawie” zależne (niezależne na granicy dokładności)**