

# Изчислителни методи и обработка на данни

ст. ас. д-р Стоян Писов

# Какво е изчислителна физика?

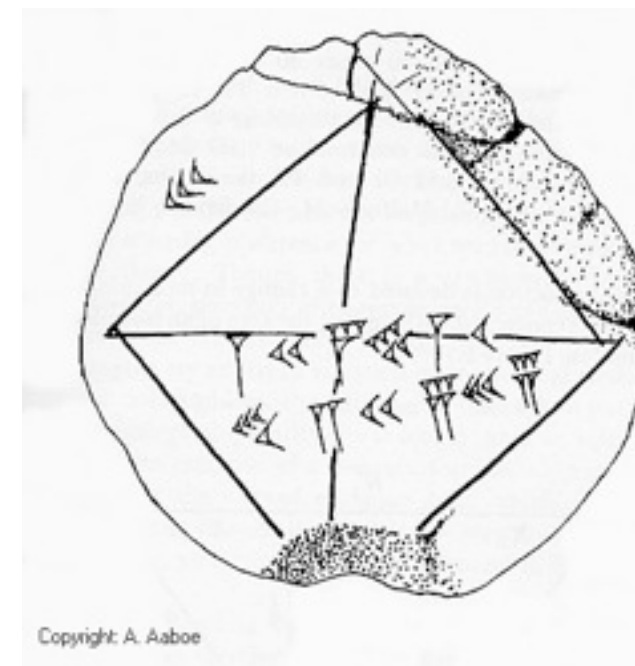
- Дял от физиката.
- Доколкото всеки физичен процес се описва с подходящ математичен модел, може да се каже че изчислителната физика е съвкупност от числени методи необходими за решаването на конкретна задача.
- Числените методи изучават алгоритми които използват числено представяне при описанието на проблемни задачи от численият анализ.

# История на числените

## МЕТОДИ

- Древен Вавилон (1800 г. пр. х) - глинена плочка с указания за пресмятане на  $\sqrt{2}$
- Древна Гърция (190 г. пр. хр.) - Линейна интерполация при определяне междинни стойности в таблици
- Епохата на Нютон и Лагранж 17 - 18 век - числена процедура за намиране на корени, интерполиране с полиноми:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$



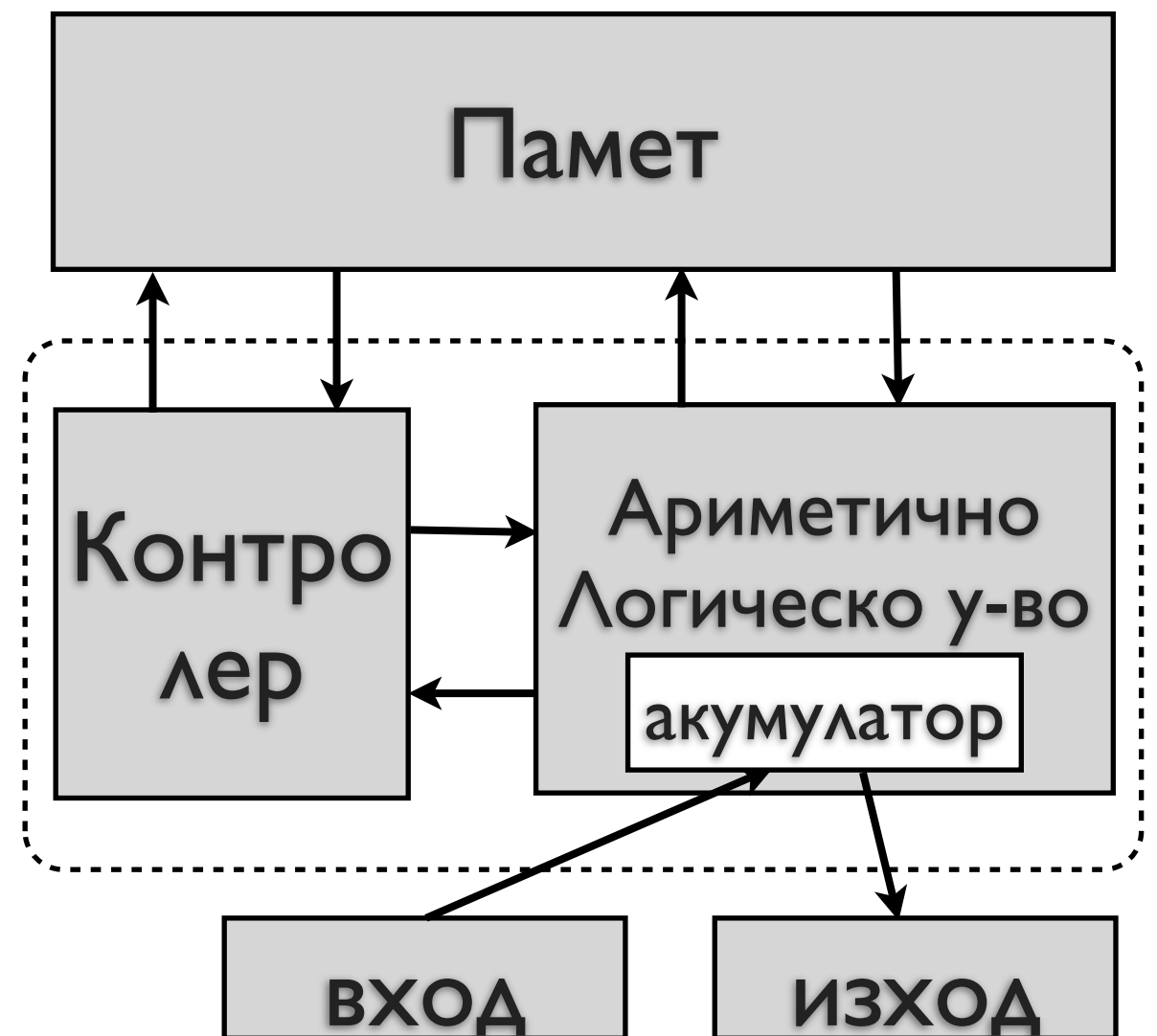
# История на числените методи

- Механичният калкулатор (1822 г.), Чарлз Бабидж създава Диференциална машина - табулира полиномиални функции
- Машина на Тюринг (1936 г.) - Теоретично устройство, принципното действие на което се използва в съвременните компютри.
- Електронно Изчислителни Машини (1941 г.-)
  - Аналогови - директно решаване на числената задача
  - Двоични - програмируеми, общо предназначение

# Съвременен компютър

## Архитектура на фон Нойман

- Позволява изпълнението на различни задачи
- Компютърната програма се съхранява в паметта



# Двоично представяне на числата

- Цели числа ( $\mathbb{Z}$ ) - Integer (int, integer)
- Реални числа ( $\mathbb{R}$ ) - Real (float, double)
- Комплексни числа ( $\mathbb{C}$ )- Complex (complex)

# Цели числа

- Двоично представяне - 1В, 2В, 4В, 8В и т.н.
- Диапазона зависи от размера на представянето.
- Съществуват различни представяния:
  - знак-и-големина
  - обратен код
  - допълнителен код
  - други

# Цели числа представяне

## знак-и-големина

- най-старшият бит определя знака
- има две нули +0 и -0

двоично представяне	знак	без знак
00000000	+0	0
00000001	1	1
...	...	...
01111111	127	127
10000000	-0	128
10000001	-1	129
...	...	...
11111111	-127	255



# Цели числа представяне

обратен код

- отрицателните числа се получават като се приложи маска побитово НЕ

- има две нули  $+0$  и  $-0$

двоично представяне	знак	без знак
00000000	+0	0
00000001	1	1
...	...	...
01111110	126	126
01111111	127	127
10000000	-127	128
10000001	-126	129
...	...	...
11111110	-1	254
11111111	-0	255

# Цели числа представяне

допълнителен код

- отрицателните числа се получават като се приложи маска побитово НЕ и събиране с 1
- има една 0
- лесно събиране

двоично представяне	знак	без знак
00000000	+0	0
00000001	1	1
...	...	...
01111110	126	126
01111111	127	127
10000000	-128	128
10000001	-127	129
...	...	...
11111110	-2	254
11111111	-1	255

# Реални числа

## представяне

- представяне с фиксирана запетая
  - броят на значещите цифри е постоянен
  - 12345.67, 123.45, 1.23, 0.12, 0.01
- представяне с плаваща запетая
  - дробната запетая може да се мести (плава)
  - 1.234567, 123456.7, 0.00001234567, 1234567000000000

# Числа с плаваща запетая

## IEEE 754 стандарт

- Общо представяне:

$$r = \text{significant digits} \times \text{base}^{\text{exponent}}$$

или  $r = s \times b^e$ ,  $s$  - значещ разред (significant),  $b$  - основа (base),  $e$  - експонента (exponent)

- Двоично представяне:

$$\left( 1 + \sum_{n=1}^{p-1} bit_n \times 2^{-n} \right) \times 2^e$$

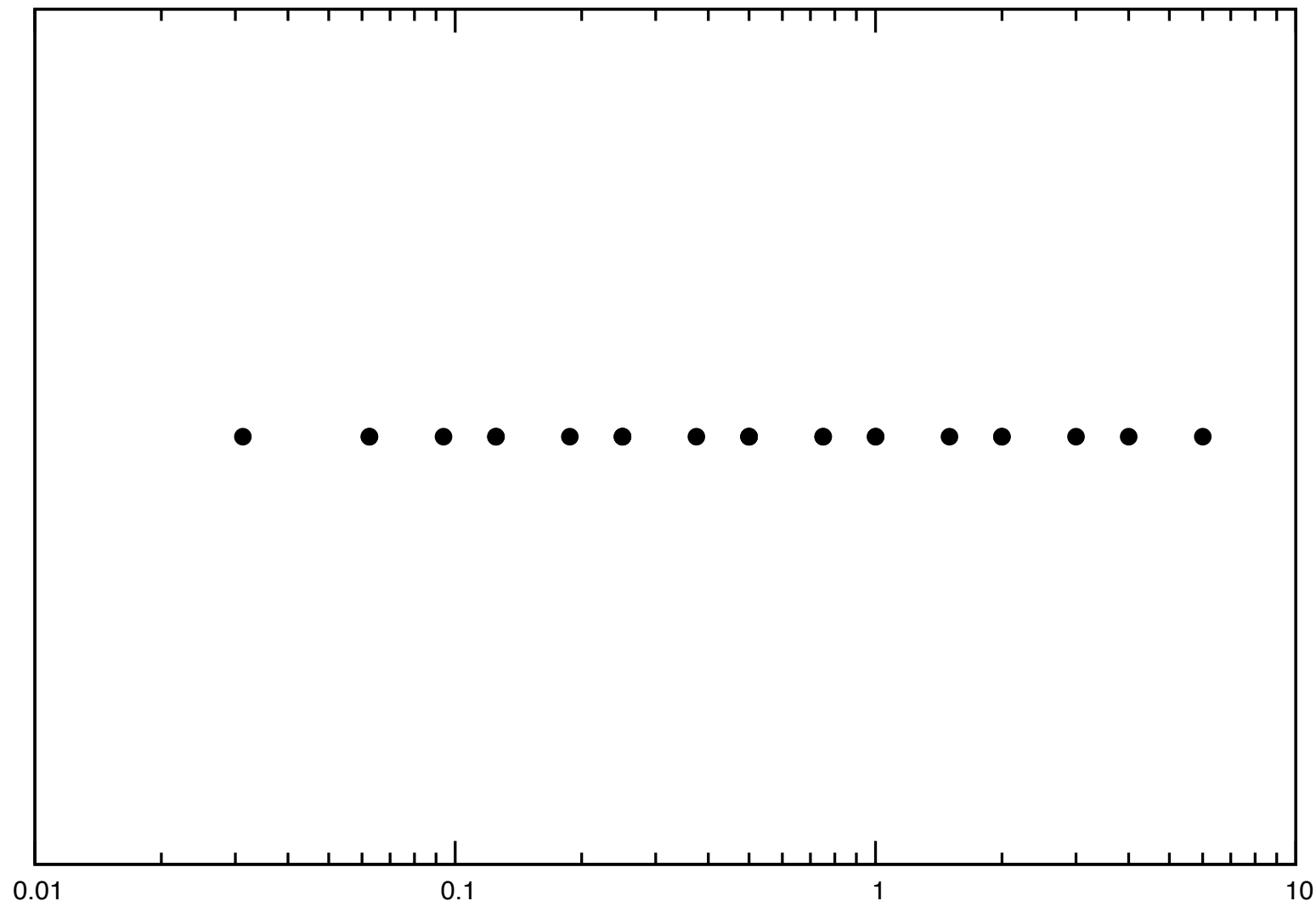
# Числа с плаваща запетая IEEE 754 стандарт

Пример: представяне на числото с 24  
битова точност 11001001 00001111  
11011011:

$$3.1415928 = 2 \times 1.5707964 = (1 + 1 \times 2^{-1} + 0 \times 2^{-2} + 0 \times 2^{-4} + \dots + 1 \times 2^{-22} + 1 \times 2^{-23}) \times 2^1$$

# Числа с плаваща запетая IEEE 754 стандарт

Пример: Множеството от 5 bits числа с  
 $p=4$ ,  $b=2$ ,  $e=2$ .



# Числа с плаваща запетая

## IEEE 754 стандарт

- Единична точност (Single precision) - тип float в C/C++ или real във Fortran, 32 bits ( $s = 24$  bits,  $b = 2$ ,  $e = 8$  bits)
- Двойна точност (Double precision) - тип double в C/C++ или real\*8 във Fortran, 64 bits ( $s = 53$  bits,  $b = 2$ ,  $e = 11$  bits)
- Разширена точност (Extended precision) - тип long double в C/C++, 80 bits
- Половин точност (Half), 16 bits

# Числа с плаваща запетая

## IEEE 754 стандарт

Специални стойности:

- Нулата има знак:  $+0$  и  $-0$
- Машинна нула - числата около 0 които са по-малки от най-малкото число с плаваща запетая в конкретното представяне
- Безкрайности - представят се като всяко друго число, обикновено резултат от препълване или делене на нула



# Числа с плаваща запетая

## IEEE 754 стандарт

Специални стойности:

- NaN - специална стойност Not a Number, резултат от невалидна операция:  $0/0$ ,  $\text{sqrt}(-1)$  или  $\infty \times 0$
- Машинен епсилон - число (оценка) което характеризира точността на представянето с плаваща запетая. В зависимост от начина на закръгляне горната граница се дефинира като:

$$E_{mach} \approx B^{1-P}$$

# Пример събиране:

Нека разгледаме събирането на две числа със седем значещи цифри:

$$123456.7 = 1.234567 * 10^5$$

$$101.7654 = 1.017654 * 10^2 = 0.001017654 * 10^5$$

Сумирането може да се извърши като се приведат двете числа в един и същи порядък, т.е. една и съща експонента:

$$\begin{aligned} 123456.7 + 101.7654 &= (1.234567 * 10^5) + (1.017654 * 10^2) \\ &= (1.234567 * 10^5) + (0.001017654 * 10^5) \\ &= (1.234567 + 0.001017654) * 10^5 \\ &= 1.235584654 * 10^5 = 123558.5 \end{aligned}$$

# Производителност FLOPS

FLOPS - **F**loating point **O**perations per **S**econd (Операции с плаваща запетая за секунда)

Типични примери:

- Домашен компютър с процесор Intel Core 2 Duo работещ на тактова честота 2 GHz има теоретична производителност 16 Gflops = (2 cores) × (4 flop) × (2 GHz)
- Изчислителния клъстер Physon - 1856 Gflops
- BladeCenter HS21 Cluster (номер 500 в класацията top500 за 06/2010) - 54.79 Tflops
- Jaguar XT5-NE (номер 1 в класацията top500 за 06/2010) - 2331.00 Tflops

# Задача:

## Определяне машинният епсилон

Псевдо код:

```
Eps = 1
```

```
repeat
```

```
    Eps = Eps / 2
```

```
until (1 + Eps/2) != 1
```

```
print Eps
```

# Исползвана литература:

1. [http://en.wikipedia.org/wiki/Numerical\\_analysis](http://en.wikipedia.org/wiki/Numerical_analysis)
2. [http://en.wikipedia.org/wiki/Charles\\_Babbage](http://en.wikipedia.org/wiki/Charles_Babbage)
3. [http://en.wikipedia.org/wiki/Turing\\_machine](http://en.wikipedia.org/wiki/Turing_machine)
4. [http://en.wikipedia.org/wiki/  
Von\\_Neumann\\_architecture](http://en.wikipedia.org/wiki/Von_Neumann_architecture)
5. [http://en.wikipedia.org/wiki/Floating\\_point](http://en.wikipedia.org/wiki/Floating_point)