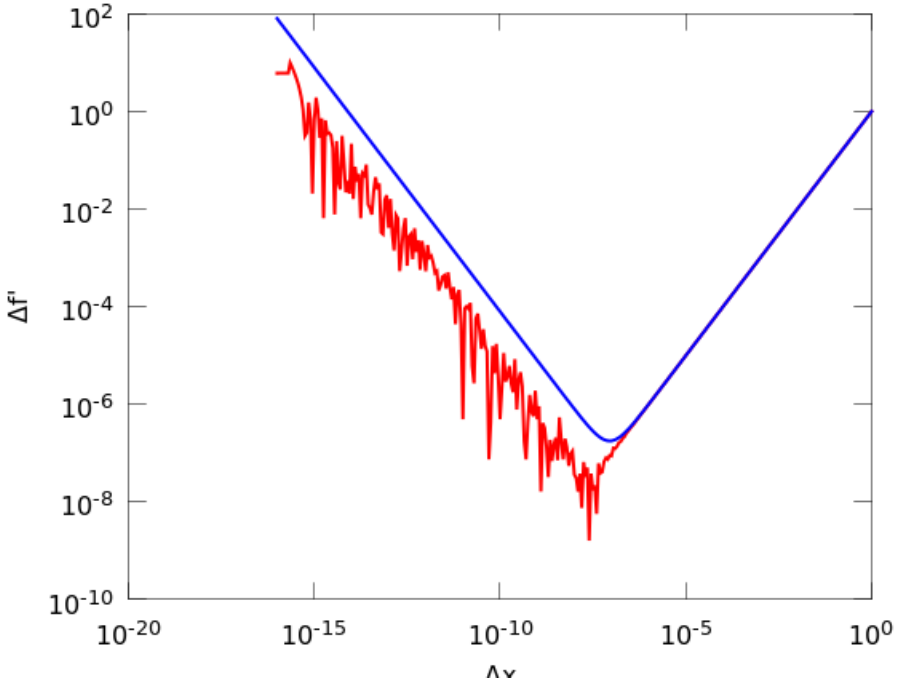
Лекция 11. Двоичная арифметика и проблема точности вычислений



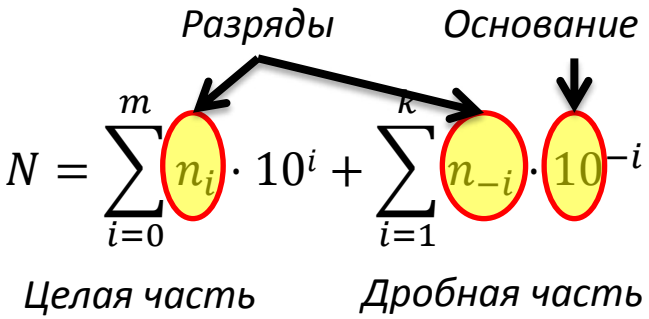
Краткое содержание

1. Двоичная система счисления: целые числа и дроби
2. Восьмеричная система счисления
3. Числа с фиксированной и плавающей запятой
4. Формат IEEE-754, машинный эпсилон
5. Численное дифференцирование

Позиционная система счисления

В позиционной системе счисления значение цифры (знака) зависит от разряда, в котором она находится

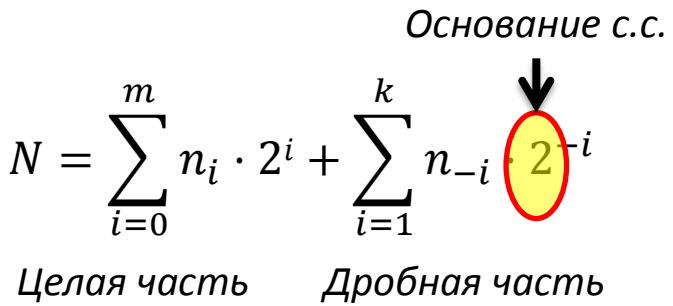
Десятичная система счисления



Примеры

1. $123_{10} = 1 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0$
2. $0.25_{10} = 2 \cdot 10^{-1} + 5 \cdot 10^{-2}$
3. $2.75_{10} = 2 \cdot 10^0 + 7 \cdot 10^{-1} + 5 \cdot 10^{-2}$

Двоичная система счисления



Бит – двоичный разряд

Примеры

1. $1010_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 8_{10} + 2_{10} = 10_{10}$
2. $0.101_2 = 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} = \frac{1}{2} + \frac{1}{8} = \frac{5}{8} = 0.625_{10}$
3. $10.11_2 = 1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} = 2 + \frac{1}{2} + \frac{1}{4} = 2\frac{3}{4} = 2.75_{10}$

Перевод в двоичную систему счисления: целые

Алгоритм

1. ***Разделить*** число на 2 с остатком
2. ***Остаток*** – двоичный разряд числа, ***записать его слева*** от уже выписанных разрядов
3. Если целая часть – 0, то закончить, иначе перейти к шагу 1

В десятичной системе

```
1023 / 10 = 102 ост 3
102  / 10 = 10  ост 2
10   / 10 = 1   ост 0
1    / 10 = 0   ост 1
```

Пример 1: 42_{10}

```
42 / 2 = 21 ост 0
21 / 2 = 10 ост 1
10 / 2 = 5  ост 0
5  / 2 = 2  ост 1
2  / 2 = 1  ост 0
1  / 2 = 0  ост 1
```

Ответ: 101010_2

Проверка: $2^5+2^3+2^1 =$

$= 32 + 8 + 2 = 42$

Пример 2: 16_{10}

```
16 / 2 = 8 ост 0
8  / 2 = 4 ост 0
4  / 2 = 2 ост 0
2  / 2 = 1 ост 0
1  / 2 = 0 ост 1
```

Ответ: 10000_2

Перевод в двоичную систему счисления: дроби

Алгоритм (для чисел <1)

1. ***Умножить*** число на 2
2. ***Целая часть*** – двоичный разряд числа, ***записать его справа*** от уже выписанных разрядов
3. Взять дробную часть. Если она – 0 (или достигнута желаемая точность), то закончить, иначе перейти к шагу 1

В десятичной системе

```
0.675 * 10 = 6.75
0.75 * 10  = 7.5
0.5 * 10   = 5.0
```
Ответ: 0.675_{10}

```
1/6 * 10 = 10/6 = 1+2/3
2/3 * 10 = 20/3 = 6+2/3
2/3 * 10 = и.т.д.
```
Ответ: $0.1(6)_{10}$

Пример 1: 0.75_{10}

```
0.75 * 2 = 1.5
0.5 * 2  = 1.0
```
Ответ: 0.11_2

Пример 2: 0.3125_{10}

```
0.3125 * 2 = 0.625
0.625 * 2  = 1.25
0.25 * 2   = 0.5
0.5 * 2    = 1.0
```
Ответ: 0.0101_2

Пример 3: 0.2_{10}

```
0.2 * 2 = 0.4
0.4 * 2 = 0.8
0.8 * 2 = 1.6
0.6 * 2 = 1.2
0.2 * 2 = и.т.д.
```
Ответ: $0.(0011)_2$

Внимание! На компьютере лучше использовать побитовые операции

Шестнадцатеричная и восьмеричная системы счисления

Системы счисления с основанием 8 и 16 соответственно. Восьмеричная цифра содержит 3 бита, шестнадцатеричная – 4.
Внимание! При переводе чисел между двоичной, восьмеричной и шестнадцатеричной системами счисления не пользуйтесь десятичной в качестве промежуточной! Двоичная ощутимо удобнее в этом случае!

Шестнадцатеричные цифры

Цифра	DEC	BIN	Цифра	DEC	BIN
0	0	0000	8	8	1000
1	1	0001	9	9	1001
2	2	0010	A	10	1010
3	3	0011	B	11	1011
4	4	0100	C	12	1100
5	5	0101	D	13	1101
6	6	0110	E	14	1110
7	7	0111	F	15	1111

Числа с плавающей запятой

Числа с фиксированной запятой

1. Фиксированное количество разрядов
2. Фиксированное положение запятой (разделителя целой и дробной части)

Числа с плавающей запятой

1. Деление числа на мантиссу и порядок
2. Фиксированное количество разрядов в мантиссе

В случае чисел с фиксированной запятой постоянна абсолютная погрешность

Примеры

```
  1.23456
123.45600
  0.01234
  0.00012
```

В случае чисел с плавающей запятой постоянна относительная погрешность

Примеры

$(1.23456)*10^{0}$

$(1.23456)*10^{2}$

$(1.23456)*10^{-2}$

$(1.23456)*10^{-4}$

Числа с плавающей запятой: формат IEEE-754

```
Тип double (8 байт)
+-----+--+--+--+--+--+--+--+
|seeeE|EM|MM|MM|MM|MM|MM|MM|
+-----+--+--+--+--+--+--+--+
Всего – 64 бита (8 байт)
Знак – 1 бит
Порядок – 11 бит
Мантисса – 52 бита
```

$$N = (-1)^s \cdot \left(1 + \sum_{i=1}^{k} n_i 2^{-i}\right) 2^M$$

Особенности формата

1. Знак: 0 – «плюс», 1 – «минус»
2. Порядок: хранится в виде суммы с числом 1023 (2^{10}-1)
3. Мантисса: старший разряд – всегда единица и опускается
4. 0 кодируется особым образом

Отображение в Octave

```
>> format hex;
```

После этого все числа будут выводиться в шестнадцатеричном виде и формате IEEE-754

Пример 1: 2_{10}

```
>> format hex; 2
ans = 4000000000000000
Разбор:
40 00 00 00 00 00 00 00
400₁₆ = 0100 0000 0000
Знак – «+»
Порядок – 0100 0000 0000₂ = 1024
= 1023 + 1 => 1
Мантисса – [1]0000 0000 и т.д.
```

$$N = (2^0) \cdot 2^1 = 2$$

Числа с плавающей запятой: формат IEEE-754

Пример 2: -21.75$_{10}$

```
>> format hex; -21.75
>> -21.75
ans = c035c00000000000
C0 35 C0 00 00 00 00 00
```

$C03_{16} = 1100\ 0000\ 0011_2$

Знак – «-»

Порядок – $100\ 0000\ 0011_2$ =1027 = 1023 + 4 => 4

Мантисса – $5C_{16}$=[1]0101 1100…

$$N = -\left(2^0 + 2^{-2} + 2^{-4} + 2^{-5} + 2^{-6}\right) \cdot 2^4 =$$
$$-\left(2^4 + 2^2 + 2^0 + 2^{-1} + 2^{-2}\right) =$$
$$-\left(16 + 4 + 1 + \frac{1}{2} + \frac{1}{4}\right) = -21\frac{3}{4}$$

Особые случаи:

```
>> format hex;
>> 0
ans = 0000000000000000
```

Все биты – 0

```
>> +Inf
ans = 7ff0000000000000
```

Биты мантиссы – 0, биты порядка - 1

```
>> -Inf
ans = fff0000000000000
```

Биты мантиссы – 0, биты порядка - 1

```
>> NaN
ans = 7ff8000000000000
```

Биты мантиссы – не все 0, биты порядка - 1

Примечание: при работе с форматом IEEE-754 с помощью прямого доступа к ячейкам памяти (например, в языках Си и Ассемблер) помните о Little Endian/Big Endian!
Little Endian – на x86 процессорах
Big Endian – на PowerPC процессорах

Числа с плавающей запятой: погрешности

В случае типа `double` стандарта IEEE-754 под мантиссу отведено 52 разряда, единица в последнем из них означает $\varepsilon = 2^{-52}$. Таким образом, относительная погрешность числа типа `double` составляет примерно ε.

ε – машинный эпсилон

Пример 1: $1 + \delta$

```
>> format hex;
>> s1 = 1 + 2^-50;
s1 = 3ff0000000000004
>> s2 = 1 + 2^-52;
s2 = 3ff0000000000001
>> s3 = 1 + 2^-53;
s3 = 3ff0000000000000
>> format short;
>> s1 - 1
ans =  8.8818e-16
>> s2 - 1
ans =  2.2204e-16
>> s3 - 1
ans = 0
```

Пример 2: сумма чисел

```
>> a = 0.1; s = 0;
>> for i=1:10000; s = s + a; end;
>> s
s =  1000.00000000016
>> a = 0.125; s = 0;
>> for i=1:10000; s = s + a; end;
>> s
s =  1250
>> s - 1250
ans = 0
```

В первом случае накапливаются ошибки округления (т.к. 1/10 – бесконечная периодическая дробь), а во втором – нет (т.к. 1/8 имеет точное представление)

Численное дифференцирование

$$f(x) = x^2; x = 3; f(x) = 9$$
$$f'(x) = 2x; x = 3; f'(x) = 6;$$

$$f'(x) \approx \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

Погрешность формулы

$$f'(x) \approx \frac{(x + \Delta x)^2 - x^2}{\Delta x} = \frac{x^2 - 2x\Delta x + (\Delta x)^2 - x^2}{\Delta x} = 2x + \Delta x$$

Погрешность округления

$$\Delta(x + \Delta x) = x\varepsilon + \Delta x\varepsilon \approx x\varepsilon; \varepsilon(x + \Delta x) = \varepsilon; \varepsilon(x^2) = \varepsilon[(x + \Delta x)^2] = 2\varepsilon;$$

$$\Delta[(x + \Delta x)^2 - x^2] = 4\varepsilon x^2; \varepsilon[(x + \Delta x)^2 - x^2] = \frac{2\varepsilon x}{\Delta x}$$

$$\varepsilon[f'(x)] = \frac{2\varepsilon x}{\Delta x} + \varepsilon \approx \frac{2\varepsilon x}{\Delta x}; \Delta[f'(x)] = \frac{4\varepsilon x^2}{\Delta x}$$

Суммарная погрешность

$$\Delta f' = \frac{4\varepsilon x^2}{\Delta x} + \Delta x; \frac{\partial \Delta f'}{\partial \Delta x} = -\frac{4\varepsilon x^2}{(\Delta x)^2} + 1 = 0 \Rightarrow \Delta x = 2x\sqrt{\varepsilon}; \Delta f' = 4x\sqrt{\varepsilon}$$

Численное дифференцирование

$$\Delta f' = \frac{4\varepsilon x^2}{\Delta x} + \Delta x; \Rightarrow \Delta x_{opt} = 2x\sqrt{\varepsilon} =; \Delta f'_{opt} = 4x\sqrt{\varepsilon}$$

```
dx = 10.^(-16:0.05:0); x = 3; % Variant 1
%dx = 2.^(-52:1:1); x = 3; % Variant 2
df = ((x + dx).^2 - x.^2) ./ dx;
Ddf = abs(df - 2*x);
Ddf_theor = 4*eps*x.^2./dx + dx;
loglog(dx,Ddf,'r-','LineWidth',2,...
  dx,Ddf_theor,'b-', 'LineWidth', 2);
xlabel('\Delta{x}'); ylabel('\Deltaf''');
```