Погрешности округлений в компьютерах

CPU может работать только с целыми числами: +, −, сравнение. Операции /, *, и др. и работа с веществ. числами реализуется при пом. вычислит. методов, содержащихся в Command. сан, "зашитых" в С-процессор и в спец. библиотеках разных трансл.

Приближ. представление и операции с веществ. числами - основной источник погрешностей вычислений.

Вычислит. методы должны строиться с наименьшими вычислениями и без накопления вычисл. погрешностей.

Числа с плавающей точкой

Рассмотрим принцип представления веществ. чисел с плав. точкой при помощи целых чисел в 10-ричной системе:

$$X = 0.463 = 4\cdot 10^{-1} + 6\cdot 10^{-2} + 3\cdot 10^{-3}$$

точность ≡ кол-во цифр

основание исчисления

$$X = 46.3 = \left(4\cdot 10^{-1} + 6\cdot 10^{-2} + 3\cdot 10^{-3}\right)\cdot 10^{2}$$

показатель

дробная часть - мантисса

В общем виде: (для любой системы исчисления)

$$x = \pm\left(d_1 z^{-1} + d_2 z^{-2} + \dots + d_n z^{-n}\right) z^{e}$$

Четыре целых числа d_i, n, e_{min}, e_{max} определяют число с плавающей точкой, целые числа d_i зависят от системы исчисления:

$$0 \le d \le z-1.$$

$z=2$: $d=0,1$
$z=10$: $d=0,1,2,\dots,9$
$z=16$: $d=0,1,2,\dots,F$

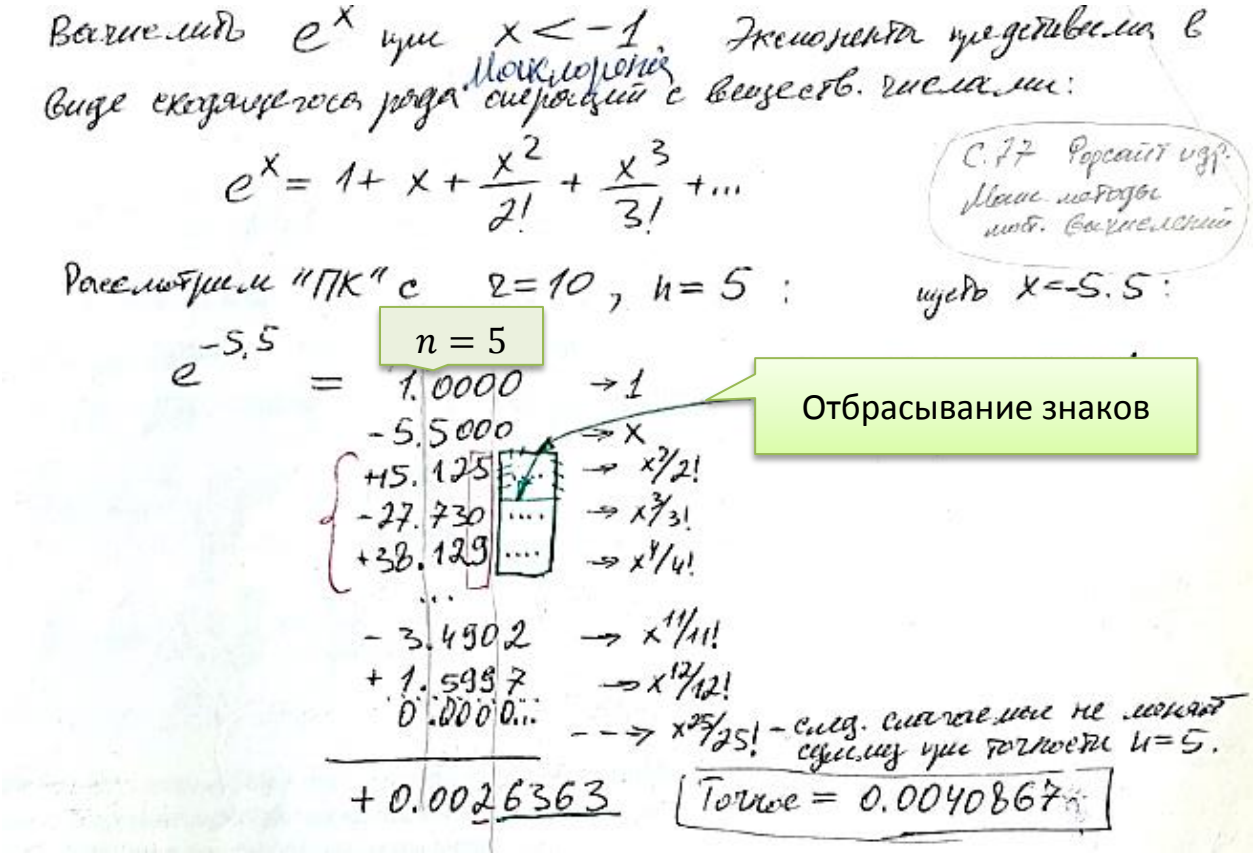
Т.к. n, e_{min}, e_{max} - ограничены $(\neq\infty) \Rightarrow$ ограниченность точности и диапазона веществ. чисел. В ПК фирмы IBM $z=16$, а (n, e) могут изменяться $\Rightarrow$ простая/двойная точность

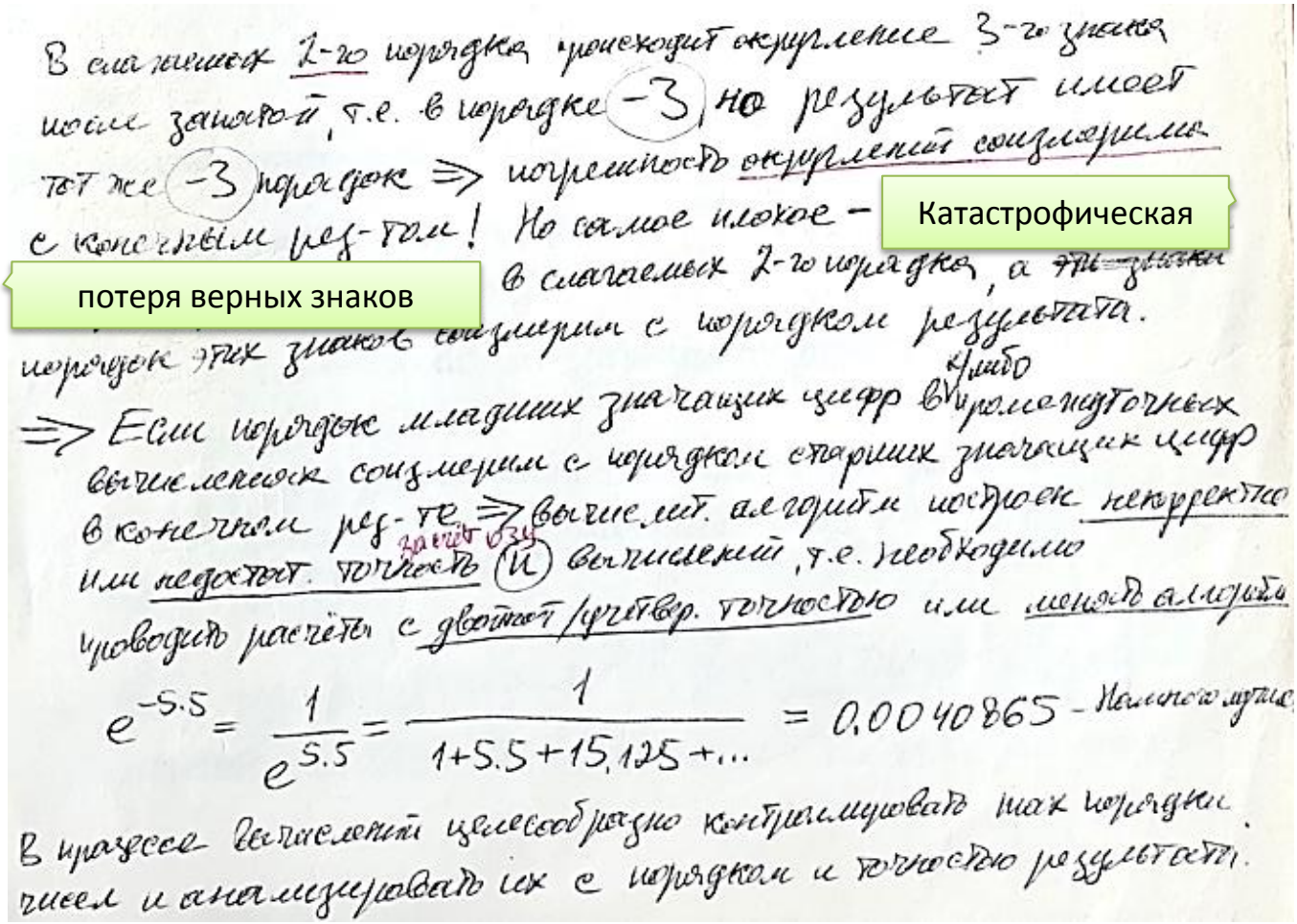
Т.о. ПК не может работать с весьм. малыми и бесконечно большими величинами $\Rightarrow$ погрешности вычислений.

C#:
float - 4 байта - 7 цифр
double - 8 байт - 15 цифр

Delphy:
real, double - 8 байт - 15 цифр
extended - 10 байт - 19 цифр

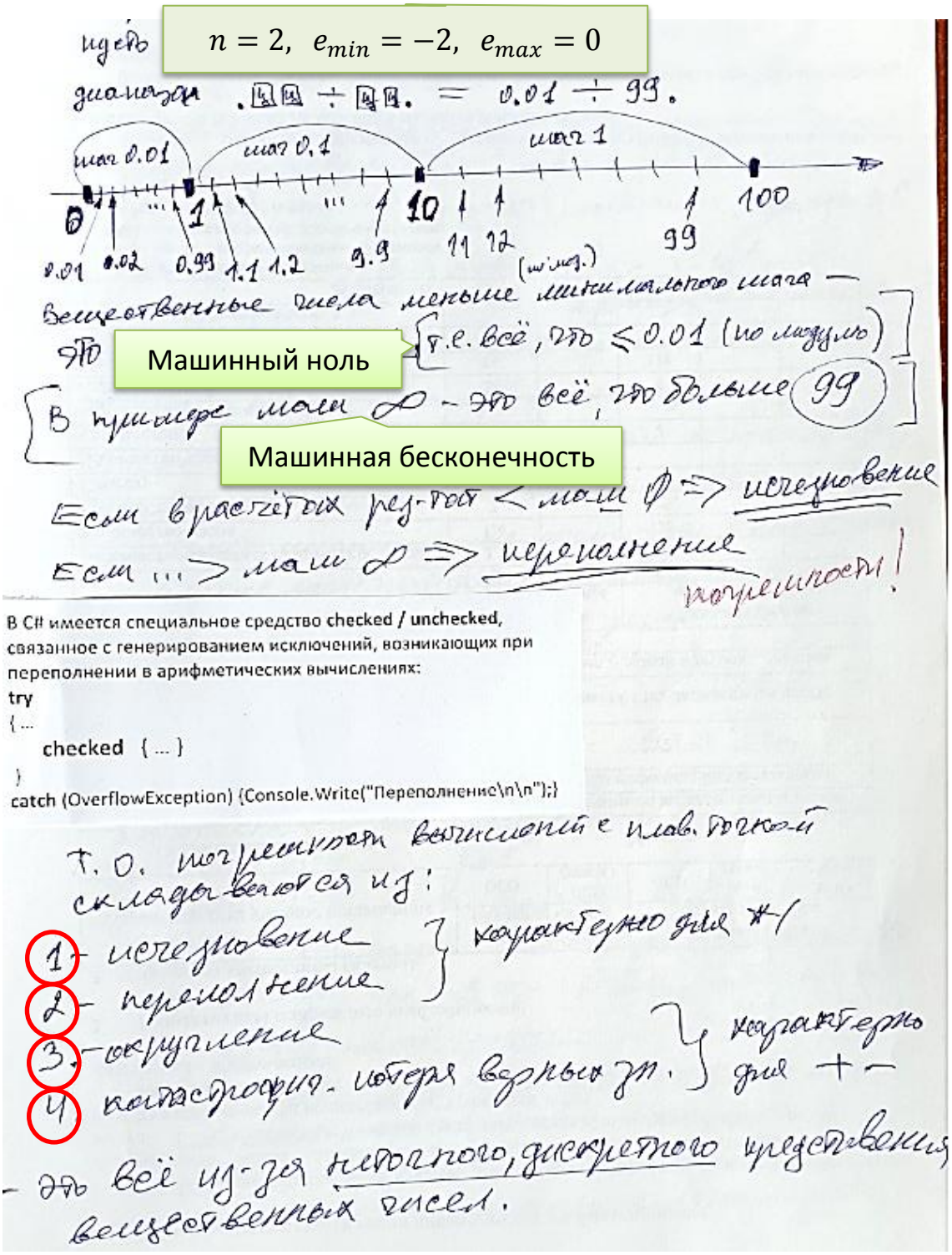
Пример погрешностей при работе с вещественными числами





Дискретность чисел с плавающей запятой (точкой)

$n = 2,\ e_{min} = -2,\ e_{max} = 0$



В C# имеется специальное средство checked / unchecked, связанное с генерированием исключений, возникающих при переполнении в арифметических вычислениях:

```
try
{...
    checked  { ... }
}
catch (OverflowException) {Console.Write("Переполнение\n\n");}
```

Т. о. погрешности вычислений с плав. точкой складываются из:

1. исчезновение } характерно для +/−
2. переполнение }
3. округление } характерно для +−
4. катастрофа. потеря верных зн. }

— это всё из-за непрерывного, дискретного представления вещественных чисел.

Нормальная форма:
мантисса $\in [0,1)$, т.е.:
$0_{,} \ldots E_{\ldots}$ здесь для
float $\varepsilon = 3{,}4E{-}38$

Нормализованная
мантисса $\in [1,10)$, т.е.
$N_{,}^{\neq 0} \ldots E_{\ldots}$ здесь $\varepsilon = 1{,}4E{-}45$

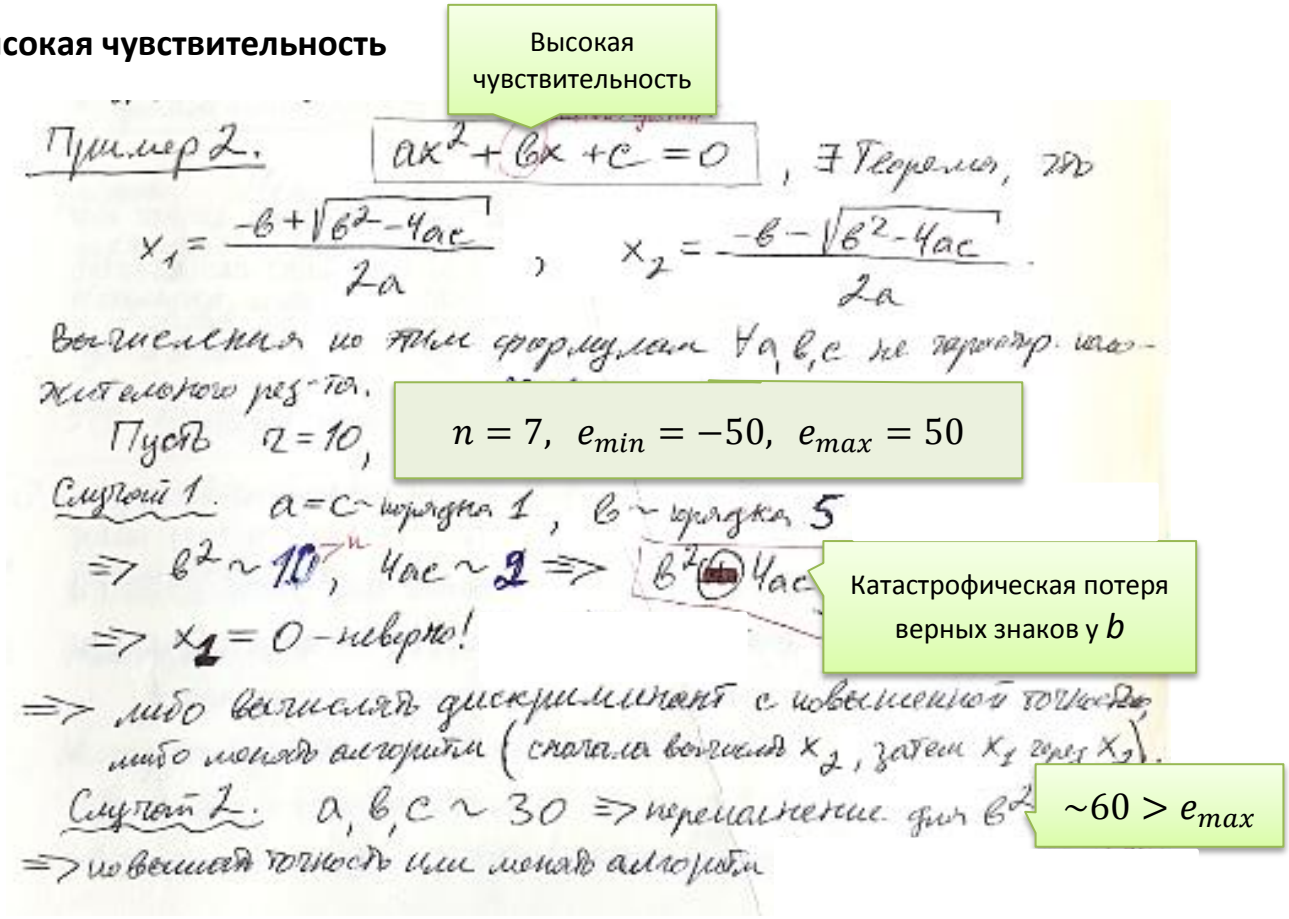
т.к. целая часть в двоичн. числе всегда "1", её не хранят и экономят место на точность

Погрешности вычислительных алгоритмов

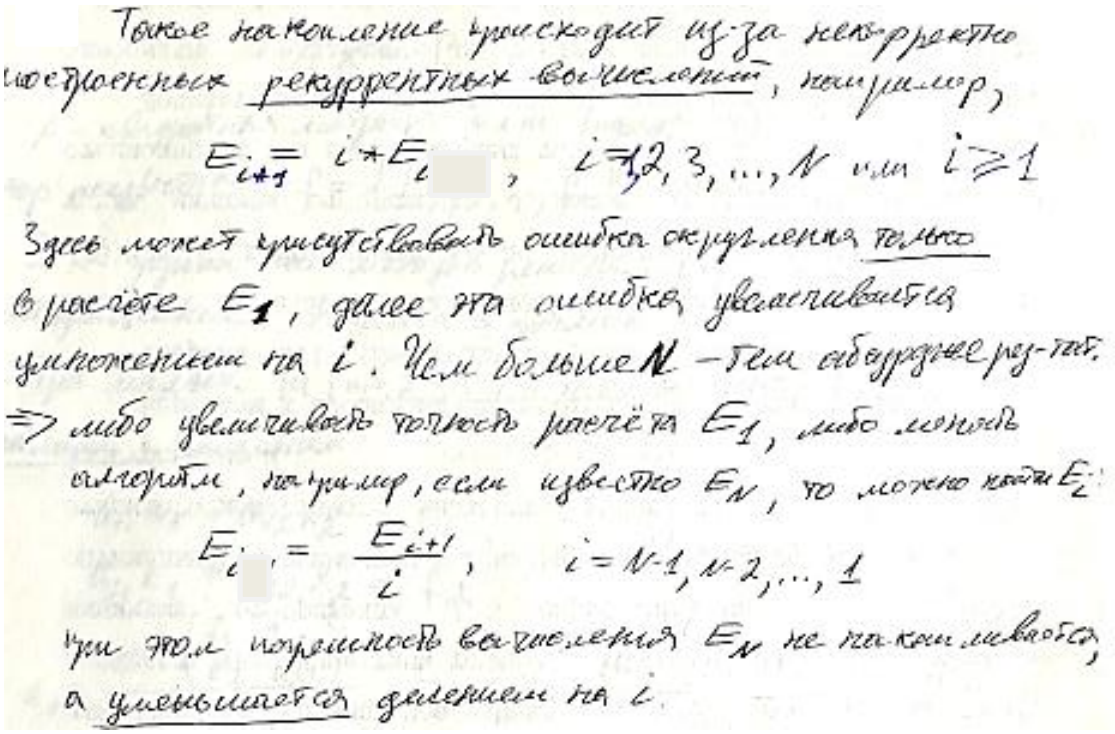
Внешне проявляются в виде абсурдного результата при формально правильной программе вычислений.

Причины:

1. Высокая чувствительность



2. Алгоритмы с накоплением погрешностей



3. Некорректность математической постановки задачи