Московский энергетический институт

Кафедра Вычислительных машин систем и сетей

КУРС ПРОБЛЕМЫ ОРГАНИЗАЦИИ ВЫЧИСЛЕНИЙ

Лекция на тему :

«Высокоточные вычисления»

Москва 2019 г.

Направления по курсу ПОВ

1. *«Высокоточные компьютерные арифметики» (д.т.н., Оцоков Ш.А)*
2. *Машинное обучение (д.т.н., проф. Дзегеленок И.И., д.т.н., Оцоков Ш.А)*
3. *Геометрическое моделирования (к.т.н., Орлов Д.А.)*
4. *Технология виртуальной реальности (к.т.н., Харитонов В.Ю)*

Паблик в соц сети: http://vk.com/club50059448

Компьютерная арифметика

Было бы ошибкой считать, что компьютерная арифметика необходима только разработчикам процессоров. Мы рассмотрим дальше примеры, как более эффективно точнее составлять расчётные программы, избегать вычислительных ошибок, свойственных арифметики с плавающей точкой.

Основные вопросы предмета компьютерная арифметика – это:

1. Разработка эффективных цифровых схем.
2. Ускорение арифметических операций и вычисление специальных функций.
3. Разработка алгоритмов быстрого выполнения арифметических операций.
4. Анализ ошибок округления,
5. Аппаратная реализация.
6. Тестирование, верификация программ

Требования к системам счисления

- «возможность представления чисел в заданном диапазоне
- однозначность представления
- простоту записи
- удобство работы человека с машиной
- трудоёмкость выполнения арифметических операций
- экономичность системы (количество элементов, необходимых для представления многоразрядных чисел)
- удобство аппаратной реализации

Экономичная система счисления

Четкое размещение максимума экономичности может быть установлено методом последующих рассуждений. **Пусть имеется п символов для записи чисел, а основание системы счисления р.** Тогда количество разрядов числа k = п/р, а полное количество чисел (N), которые могут быть составлены, равно:

$$N = p^{\frac{n}{p}}. \qquad (4.10)$$

Если считать N(p) непрерывной функцией, то можно отыскать то значение $p_т$, при котором iN воспринимает наибольшее значение. Функция имеет вид, представленный на рис.4.3.

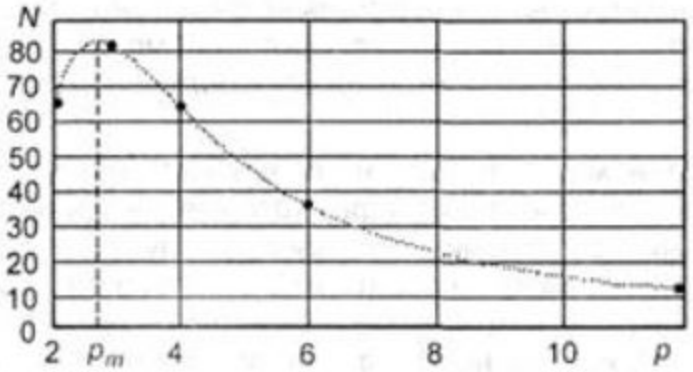


Рис. 4.1. Зависимость количества чисел от основания системы счисления при использовании 12-ти возможных цифр для записи чисел

Каждое целое беззнаковое число представляется вектором длины $k + l$ с k цифрами для целой части и l для дробной части. Так, например, вектор представляет число

$$x_{k-1} x_{k-2} \ldots x_1 x_0 x_{-1} x_{-2} \ldots x_{-l} .$$

Пример 2. Система счисления с отрицательным основанием $-q$, множество цифр $[0, q-1]$.

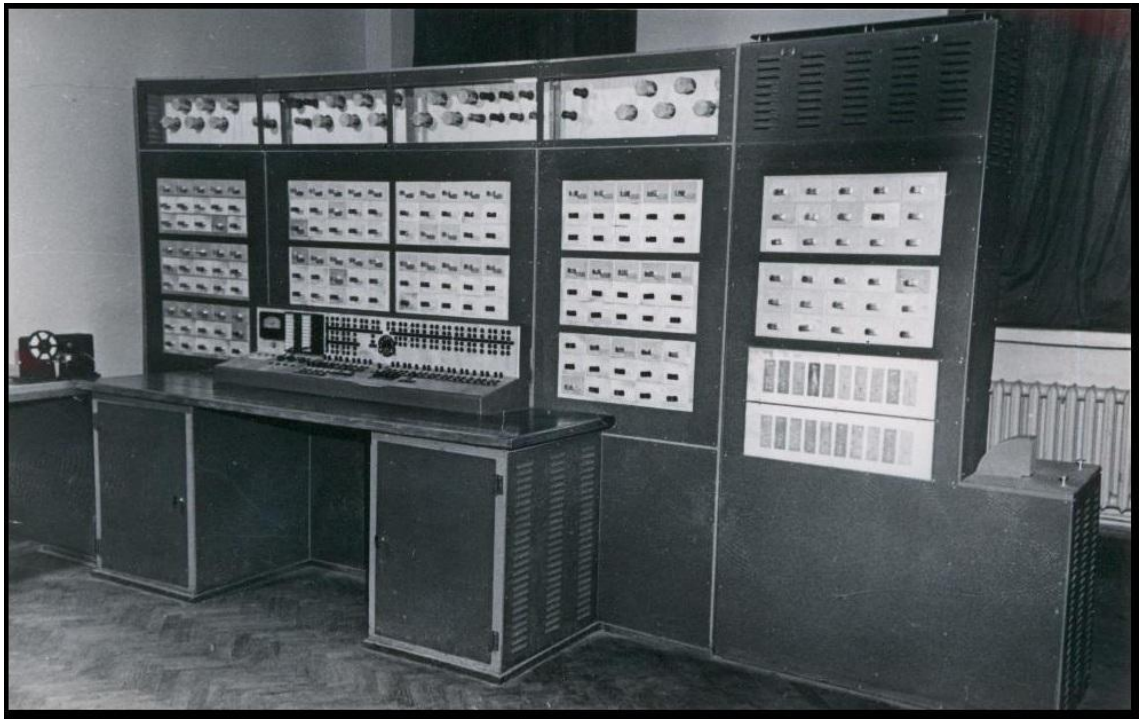
$$\sum_i x_i \cdot (-q)^i = \sum_{i \text{ чет}} x_i \cdot (q)^i - \sum_{i \text{ не чет}} x_i \cdot (q)^i$$

Пример 3. Неизбыточная знакоразрядная система счисления с множеством цифр $[-\alpha, q-1-\alpha]$ с основанием q, например, $[-4, 5]$ для основания $q = 10$, например,

$(3\ \ -1\ \ 5)_{10}$ представляет десятичное число 295= 300–10+5.

Пример 4. Избыточная знакоразрядная система счисления с цифрами $[-\alpha, \beta]$ с $\alpha + \beta \geq q$, например, множество цифр $[-7, 7]$ для $q = 10$. В такой избыточной системе счисления возможны неоднозначные представления чисел, например, число 295:

Сетунь – первый в мире троичный компьютер



2. ФОРМАТ С ПЛАВАЮЩЕЙ ТОЧКОЙ

В формате представления чисел с плавающей точкой имеем:

$$x = (-1)^s \cdot m \cdot q^e,$$

где

$s \in \{0,1\}$ - знак числа,

m – мантисса, $m \geq 0$,

e – экспонента (целое число).

Число с плавающей запятой имеет четыре компонента: знак s, мантиссу m, основание системы счисления q и показатель e. Вместе эти четыре компонента представляют собой число.

Мантисса числа x имеет n значащих цифр.

Специальный случай, когда $m = 0$ служит для представления нуля.

Пусть значение экспоненты принадлежит диапазону $e_{min} \le e < e_{max}$ Число называется *представимым* в формате с плавающей точкой, если его можно представить в виде:

$$(-1)^s \cdot m \cdot q^e, \text{ с } e_{min} \le e < e_{max}$$

Пусть рассматривается случай 1, когда мантисса удовлетворяет неравенству $q^{-1} \le m < 1$, тогда минимальное представимое число равно $q^{e\,min\ -1}$ и максимальное $q^{e\,max}\ (1 - q^{-n})$.

На рисунке 1 показан пример распределения чисел с плавающей точкой FLP и различные области. В частности, на рисунке 1 представлены такие специальные значения: $-\infty, 0, \infty$ (0 не может быть точно представимо в нормальной форме формата с плавающей точкой) и области переполнения и исчезновения порядка. Переполнение происходит, когда результат меньше $-\max$ или больше $\max$. С другой стороны, исчезновение порядка встречается для результатов в диапазон $(-\min, 0)$ или $(0, \min)$.

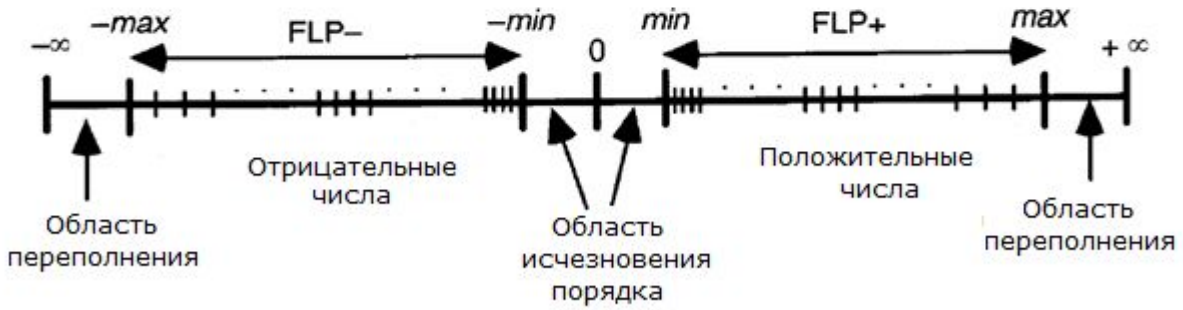


Рисунок 1. Области формата с плавающей точкой

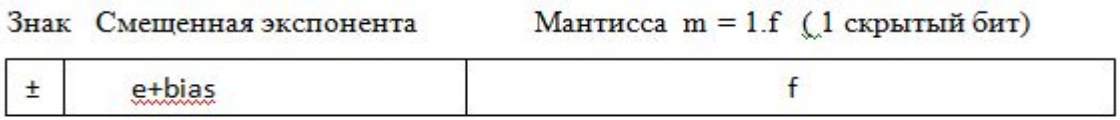
В соответствии с IEEE 754 форматом используется смещённое представление экспоненты, для формата с одинарной точностью если экспонента занимает 8 бит, смещение составляет 127, для двойной точности смещение составляет 1023 [ ]. Это означает, что если $\overline{k}$ смещенное значение экспоненты, тогда несмещенное значение экспонента равно $\overline{k}$ - 127,

Практически все цифровые компьютеры имеют отдельные форматы для целых чисел и чисел с плавающей запятой, хотя, в принципе, целые числа могут быть представлены в формате с плавающей запятой. Одна из причин заключается в том, что целочисленная арифметика является более простой и быстрой. Другая причина состоит в том, что с отдельным целочисленным форматом, который не имеет экспоненциальной части, большие числа могут быть представлены точно.

Если требуется представлять целых числа в формате с плавающей запятой, то рекомендуется включить в представление «неточный флаг». Для чисел, которые имеют точные представления в формате с плавающей запятой, «неточный флаг» установите значение 0. Когда результат вычисления с точными операндами слишком мал или слишком велик, чтобы быть представленным точно, «неточный флаг» результата может быть установлен в 1. Обратите внимание, что не все компиляторы имеют встроенную возможность получить доступ к этому флагу процессора.

Основная задача при разработке стандартного формата с плавающей точкой – сделать числовые программы предсказуемыми и полностью переносимыми в смысле получения тех же результатов при работе на разных машинах.

Формат с плавающей точкой одинарной точности имеет длину 32 бита, тогда как для двойной точности требуется 64 бит. Эти два формата имеют 8- и 11-битные поля экспоненты и используют экспоненциальные смещения 127 и 1023 соответственно.



32 бит: 8 бит, смещение = 127 23 + 1 бит, одинарная точность
64 бит: 11 бит, смещение = 1023 52 + 1 бит, двойная точность

Рисунок 2. IEEE стандарт формата с плавающей точкой

Мантисса находится в диапазоне [1, 2], с ее единственным целым битом, который всегда равен 1 и удаляется и отображается только дробная часть. Обозначения «23 + 1» или «52 + 1» для длины мантиссы объясняют роли скрытого бита, что вносит свой вклад в точность, не занимая места.

Поскольку 0 не может быть представлен нормированным значением, ему должен быть назначен специальный код. В стандартном формате IEEE ноль имеет представление с положительным или отрицательным знаком. Так как существует две бесконечности ($-\infty$, $+\infty$), естественно иметь два представления нуля ($+0 = 1/+\infty$, $-0 = 1/-\infty$) в соответствием со стандартом IEEE 754. Специальные коды NaN (не-число) также необходимы для представления неопределенных результатов, таких как, например, 0/0, $(+\infty) - (-\infty)$, корень квадратный из минус единицы и др. Специальные коды позволяют исключать распространение исключений в процессе вычислений и останавливать ход работы программы. Некоторые реализации формата с плавающей точкой отличаются между собой различными видами NaN.

Когда один из операндов есть специальное число, другое число с плавающей точкой, то результат определяется следующими правилами:

Обычное число $\div (+\infty) = \pm 0$,

$(+\infty) \times$ Обычное число $= \pm \infty$,

NaN + Обычное число = NaN.

Субнормальные или денормализованные числа необходимы в соответствии со стандартом IEEE 754, чтобы сделать эффект исчезновения порядка менее резким. Субнормальные значения определяются как числа без скрытой единицы и с наименьшим возможным показателем. Другими словами, некоторые небольшие значения, которые не представимы в качестве нормализованных чисел, должны быть округлены до 0, если они встречаются в ходе вычислений, могут быть более точно представлены, как денормализованные. Например, $(0{,}0001)_2\ 2^{-126}$ – это денормализованное число, которое не имеет нормализованного представления в формате одинарной точности IEEE.

Таблица 1. Параметры формата с плавающей точкой

Длина слова, бит	32	64
Длина мантиссы, бит	23 + 1 скрытый	52 + 1 скрытый
Диапазон мантиссы	$[1{,}2\text{-}2^{-23}]$	$[1{,}2\text{-}2^{-52}]$
Длина экспоненты, бит	8	11
Смещение экспоненты	127	1023
Ноль (±0)	Exp + bias = 0, f = 0	Exp + bias = 0, f = 0
Денормальная	Exp + bias = 0, f ≠ 0 представляется $\pm 0.f \times 2^{-126}$	Exp + bias = 0, f ≠ 0 представляется $\pm 0.f \times 2^{-1022}$
Бесконечность (±∞)	Exp + bias = 255, f = 0	Exp + bias = 2047, f = 0
Не число (NaN)	Exp + bias = 255, f ≠ 0	Exp + bias = 2047, f ≠ 0
Обычные числа	Exp + bias ϵ [1,254] Exp ϵ [-126,127] представление $1.f \times 2^e$	Exp + bias ϵ [1,2046] Exp ϵ [-1022,1023] представление $1.f \times 2^e$
Минимальное число	$2^{-126} \approx 1{,}2 \cdot 10^{-38}$	$2^{-1022} \approx 2{,}2 \cdot 10^{-308}$
Максимальное число	$2^{128} \approx 3{,}4 \cdot 10^{38}$	$2^{1024} \approx 1{,}8 \cdot 10^{308}$
Число десятичных цифр	7	16

На рисунке 3 представлена графическая интерпретация денормализованных чисел.

Рисунок 3. Графическая интерпретация субнормальных чисел

Пусть $x = 0.11 \cdot 2^{e_{min}}$, $y = 0.1 \cdot 2^{e_{min}}$,

Разность

$x - y = 0.01 \cdot 2^{e_{min}}$

нельзя точно представить в формате с плавающей точкой в нормальной форме и будет округляться до нуля. В следующем примере возникнет ошибка деления на ноль.

```
if (y!=x) then
    z = 1.0 / (y-x);
```

Субнормальные числа позволяют решить эту проблему. Разность становится представимой, следовательно, ошибка деления на ноль больше не возникнет с субнормальными числами.

Особенности формата с плавающей точкой

последствия

Формат с плавающей точкой

Неравномерное распределение чисел с плавающей точкой

Резкая потеря точности при вычислениях с разномасштабными величинами

Нарушение законов алгебры (коммутативности, дистрибутивности и др.)

$x \neq (x+x)-x$

Значения математических эквивалентных выражений могут быть не равными друг другу (вычислительные аномалии)

…

…

…

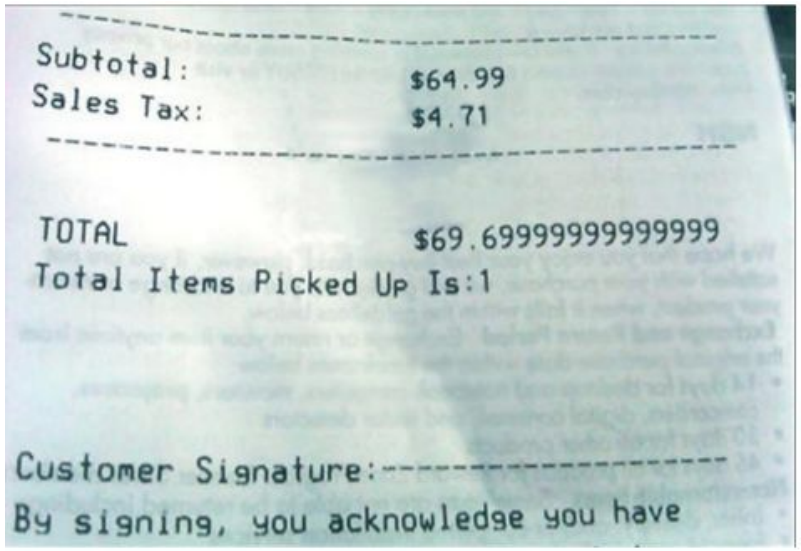
Нарушение законов алгебры

- No associative property for floats
- $(a + b) + (c + d)$ (parallel) $\neq ((a + b) + c) + d$ (serial)
- Looks like a “wrong answer”

Недостатки формата с плавающей точкой

1. Числа с плавающей точки дают различные результаты на различных аппаратных платформах.
2. Сложность использования численных методов (требуются экспертные знания в области Error Analyze)
3. Резкий рост времени вычислений при увеличении точности
4. В формате с плавающей точкой скрыты ошибки переполнения, исчезновения порядка (на флаги процессора никто не смотрит)

Пример ошибки при сложении чисел в формате с плавающей точкой:



Пример нарушения алгебраического свойства ассоциативности

$(a \oplus b) \oplus c \neq a \oplus (b \oplus c)$ $\oplus -$ сложение чисел с плавающей точкой

Любое число с плавающей точкой в нормальной форме можно представить в следующем виде:

$a \cdot q^b$,

где

q – основание системы счисления,

b – порядок числа,

a – мантисса числа и $q^{-1} \le |a| < 1$.

$q = 2,$

Мантисса числа с плав. точкой	Порядок числа			
	b = 0	b = 1	b = 2	b = 3
100	1/2	1	2	4
101	5/8	5/4	5/2	5
110	3/4	3/2	3	6
111	7/8	7/4	7/2	7

$$\left(\frac{1}{2} \oplus \frac{1}{2}\right) \oplus 6 \neq \frac{1}{2} \oplus \left(6 \oplus \frac{1}{2}\right)$$

$7 \neq 6$

Неравномерное распределение чисел с плавающей точкой

(Длина мантиссы k= 3,

порядок от 0 до 4.)

Пример.

$$\vec{\mathbf{x}} = (10^{17}, 1223, 10^{18}, 10^{15}, 3, -10^{12})$$

$$\vec{\mathbf{y}} = (10^{20},\ 2,\ -10^{19},\ 10^{13},\ 2111,\ 10^{16})$$

Истинный результат $(\vec{\mathbf{x}}, \vec{\mathbf{y}})$ = 8779

Вычисленный в формате с плав. точкой один. точн.

равен 4.6E+0020.

ПРИМЕР ЗАДАЧИ, ИМЕЮЩЕЙ РЕЗКИЙ РОСТ ОШИБОК ОКРУГЛЕНИЯ

Матрица Гильберта $A = \{a_{ij}\},\ a_{ij} = \dfrac{1}{i + j - 1}$

Обращение матрицы Гильберта порядка 3

$$A = \begin{bmatrix} 1 & 1/2 & 1/3 \\ 1/2 & 1/3 & 1/4 \\ 1/3 & 1/4 & 1/5 \end{bmatrix}$$

С точностью 2 знака после запятой

$$A_{прибл}^{-1} = \begin{bmatrix} -1{,}17 & 19{,}51 & -23 \\ 19{,}51 & -112{,}94 & 112 \\ -23 & 112 & -100 \end{bmatrix}$$

Макс. относ. погрешн. более 100%.

С точностью 3 знака после запятой

$$A_{прибл}^{-1} = \begin{bmatrix} 10{,}101 & 29{,}598 & 64{,}798 \\ -41{,}039 & -192{,}78 & -202{,}4 \\ 34{,}6 & 202{,}4 & 200 \end{bmatrix}$$

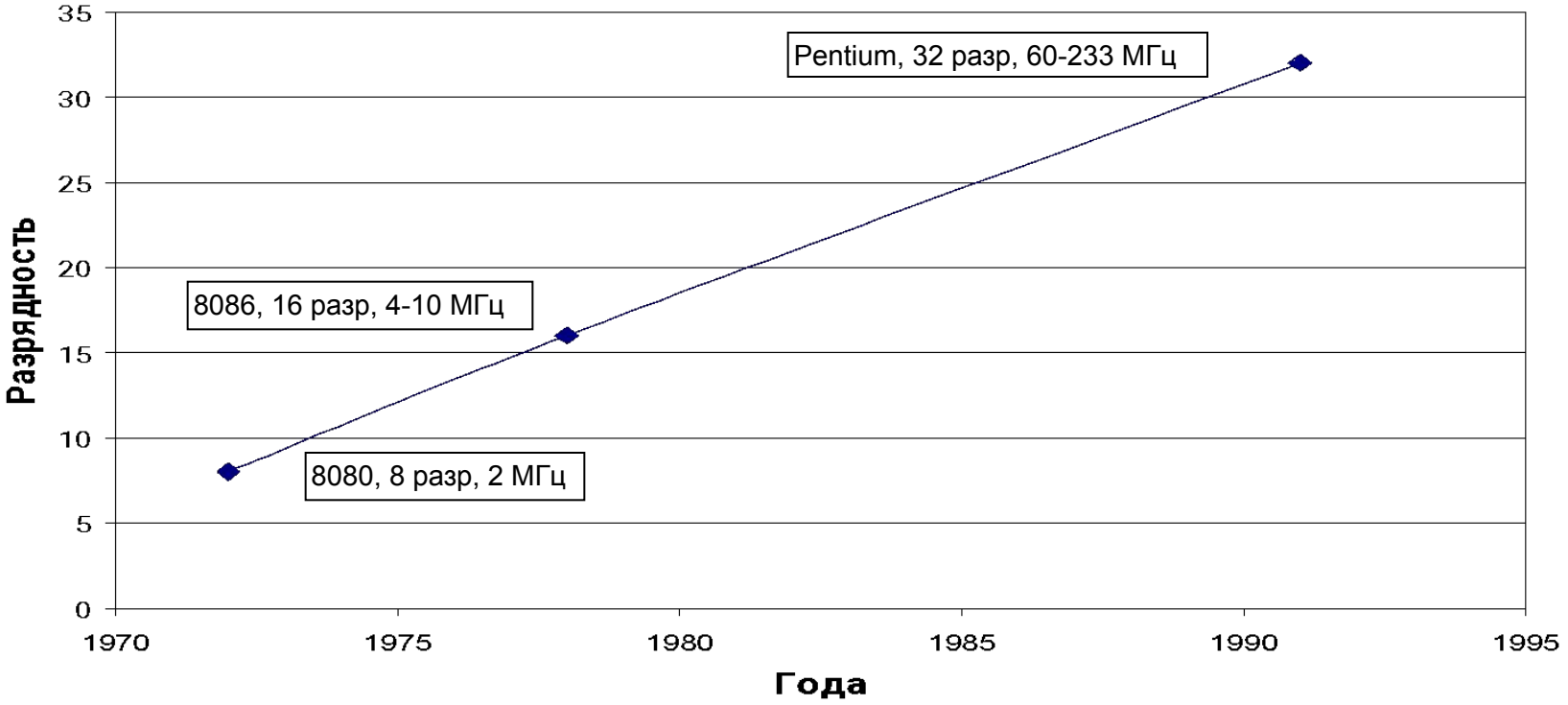
Макс. относ. погрешность более 100%.

Точный результат:

$$A_{точн}^{-1} = \begin{bmatrix} 9 & -36 & 30 \\ -36 & 192 & -180 \\ 30 & -180 & 180 \end{bmatrix}$$


Задачи критичные к точности компьютерных вычислений
Плохо обусловленные задачи с точно заданными исходными данными
Задачи, которые становятся плохо обусловленными при некоторых условиях
Задачи с разномасштабными коэффициентами
Обращение матрицы Гильберта
Задачи чувствительные к изменению шага дискретизации
Дискретное преобразование Фурье с сильно отличающимися по величине входными данными
Отсечение триангуляционного объекта, проецируемого на плоскость XY
Метод Эйлера для решения задачи Коши
Двумерная краевая задача для дифференциальных уравнений второго порядка
Краевая задача для нестационарного уравнения теплопроводности с сильно отличающимися друг от друга по величине коэффициентами

Рост разрядности и тактовой частоты процессоров по годам



Гипотеза: Технологические трудности создания процессоров высокой разрядности

Интервальная арифметика

Мы будем рассматривать всевозможные конечные вещественные интервалы $[a, b]$ $(a \leqslant b)$. Операции над ними определяются следующим образом:

- Сложение: $[a,b] + [c,d] = [a + c, b + d]$
- Вычитание: $[a,b] - [c,d] = [a - d, b - c]$
- Умножение: $[a,b] \times [c,d] = [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)]$
- Деление: $[a,b] / [c,d] = [\min(a/c, a/d, b/c, b/d), \max(a/c, a/d, b/c, b/d)]$

Получаем ответ в виде интервала, например: [10.8, 10.9].

Истинный результат содержится в этом интервале.

Недостаток подхода: расширение интервалов в процессе вычислений.

Например: [-100, 100].

Pascal XSC

Традиционный подход повышения точности вычислений

Применение библиотек высокоточных вычислений,

таких как: ZREAL(Россия), MPARITH(Германия), GMP(США)

и др. $2 \le n_{mant} = \text{var}$

Основная проблема

Резкое увеличение времени выполнения арифметических операций от точности. Это приводит к резкому росту времени решения задач большой размерности.

Подход к решению проблемы высокоточных вычислений на основе модулярной арифметики

К настоящему времени модулярная арифметика использовалась как средство повышения быстродействия в криптографии, нейронных сетях, цифровой обработке сигналов и др.

Проведенные исследования показали качественно новые возможности применения модулярной арифметики в повышении точности вычислений и ослаблении зависимости времени вычислений от точности, для некоторых частных задач:

- решение дифференциальных уравнений методами Рунге-Кутта,
- нахождение скалярного произведения векторов,
- решения систем линейных уравнений методами Гаусса-Зейделя,
- релаксации,
- дискретном преобразовании Фурье .

ПРИНЦИПЫ РЕАЛИЗАЦИИ МОДУЛЯРНЫХ ВЫЧИСЛЕНИЙ

Пусть m – некоторое простое число, называемое ***модулем***, тогда по теореме о делении с остатком любое целое число a можно представить в виде:

$$a = m \cdot q + r ,$$

где q – частное, r – остаток от деления a на m.

Если два целых числа при делении на m дают один и тот же остаток, то они называются ***равноостаточными*** или ***сравнимыми по модулю*** m. Например, числа 13 и 8 сравнимы по модулю 5, т.к. 13=2*5+3 и 8=1*5+3, или числа 10 и 3 сравнимы по модулю 7, т.к. 10=1*7+3 и 3=0*7+3.

Математически свойство сравнимости по модулю m двух чисел a и b записывается так:

$$a \equiv r(\bmod m) \text{ или } |a|_m \equiv r \qquad (1)$$

Пример. $13 \equiv 8(\bmod 5)$, $10 \equiv 3(\bmod 7)$.

ПРИНЦИПЫ РЕАЛИЗАЦИИ МОДУЛЯРНЫХ ВЫЧИСЛЕНИЙ

Свойства сравнений.

1°. ***Рефлексивность***.

Если $a = b$, то $a \equiv b (\mathrm{mod}\ m)$. А если $a < m$ и $b < m$, то из $a \equiv b (\mathrm{mod}\ m)$ следует, что $a = b$.

2°. Если $a \equiv r (\mathrm{mod}\ m)$, то и $a \pm k \cdot m \equiv r (\mathrm{mod}\ m)$.

Доказательство:

$a = q_1 \cdot m + r, \quad r_1 < m$

$a \pm k \cdot m = (q_1 + k) \cdot m + r$

3°. ***Сравнения можно почленно складывать***

Из $a_1 \equiv r_1 (\mathrm{mod}\ m)$ и $a_2 \equiv r_2 (\mathrm{mod}\ m)$ следует, что $a_1 + a_2 \equiv r_1 + r_2 (\mathrm{mod}\ m)$.

Доказательство:

$a_1 = q_1 \cdot m + r_1, \quad r_1 < m$

$a_2 = q_2 \cdot m + r_2, \quad r_2 < m$

$a_1 + a_2 = (q_1 + q_2) \cdot m + r_1 + r_2$

Если $r_1 + r_2 < m$, то $a_1 + a_2 = (q_1 + q_2) \cdot m + r_1 + r_2$, отсюда следует, что

$a_1 + a_2 \equiv r_1 + r_2 (\mathrm{mod}\ m)$,

Если $r_1 + r_2 \geq m$, то $a_1 + a_2 = (q_1 + q_2 + 1) \cdot m + (r_1 + r_2 - m)$

$a_1 + a_2 \equiv (r_1 + r_2 - m) (\mathrm{mod}\ m) \equiv (r_1 + r_2) (\mathrm{mod}\ m)$

ПРИНЦИПЫ РЕАЛИЗАЦИИ МОДУЛЯРНЫХ ВЫЧИСЛЕНИЙ

4°. ***Сравнения можно почленно перемножать, т.е.***

Из $a_1 \equiv b_1 (\bmod m)$ и $a_2 \equiv b_2 (\bmod m)$ следует, что $a_1 \cdot a_2 \equiv b_1 \cdot b_2 (\bmod m)$.

Из свойства 3° следует, что сравнения можно почленно возводить в степень, т.е.

Из $a \equiv b (\bmod m)$, следует, что $a^n \equiv b^n (\bmod m)$, где n – целое число.

Для представления ***отрицательных чисел*** используется формула:

$-a \equiv m - a (\bmod m)$,

Например,

$-5 \equiv 8 (\bmod 13)$,

$-15 \equiv -2 (\bmod 13) \equiv 13\text{-}2 (\bmod 13) \equiv 11 (\bmod 13)$,

т.к. $-15 = (-1)*13 - 2$

Рассмотрим ***деление сравнений***.

$$\frac{a}{b} (\bmod m) \equiv a \cdot (b)^{-1} (\bmod m),$$

где b^{-1} – число, ***обратное*** к b по модулю m, т.е. число x удовлетворяющее сравнению $x \cdot b (\bmod m) \equiv 1$.

Модулярная арифметика с дробями

Обратное число к a это такое число x, для которого выполняется соотношение:

$a \cdot x \equiv 1 (\mathrm{mod}\, m)$.

Пусть модуль $m = 7$, $a = 2$

$2 \cdot x \equiv 1 (\mathrm{mod}\, 7)$

$x = 4,\ 2 \cdot 4 \equiv 1 (\mathrm{mod}\, 7)$

Т.е. $\frac{1}{2} \mathrm{mod}\ 7 \equiv 4,$

Таблица соответствия по модулю $m = 7$.

Дробь	***Модулярное представление***
1/2	4
1/3	5
1/6	6
43/6, 1/6, 6 … ? (нет единственности)	**6**

Вычисления с дробями Фарея в модулярной арифметике

Дроби Фарея – это несократимые дроби, числители и знаменатели которых по абсолютной величине меньше чем некоторая константа, называемая порядком дробей Фарея, т.е.

$$F_N = \left(\frac{a}{b} \in Q, \quad |a| \le N, |b| \le N \right),$$

где F_N – множество дробей Фарея,

Q – множество рациональных чисел,

N – порядок дробей Фарея.

Например, дроби Фарея третьего порядка:

$$\frac{0}{1}, 1, 2, 3, -1, -2, -3, \frac{1}{2}, \frac{1}{3}, \frac{2}{3}, \frac{3}{2}, -\frac{1}{2}, -\frac{1}{3}, -\frac{2}{3}, -\frac{3}{2}.$$

Справедлива теорема о том, что если

$$2 \cdot N^2 + 1 \le m,$$

где m – модуль, то каждая дробь Фарея порядка N имеет единственное модулярное представление.

Пример 1 задачи, чувствительной к изменению шага интегрирования

Задача Коши

$x'(t)=t$, $x_0=0$, $t_0=0$

Точное решение:

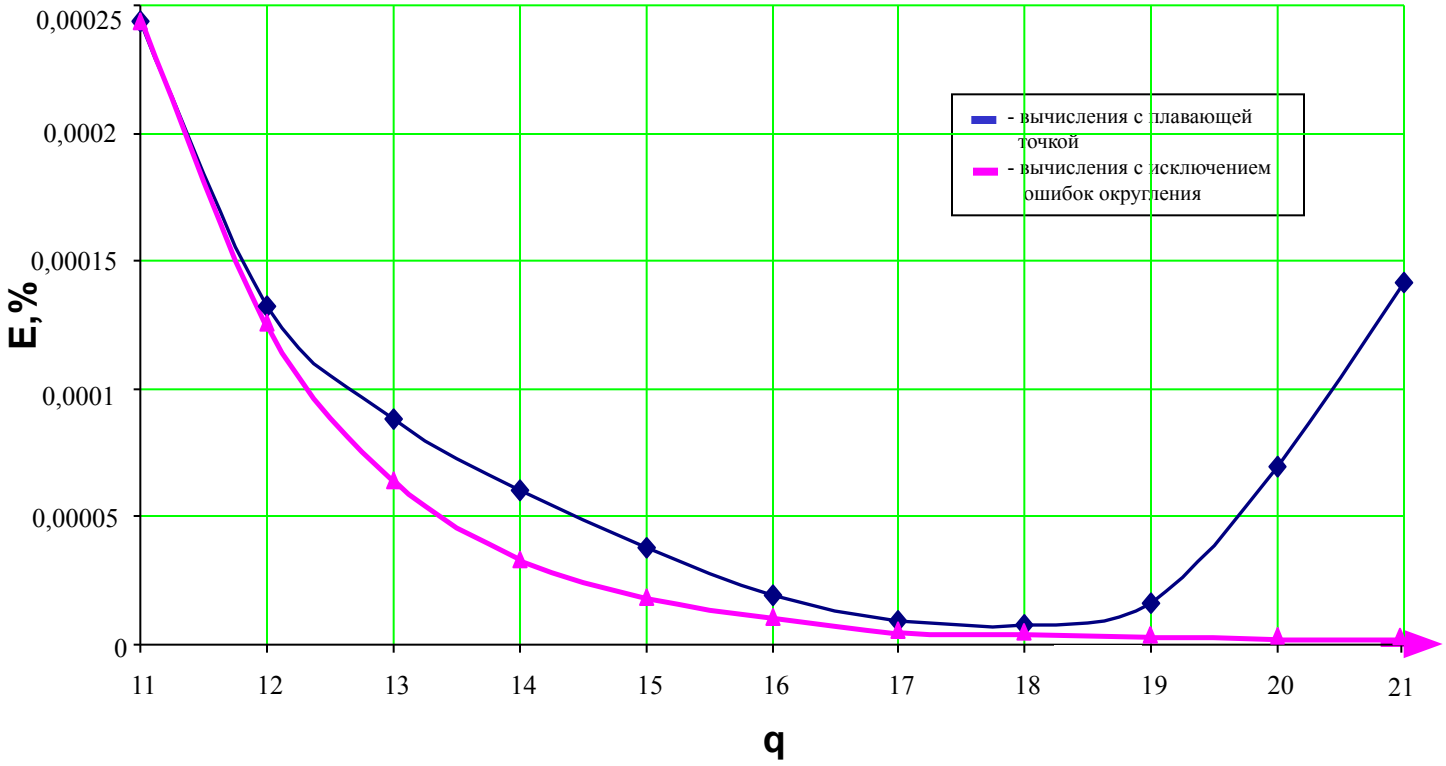
$$x(t)=\frac{t^2}{2}$$

Шаг интегрирования:

$$h=1/2^q, q>1$$

E – относительная погрешность решения

Результат решения методом Эйлера



Пример 2 задачи, чувствительной к изменению шага интегрирования

Простейшее дифференциальное уравнение

$$y''(x) = -f(x) \quad f(0) = 0,\ f(1) = 0,\ 0 \le x \le 1$$

$$h = \frac{1}{n},\ x = h,\ldots,\ x = nh$$

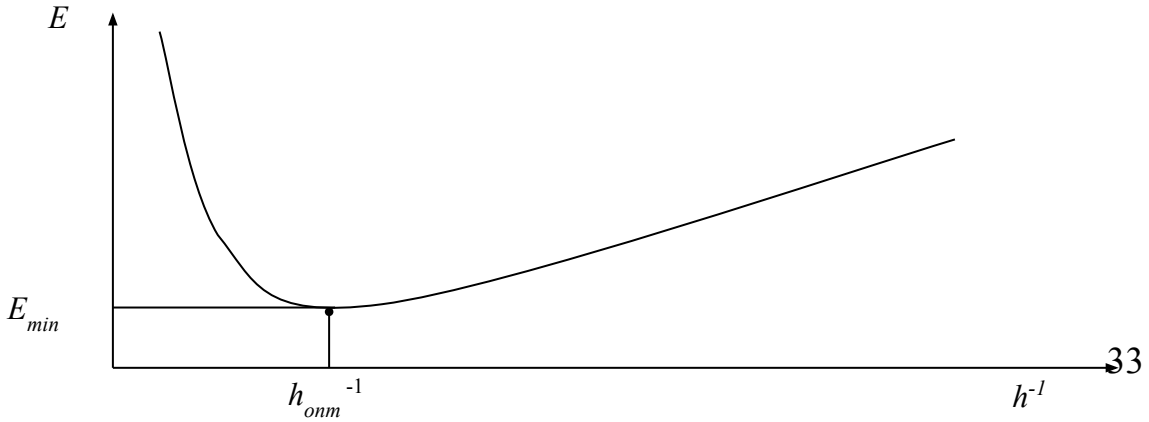
$$\frac{d^2y}{dx^2} \approx \frac{y(x+h) - 2y(x) + y(x-h)}{h^2}$$

$$-y_{j+1} + 2y_j - y_{j-1} = h^2 f(jh),$$

$$\begin{bmatrix} 2 & -1 & 0 & 0 & \ldots & 0 \\ -1 & 2 & -1 & 0 & \ldots & 0 \\ 0 & -1 & 2 & -1 & \ldots & 0 \\ . & . & . & . & . & . \\ 0 & \ldots & 0 & -1 & 2 & -1 \\ 0 & \ldots & 0 & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ . \\ \\ y_n \end{bmatrix} = h^2 \begin{bmatrix} f(h) \\ f(2h) \\ f(3h) \\ \ldots \\ f((n-1)h) \\ f(nh) \end{bmatrix}$$

Число обусловленности:

$$k(n) \approx \frac{4(n+1)^2}{\pi^2}$$



ПРИМЕНЕНИЕ ВЫЧИСЛЕНИЙ С ИСКЛЮЧЕНИЕМ ОШИБОК ОКРУГЛЕНИЯ В ВЫЧИСЛИТЕЛЬНО НЕУСТОЙЧИВЫХ АЛГОРИТМАХ

Рассмотрим задачу вычисления функции e^x . Известно, что эта задача хорошо обусловлена.

$$\nu_\delta \approx |x|$$

Обусловленность вычислительного алгоритма при x<0

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots + \frac{x^n}{n!} + \ldots . \qquad \nu_\delta = e^{2\cdot|x|} \quad \text{при } x<0$$

Пример.

Найти значение функции e^x при x= -15.

Верное значение $e^{-15} = 1 / e^{15} \approx 0.000000305902$

1. Традиционные вычисления

После выполнения 82 итераций было получено: $e^{-15} \approx 0.000000256502$

Относительная погрешность составила 19,2%.

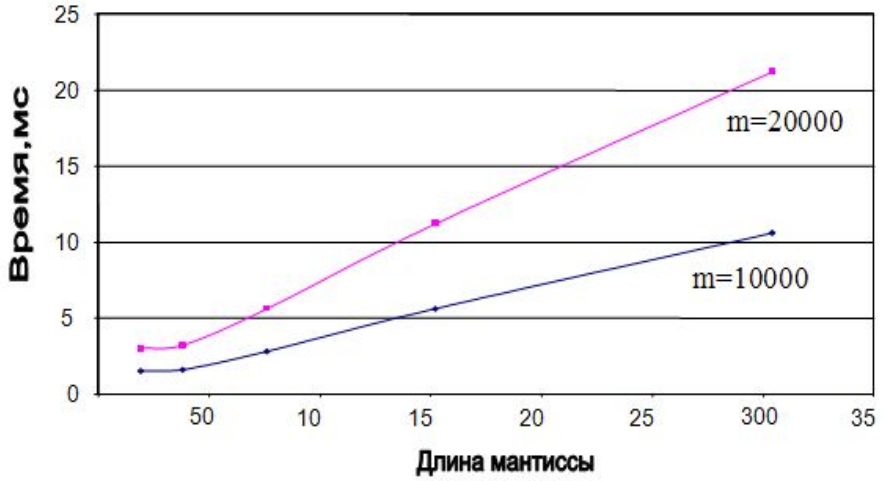
2. Вычисления с исключ. ошибок окр.

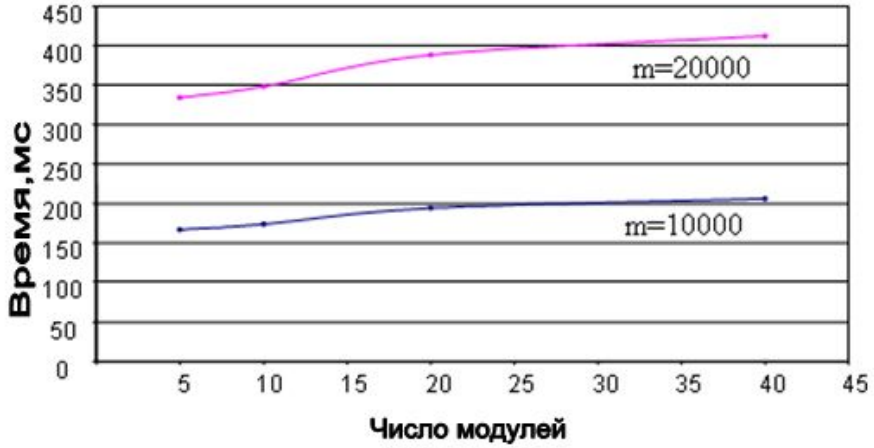
После выполнения 60 итераций было получено:

$$e^{-15} \approx \frac{1822987410130384149007132206840681602541990778449289}{5959360479558424668259567532453435686337875113375015790 1824}$$

или $e^{-15} \approx 0.000000305903159$. Отн. погр. равна 0,0001%

Оценка эффективности высокоточных вычислений на примере нахождения скалярного произведения



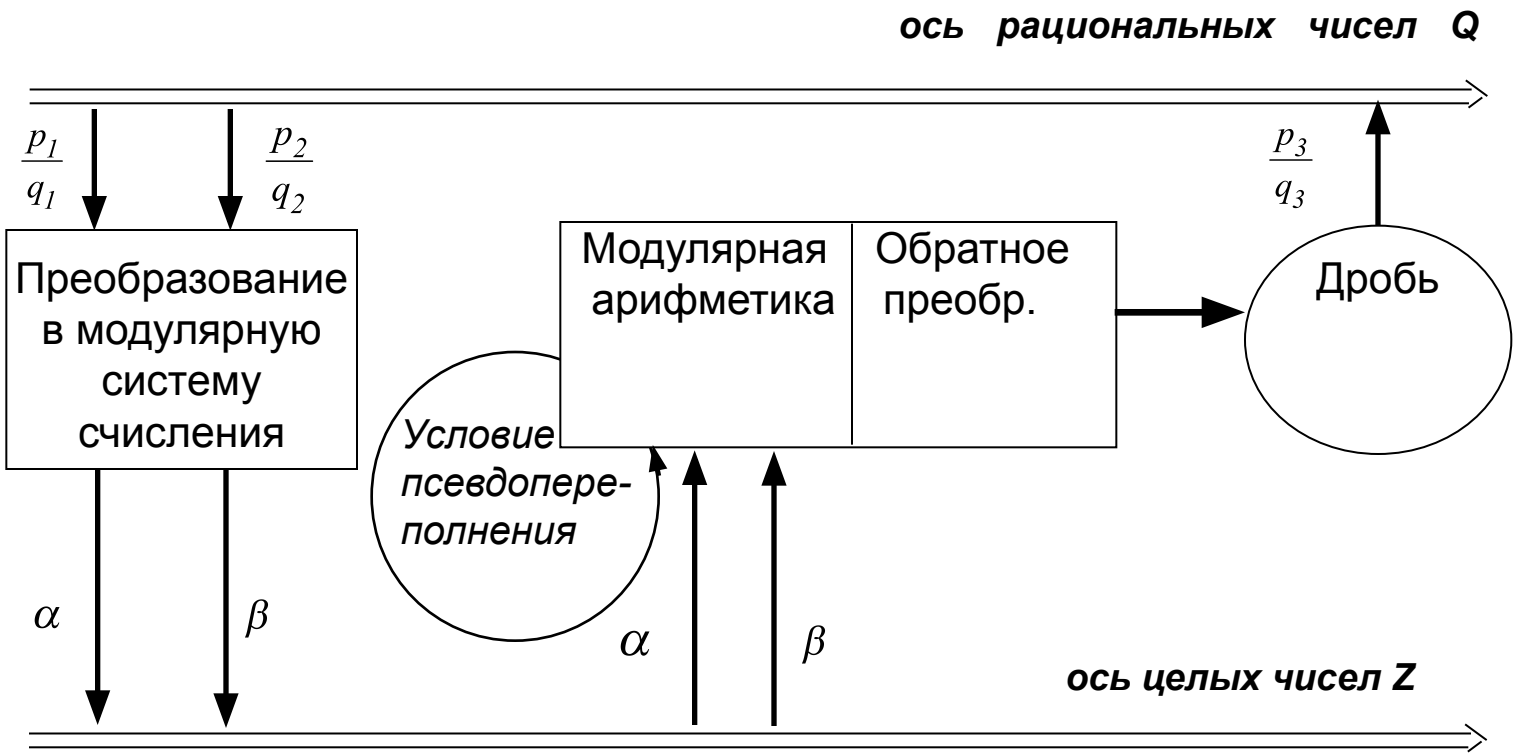


$$Eff = \frac{T_1}{T_2},$$

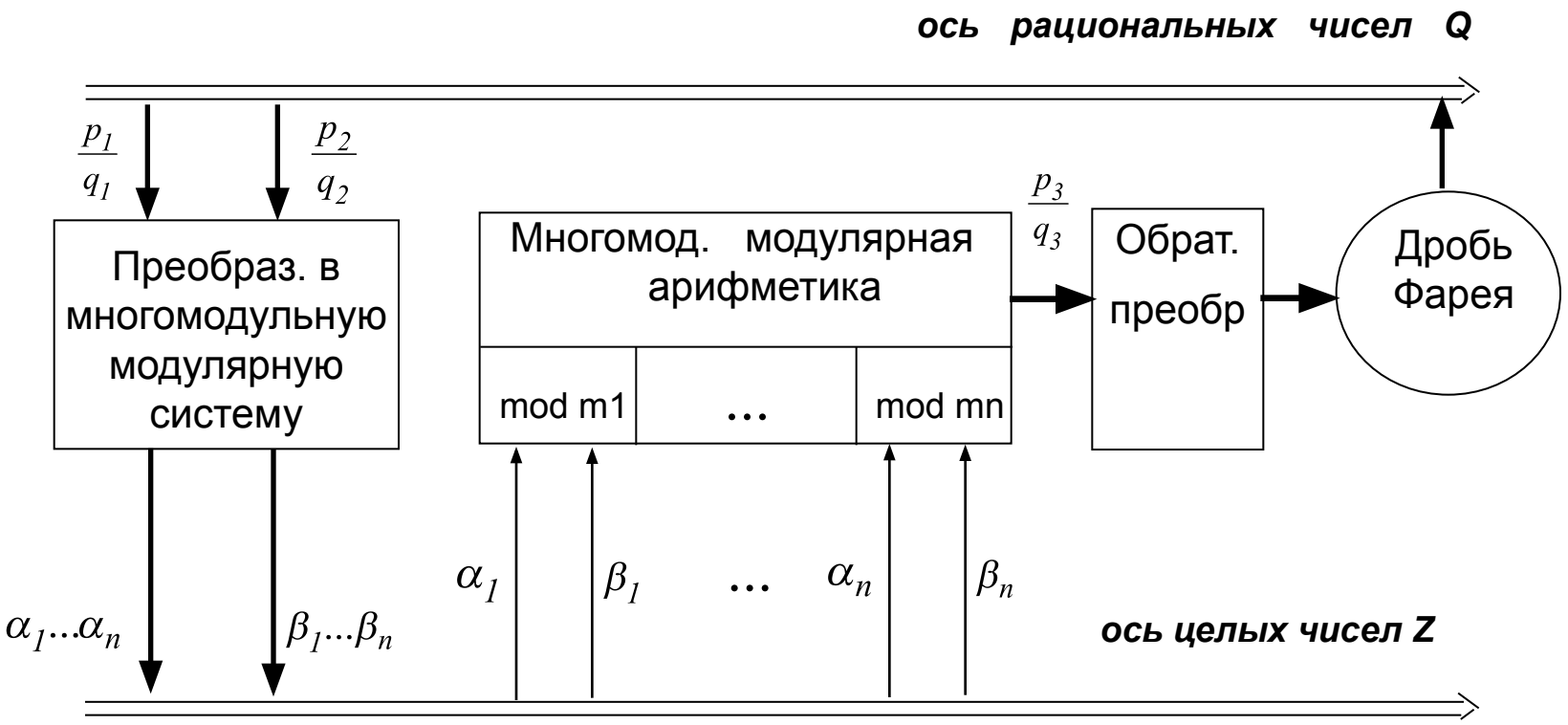
T_1 - время вычислений с использованием библиотеки MPArith,

T_2 - время вычислений в модулярной арифметике при той же точности.

МОДЕЛЬ ВЫЧИСЛЕНИЙ В МОДУЛЯРНОЙ АРИФМЕТИКИ



МОДЕЛЬ ВЫЧИСЛЕНИЙ С ИСКЛЮЧЕНИЕМ ОШИБОК ОКРУГЛЕНИЯ НА ОСНОВЕ МНОГОМОДУЛЬНОЙ МОДУЛЯРНОЙ АРИФМЕТИКИ



Порядок дробей Фарея

$$N = \left\lfloor \sqrt{\frac{m_1 \cdot m_2 \cdot \ldots \cdot m_n - 1}{2}} \right\rfloor$$

ИСХОДНЫЕ ПРИНЦИПЫ РЕАЛИЗАЦИИ МОДЕЛИ

Поле p-адических чисел определяется как пополнение множества рациональных чисел по p-адической метрике, которая является неархимедовой и для нее выполняется неравенство «равнобедренного треугольника»

Любое рациональное число α имеет единственное p-адическое разложение:

$$\alpha = \sum_{j=n}^{\infty} a_j p^j, \text{ где } a_j \in \{0,1,\ldots, p-1\}, \ \|\alpha\|_p = p^{-n}$$

Код Гензеля H(p,r,α) - отрезок длины r бесконечного p-адического разложения числа α.

Теорема

$$\text{Пусть } \alpha = \frac{a}{b} = \frac{c}{d} \cdot p^n, (c,d) = (c,p) = (d,p) = 1.$$

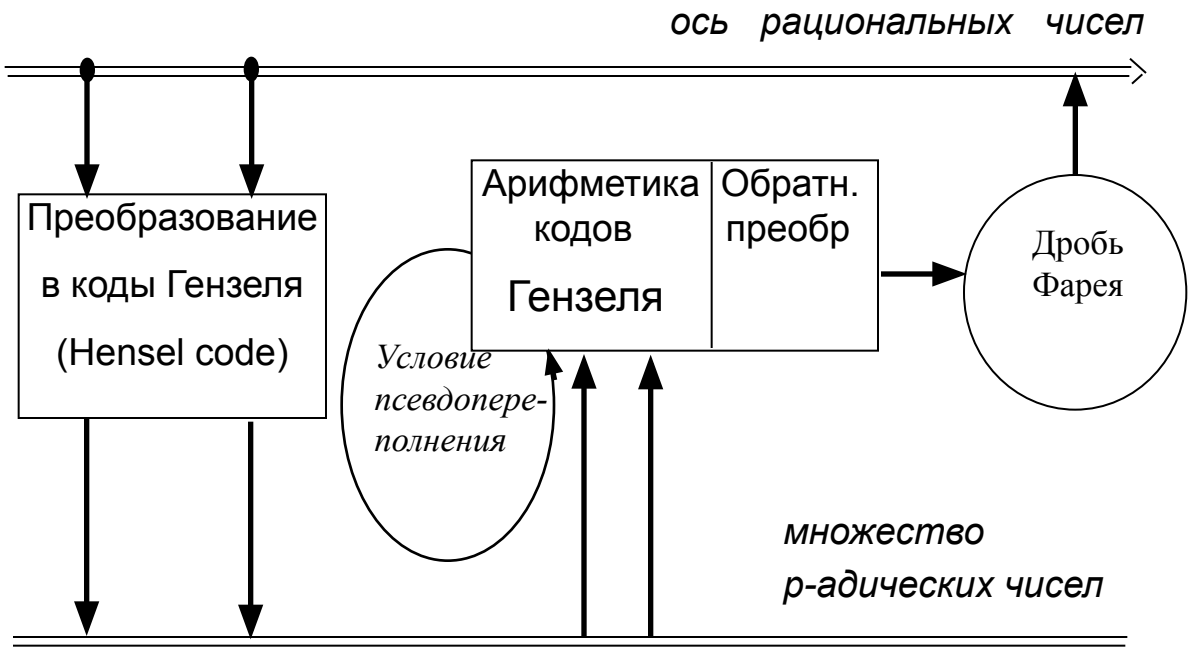
$$\text{Обозначим } H(p,r,c/d) = .a_0 a_1 \ldots a_{r-1},$$

тогда

$$(c \cdot d^{-1}) \bmod m = a_0 + a_1 p + a_2 p^2 + \ldots + a_{r-1} p^{r-1}$$

где $m=p^r$

МОДЕЛЬ ВЫЧИСЛЕНИЙ С ИСКЛЮЧЕНИЕМ ОШИБОК ОКРУГЛЕНИЯ НА ОСНОВЕ ОДНОМОДУЛЬНЫХ КОДОВ ГЕНЗЕЛЯ



Код Гензеля - конечно-разрядное p-адическое число $H(p,r,\alpha)$, для которого выполняется неравенство:

$2 \cdot N^2 + 1 \le p^r$, где N – порядок дроби Фарея, p – простое число, r – количество цифр в коде, α – дробь.

Операции сложения, вычитания, умножения и деление выполняются “слева направо”.

Цифры кода Гензеля в обратном порядке образуют p – ичное представление дроби по модулю p^r

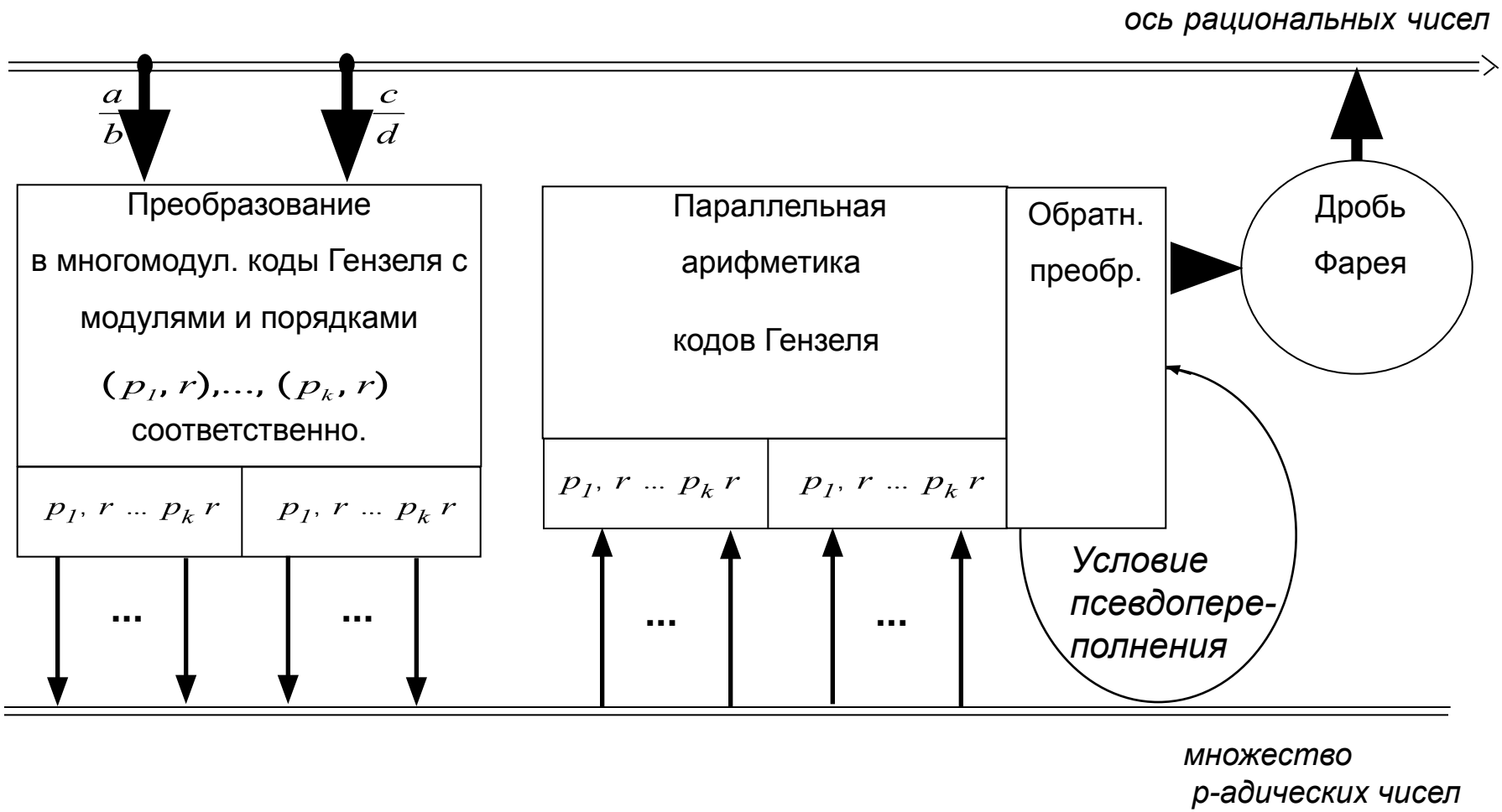
Пример. Найдем сумму $\frac{2}{3} + \frac{1}{4}$ в кодах Гензеля с $p = 5, r = 4$.

$$\left|\frac{2}{3}\right|_{54} = \left|2 \cdot 3^{-1}\right|_{625} = 209,\ [209]_{10} = [1314]_5, H(5,4,2/3) = .4131$$

$$\left|\frac{1}{4}\right|_{54} = 469,\ [469]_{10} = [3334]_5, H(5,4,1/4) = .4333$$

$$H(5,4,1/4) + H(5,4,2/3) = .4333 + .4131 = .3020,$$
$$[203]_5 = [53]_{10}, |11/12|_{625} = 53, \text{т.е. } (2/3) + (1/4) = (11/12)$$

МОДЕЛЬ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ С ИСКЛЮЧЕНИЕМ ОШИБОК ОКРУГЛЕНИЯ НА ОСНОВЕ МНОГОМОДУЛЬНЫХ КОДОВ ГЕНЗЕЛЯ



ПРИМЕР ВЫЧИСЛЕНИЙ С ИСКЛЮЧЕНИЕМ ОШИБОК ОКРУГЛЕНИЯ В МНОГОМОДУЛЬНОЙ СИСТЕМЕ ГЕНЗЕЛЯ

Найти сумму $\frac{1}{2} + \frac{1}{3}$

Выберем модули $p_1 = 5,\ p_2 = 7$, порядок $r = 2$

Определим сумму дробей в кодах Гензеля по каждому модулю.

$$H(5,2,1/2) + H(5,2,1/3) = .01,\ (10)_{p1} = (5)_{10}$$

$$H(7,2,1/2) + H(7,2,1/3) = .21,\ (12)_{p2} = (9)_{10}$$

Суммы могут вычисляться параллельно.

Применим Китайскую теорему об остатках для перевода числа:

(5,9)
из модулярной системы по модулям (25,49) в позиционную

счисления. Вычислим ортогональные базисы по формулам:

$$B_i = \partial_i \frac{M}{m_i},\ \text{где}\ \partial_i = \left|\frac{M}{m_i}\right|^{-1}_{m_i}$$

$$B_1 = 1176,\ B_2 = 50$$

$$A = |5 \cdot B_1 + 9 \cdot B_2|_M = 205,\ 205 = \left|\frac{5}{6}\right|_M$$

Сложность арифметических операций в кодах Гензеля в двоичной системе счисления:

$$+,- : O_B(r \log_2 p)$$
$$*,/ : O_B(r^2 \log_2 p)$$

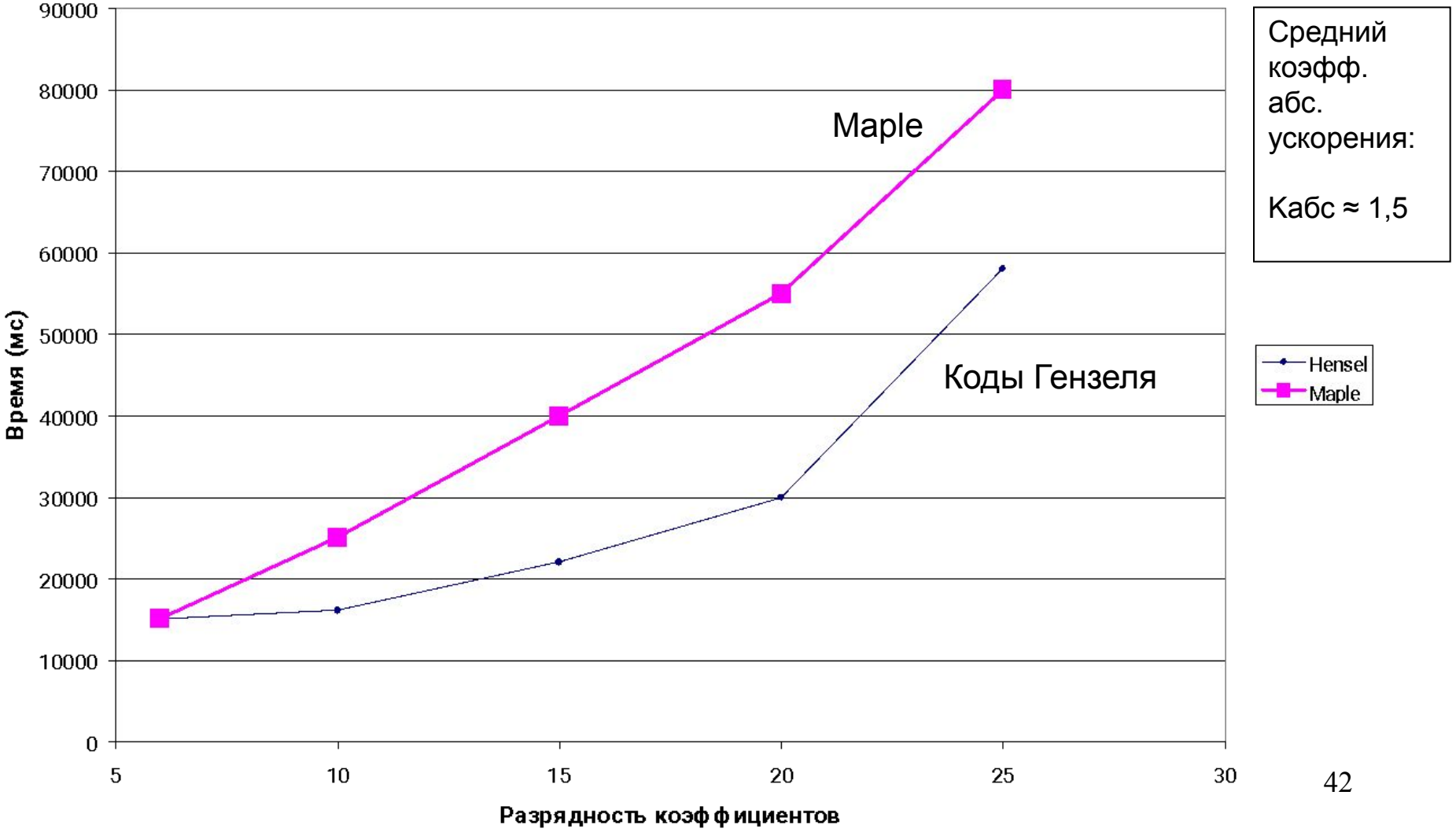
Коды Гензеля могут применяться:

Для реализации вычислений с полиномами (полиномиальная арифметика)

Для реализации вычислений с плавающей точкой без ошибок округления.

ОЦЕНКА ЭФФЕКТИВНОСТИ ВЫЧИСЛЕНИЙ С ИСКЛЮЧЕНИЕМ ОШИБОК ОКРУГЛЕНИЯ, РЕАЛИЗОВАННЫХ В MAPLE

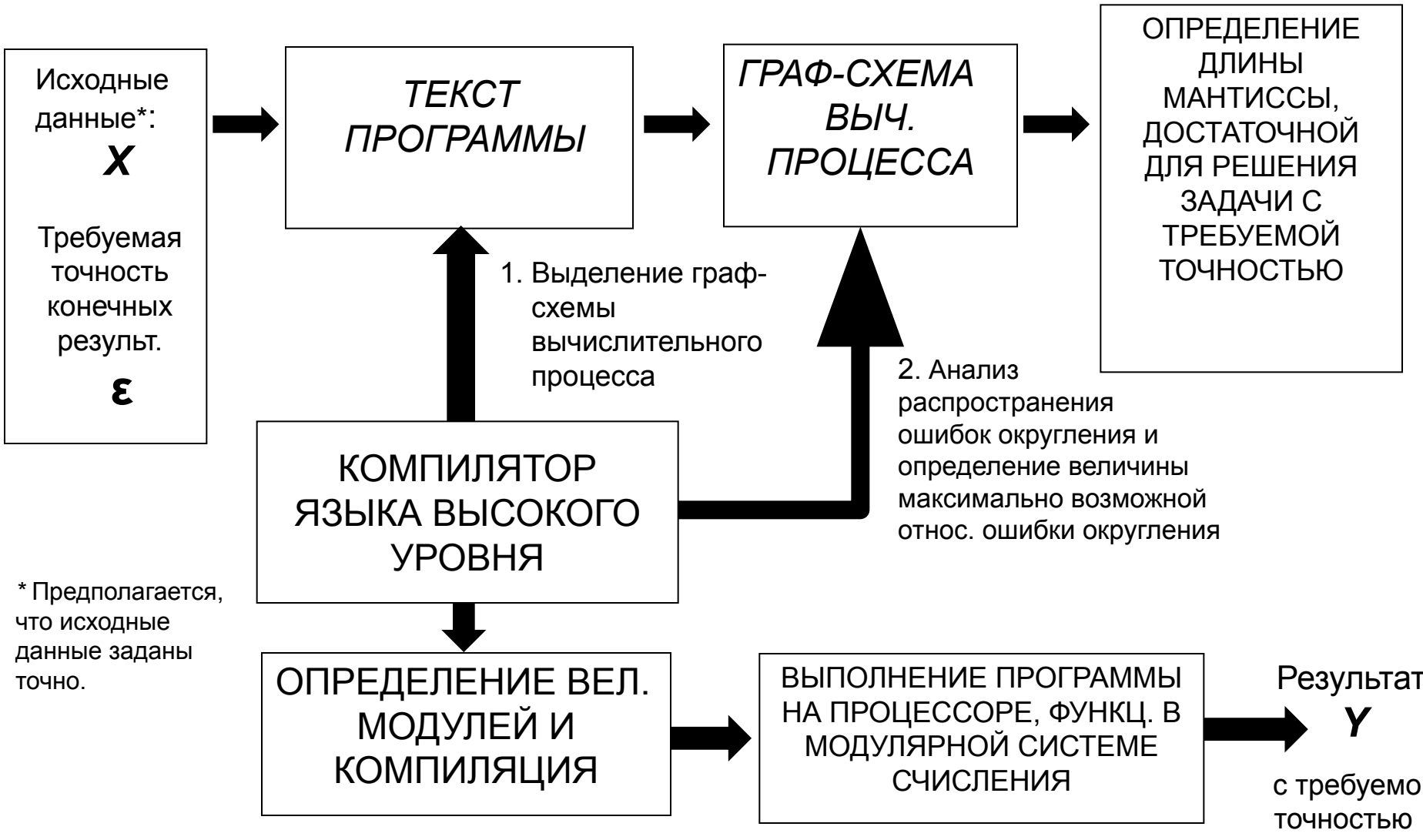
Эффективность исследовалась на примере решения системы линейных уравнений с рациональными коэффициентами размерностью 20



Средний коэфф. абс. ускорения:

Кабс ≈ 1,5

СХЕМА ОРГАНИЗАЦИИ ВЫЧИСЛЕНИЙ С ЗАДАННОЙ ТОЧНОСТЬЮ



ФОРМУЛЫ ОТНОСИТЕЛЬНЫХ ОШИБОК ОКРУГЛЕНИЯ

Пусть имеются два приближения $\bar{x}, \bar{y}$ к двум величинам x, y и e_x, e_y - соответствующие абсолютные ошибки.

Пусть t – количество значащих цифр в любом действительном числе, тогда при использовании правила отбрасывания максимальная относительная ошибка округления выразится так:

$$\left|\frac{e_y}{y}\right| = 10^{-t+1}$$

При симметричном округлении максимальная относительная погрешности выразится так:

$$\left|\frac{e_y}{y}\right| \le \frac{1}{2} \cdot 10^{-t+1}$$

Формулы относительных ошибок при 4-х арифметических операциях имеют вид:

$$\frac{e_{x+y}}{\bar{x}+\bar{y}} = \frac{\bar{x}}{\bar{x}+\bar{y}}\left(\frac{e_x}{\bar{x}}\right) + \frac{\bar{y}}{\bar{x}+\bar{y}}\left(\frac{e_x}{\bar{y}}\right) + r \qquad \frac{e_{x-y}}{\bar{x}-\bar{y}} = \frac{\bar{x}}{\bar{x}-\bar{y}}\left(\frac{e_y}{\bar{x}}\right) - \frac{\bar{y}}{\bar{x}-\bar{y}}\left(\frac{e_y}{\bar{y}}\right) + r$$

$$\frac{e_{x\cdot y}}{\bar{x}\cdot\bar{y}} = \frac{e_x}{\bar{x}} + \frac{e_x}{\bar{y}} + r \qquad \frac{e_{x/y}}{\bar{x}/\bar{y}} = \frac{e_x}{\bar{x}} - \frac{e_y}{\bar{y}} + r$$

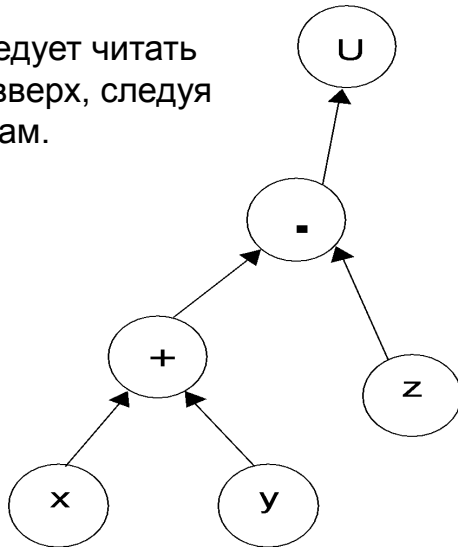
где r – ошибка округления.

ВЫДЕЛЕНИЕ ГРАФ-СХЕМЫ ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА

Пусть даны x,y,z и необходимо вычислить u=(x+y)*z

Граф вычислительного процесса имеет следующий вид:

Его следует читать снизу вверх, следуя стрелкам.

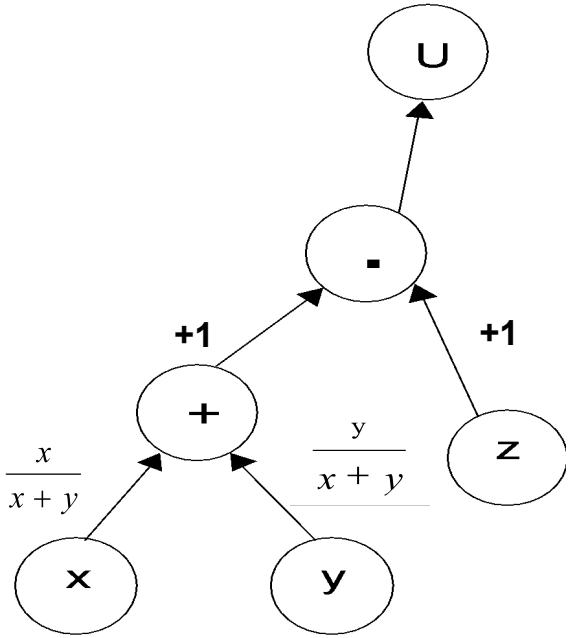


Предположим, что три исходные величины имеют относительные ошибки округления, равные соответственно

$$i_x, i_y, i_z$$

Рассмотрим сложение. Относит. ошибка величины x составляет i_x эта ошибка войдет в результат следующей операции (сложения) умноженной на коэффициент у стрелки, соединяющей x в кружке со знаком + в кружке:

$$\frac{x}{x+y} i_x$$



АНАЛИЗ РАСПРОСТРАНЕНИЯ ОШИБОК ОКРУГЛЕНИЯ

$$\frac{e_{x+y}}{x+y} = \frac{x}{x+y} i_x + \frac{y}{x+y} i_y + r_1$$

После выполнения операции умножения появляется ошибка r_2. Полная ошибка результата операции умножения выразится следующим образом:

$$\frac{e_u}{u} = \frac{x}{x+y} i_x \cdot 1 + \frac{y}{x+y} i_y \cdot 1 + r_1 \cdot 1 + i_z \cdot 1 + r_2.$$

Если все результаты соответствующим образом округлены, то ни одна из ошибок округления не превзойдет

$5 \cdot 10^{-t}$

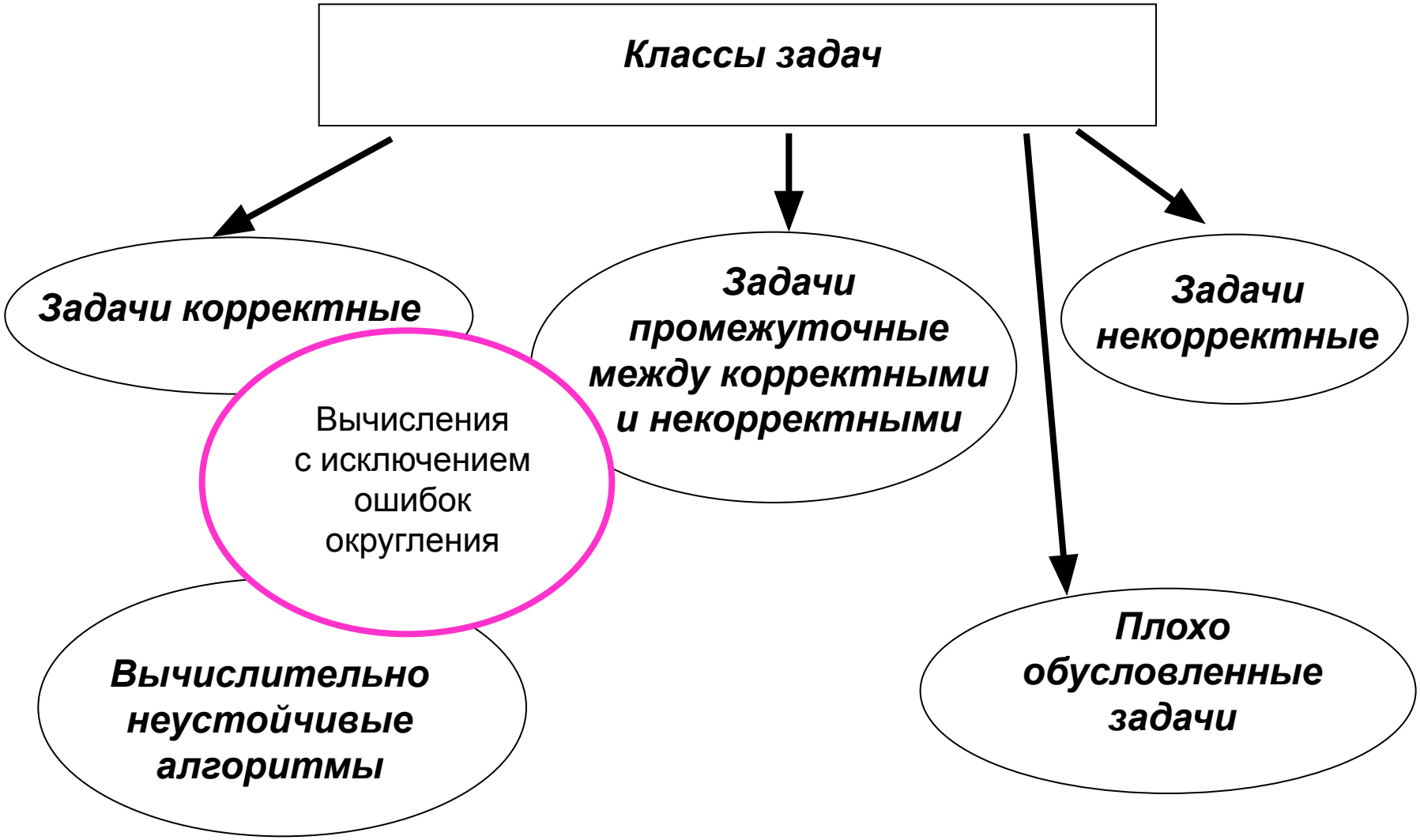
Поэтому

$\left|\frac{e_u}{u}\right| \le \left(\left|\frac{x}{x+y}\right| + \left|\frac{y}{x+y}\right| + 3\right) \cdot 5 \cdot 10^{-t}$ x, y, оба неотрицательные, то

$\left|\frac{x}{x+y}\right| + \left|\frac{y}{x+y}\right|$ Не может быть больше 1, и окончательно имеем: $\left|\frac{e_u}{u}\right| \le 20 \cdot 10^{-t} = 2 \cdot 10^{-t+1}$

ВОЗМОЖНЫЕ ПРИЛОЖЕНИЯ ВЫЧИСЛЕНИЙ С ИСКЛЮЧЕНИЕМ ОШИБОК ОКРУГЛЕНИЯ

1. **Точное вычисление обобщенных обратных матриц.**
   Например, таких как, g-обратная матрица Мура-Пенроуза. Многие алгоритмы требуют умения распознавать значение численного ранга, а это является трудной задачей при наличии ошибок округления.
2. **Целочисленное решение систем линейных уравнений.**
   Примером могут служить построение оптимальных решений в задачах целочисленного программирования.
3. **Точное вычисление характеристического многочлена матрицы.**
   Вследствие ошибок округления будут получены приближенные значения коэффициентов. Если многочлен плохо обусловлен, то корни "приближенного» характеристического уравнения могут быть плохими приближениями к корням истинного уравнения.
4. **Обращение матриц Гильберта, Адамара и др. особо чувствительных к ошибкам округления.**
5. **Для решения промежуточных между классами корректных и некорректных задач.**
   Класс задач, изменяющих корректность при решении. Это расчет устойчивости систем управления, выч. собств. знач. систем лин. одн. урав. и др.


Классы задач
Задачи корректные
Задачи промежуточные между корректными и некорректными
Задачи некорректные
Вычисления с исключением ошибок округления
Вычислительно неустойчивые алгоритмы
Плохо обусловленные задачи

Ф.С. Зайцев

Математическое моделирование эволюции тороидальной плазмы.

Семашко Н.Н Кафедра физики и ядерного синтеза (МЭИ)

Динамическая устойчивость энергосистем

...