

Федеральное агентство по образованию



Государственное образовательное учреждение высшего
профессионального образования
«МАТИ» - Российский государственный
технологический университет им. К.Э. Циолковского

ПРИБЛИЖЁННЫЕ ВЫЧИСЛЕНИЯ

Методические указания к лабораторной и курсовой работам
по курсам «Вычислительная математика» и «Информатика. Численные
методы»

Составители Гурьев Е.К., Зотов В.А.

Москва 2007

Данное руководство предназначено для студентов, изучающих численные методы в рамках курсов «Вычислительная математика» и «Информатика. Раздел Численные методы» и выполняющих лабораторные и курсовые работы. В методических указаниях рассмотрены источники и классификация погрешностей результата численного решения задачи, понятия абсолютной и относительной погрешности приближенного числа, правила записи приближенных чисел, округление этих чисел, погрешности арифметических операций над приближенными числами, погрешности функций одной и нескольких переменных, погрешности неявной функции, особенности машинной арифметики, представление вещественных и целых чисел в памяти ЭВМ с учетом особенностей различных форматов хранения данных и процедур ввода и вывода этих данных, понятие машинного эпсилон и его оценка программным путем. Также рассмотрены приближенные вычисления по формуле с использованием правил подсчета цифр, строгого учета предельных абсолютных погрешностей, метода границ. Изложение методов и приемов приближенных вычислений сопровождается большим количеством примеров, большинство из которых выполнены с помощью программирования в системе программирования Borland Pascal и дублируются в интегрированном пакете MATHCAD 11. Дан пример выполнения и оформления курсовой работы по теме «Приближенные вычисления». Приводятся варианты заданий для студентов.

ОГЛАВЛЕНИЕ

1. Источники и классификация погрешностей результата численного решения задачи.....	4
2. Приближенные числа. Абсолютная и относительная погрешности.....	4
2.1. Абсолютная и относительная погрешности.....	5
2.2. Правила записи приближенных чисел.....	7
2.3. Округление.....	9
3. Погрешности арифметических операций над приближёнными числами..	15
4. Погрешность функции.....	22
4.1. Погрешность функции многих переменных.....	22
4.2. Погрешность функции одной переменной.....	25
4.3. Погрешность неявной функции.....	29
5. Особенности машинной арифметики.....	29
5.1. Системы счисления.....	29
5.2. Представление целых чисел.....	30
5.3. Представление вещественных чисел.....	30
5.3.1. Особенности представления чисел с фиксированной и плавающей запятой в памяти ЭВМ в различных форматах данных.....	32
5.3.2. Погрешности вычисления элементарных функций с учётом представления в ЭВМ вещественных чисел в различных форматах записи данных и особенностей процедуры вывода writeln в системе программирования Borland Pascal.....	35
5.4. Арифметические операции над числами с плавающей точкой.....	36
5.5. Удвоенная точность.....	36
5.6. Вычисление машинного эпсилон.....	37
5.7. Универсальная формула оценки погрешности вычислений.....	39
6. Приближённые вычисления по формуле.....	39
6.1. Вычисления по правилам подсчета цифр.....	40
6.2. Вычисления со строгим учетом предельных абсолютных погрешностей.....	42
6.3. Вычисления по методу границ.....	45
7. Контрольные вопросы.....	48
8. Варианты заданий для лабораторных и курсовых работ.....	50
9. Образец оформления курсовой работы по дисциплинам «Информатика» и «Вычислительная математика».....	61
Литература.....	97

1. Источники и классификация погрешностей результата численного решения задачи

Получаемое на ЭВМ решение y^* почти всегда (за исключением специальных случаев) содержит погрешность, т.е. является приближенным. Невозможность получения точного решения следует уже из ограниченной разрядности вычислительной машины. Наличие погрешности решения обусловлено рядом причин:

1°. Математическая модель является приближенным описанием реального процесса. Характеристики процесса, вычисленные в рамках принятой модели, заведомо отличаются от истинных характеристик, причем их погрешность зависит от степени адекватности модели реальному процессу.

2°. Исходные данные, как правило, содержат погрешности, поскольку они либо получаются в результате экспериментов (измерений), либо являются результатом решения некоторых вспомогательных задач.

3°. Применяемые для решения задачи методы в большинстве случаев являются приближенными. Найти решение возникающей на практике задачи в виде конечной формулы возможно только в отдельных, очень упрощенных ситуациях.

4°. При вводе исходных данных в ЭВМ, выполнении арифметических операций и выводе результатов на печать производятся округления.

Пусть y — точное значение величины, вычисление которой является целью поставленной задачи. Соответствующая первым двум из указанных причин погрешность $d_i y$ называется *неустранимой погрешностью*. Такое название вызвано тем, что принятие математической модели и задание исходных данных вносит в решение ошибку, которая не может быть устранена далее. Уменьшить эту погрешность можно только, переходя к более точной математической модели и задавая более точные исходные данные.

Погрешность $d_i y$, источником которой является метод решения задачи, называется *погрешностью метода*, а погрешность $d_a y$, возникающая из-за округлений при вводе, выводе и вычислениях, — *вычислительной погрешностью*. Полная погрешность результата решения задачи на ЭВМ $dy = y - y^*$ складывается из трех составляющих: неустранимой погрешности, погрешности метода и вычислительной погрешности, т.е. $dy = d_i y + d_m y + d_a y$.

При фиксированных математической модели и входных данных повлиять на значение величины $d_i y$ нельзя. Знание порядка величины $d_i y$ позволяет осознанно выбрать метод решения задачи и разумно задать его точность. Погрешность метода должна быть в 2—10 раз меньше неустранимой погрешности. Большее значение $d_m y$ снижает точность результата, а меньшее требует увеличения затрат, практически уже не влияя на значение полной погрешности.

Вычислительная погрешность при фиксированных модели, входных данных и методе решения в основном определяется характеристиками используемой ЭВМ. Желательно, чтобы величина $d_a y$ была хотя бы на порядок меньше величины погрешности метода. Анализ погрешностей при решении прикладной задачи и соблюдение разумного компромисса между ними позволяет существенно экономить используемые ресурсы.

2. Приближенные числа.

Абсолютная и относительная погрешности

Числа, получаемые при решении на ЭВМ прикладных задач, как правило, являются приближенными. Поэтому вопрос о *точности* результатов, т.е. о мере их отклонения от истинных значений, в теории и практике методов вычислений имеет огромное значение. Рассмотрим основные понятия элементарной теории погрешностей. Введем обозначения, которые будут использоваться при сравнении величин. Кроме привычных знаков "=", "≠", "<", "≤", "≥", будем использовать знаки приближенного равенства " $\approx$ " и приближенного неравенства ">". В случае, когда положительные величины a и b являются величинами одного

порядка (т.е. $10^{-1} < a/b < 10$) будем использовать обозначение $a \approx b$. Если же a много меньше b , то будем писать $a \ll b$, что эквивалентно соотношению $a/b \ll 1$.

2.1. Абсолютная и относительная погрешности. Пусть a - точное (вообще говоря, неизвестное) значение некоторой величины, a^* - известное приближенное значение той же величины {приближенное число}. Ошибкой (или погрешностью) приближенного числа a^* называют разность $a - a^*$ между точным и приближенным значениями.

Простейшей количественной мерой ошибки является *абсолютная погрешность*

$$\Delta(a^*) = |a - a^*| \quad (1)$$

Однако по величине абсолютной погрешности далеко не всегда можно сделать правильное заключение о качестве приближения. Действительно, если $\Delta(a^*) = 0.1$, то следует ли считать погрешность большой или нужно признать ее малой? Ответ существенным образом зависит от принятых единиц измерения и масштабов величин. Если $a \approx 0.3$, то скорее всего точность приближения невелика; если же $a \approx 3 \cdot 10^8$, то следует признать точность очень высокой. Таким образом, естественно соотнести погрешность величины и ее значение, для чего вводится понятие *относительной погрешности* (при $a \neq 0$)

$$d(a^*) = \frac{|a - a^*|}{|a|} = \frac{\Delta(a^*)}{|a|} \quad (2)$$

Относительные погрешности не зависят от масштабов величин и единиц измерения. Для приведенного выше примера $d(a^*) \approx 0.33 = 33\%$ в первом случае и $d(a^*) \approx 0.33 \cdot 10^{-9} = 0.33 \cdot 10^{-7} \%$ во втором.

Так как значение a неизвестно, то вычисление величин $\Delta(a^*)$ и $d(a^*)$ по формулам (1), (2) невозможно. Более реалистичной является задача получения оценок погрешности вида

$$|a - a^*| \leq \bar{\Delta}(a^*), \quad (3)$$

$$\frac{|a - a^*|}{|a|} \leq \bar{d}(a^*), \quad (4)$$

где $\bar{\Delta}(a^*)$ и $\bar{d}(a^*)$ - известные величины, которые мы будем называть *верхними границами абсолютной и относительной погрешностей*.

Если величина $\bar{\Delta}(a^*)$ известна, то неравенство (4) будет выполнено, если положить

$$\bar{d}(a^*) = \frac{\bar{\Delta}(a^*)}{|a|}, \quad (5)$$

Точно так же если величина $\bar{d}(a^*)$ известна, то следует положить

$$\bar{\Delta}(a^*) = |a| \bar{d}(a^*). \quad (6)$$

Поскольку значение a неизвестно, то при практическом применении формулы (5), (6) заменяют приближенными равенствами

$$\bar{d}(a^*) \approx \frac{\bar{\Delta}(a^*)}{|a^*|}, \quad \bar{\Delta}(a^*) \approx |a^*| \bar{d}(a^*) \quad (7)$$

З а м е ч а н и е . Точность в качественных рассуждениях обычно выступает как противоположность погрешности, хотя для количественного их измерения используются одни и те же характеристики (например, абсолютная и относительная погрешности). Точное значение величины - это значение, не содержащее погрешности. Повышение точности воспринимается как уменьшение погрешности, а снижение точности - как увеличение погрешности. Часто используемая фраза "требуется найти решение с заданной точностью ϵ " означает, что ставится задача о нахождении приближенного решения, принятая мера погрешности которого не превышает заданной величины ϵ . При этом считается, что из контекста ясно, как измеряется величина погрешности.

Пример 1. Вычислить абсолютную и относительную погрешности числа $x^*=2.5732$, заданного всеми своими верными цифрами в строгом (узком) смысле. Абсолютная погрешность $\Delta(x^*)=0,00005$. Вычисления выполнить с помощью микрокалькулятора, программирование в системе BORLAND PASCAL и в MathCad(e).

1) С помощью микрокалькулятора вычисляем относительную погрешность

$$d(x^*) = \frac{\Delta(x^*)}{x^*} = \frac{0.00005}{2.5732} = 1.94310586040802E-0005$$

2) Вычислим эти погрешности с помощью программирования в системе Borland Pascal и в MATHCAD(e).

2.1) РЕШЕНИЕ В СИСТЕМЕ ПРОГРАММИРОВАНИЯ BORLAND PASCAL

```
program pogrchnisla;
uses crt;
var x,dx,deltax: real;
begin
clrscr;
writeln('vvedite x');
read(x);
writeln('vvedite dx');
read(dx);
writeln('x=',x:5:5);
writeln('dx=',dx:5:5);
deltax:=dx/x;
writeln('deltax=',deltax);
repeat until keypressed
end.
```

Результат

Введите x 2.5732

Введите dx 0.00005

deltax= 1.94310586040802E-0005

2. 2)РЕШЕНИЕ В MATHCAD(e)

$$x := 2.573\%$$

$$\Delta x := 0.0000\%$$

$$\delta(x) := \frac{\Delta x}{x}$$

$$\delta(x) = 1.943 \times 10^{-5}$$

2. 2. Правила записи приближенных чисел. Пусть приближенное число a^* задано в виде конечной десятичной дроби:

$$a^* = a_n a_{n-1} \dots a_0 . b_1 b_2 \dots b_m$$

Значащими цифрами числа a^* называют все цифры в его записи, начиная с первой ненулевой цифры слева.

Пример 2. Для чисел $a^* = 0.0103$ и $a^* = 0.0103000$ значащие цифры подчеркнуты. Первое число имеет 3, а второе - 6 значащих цифр.

Значащую цифру числа a^* называют верной (в широком смысле), если абсолютная погрешность числа не превосходит единицы разряда, соответствующего этой цифре.

Значащую цифру числа a^* называют верной (в строгом или узком смысле), если абсолютная погрешность числа не превосходит половины единицы разряда, в котором стоит эта цифра.

Пример 3. Если $\Delta(a^*) = 2 \cdot 10^{-6}$, то число $a^* = 0.0103000$ имеет 4 верные значащие цифры (они подчеркнуты).

Первая отброшенная (неверная) цифра называется сомнительной. Говорят, что приближенное число записано правильно, если в его записи все цифры верные. Если число записано правильно, то только по одной его записи в виде десятичной дроби можно судить о точности этого числа.

Пример 4. Пусть приближенное число $a^* = 16.784$ содержит все верные цифры. Так как последняя цифра 4 верная, то абсолютная погрешность $\Delta(a^*) = 0.001$. Тогда можно принять $a^* = 16.784 \pm 0.001$.

Правильная запись приближенных данных обязывает выписывать нули в последних разрядах, если эти нули являются верными цифрами.

Пример 5. Для числа $a^* = 109.070$ последний нуль означает, что цифра 0 в конце числа верная и абсолютная погрешность $\Delta(a^*) = 0.001$.

Широко распространенной ошибкой при записи приближенных чисел является отбрасывание последних значащих нулей (даже если они представляют собой верные цифры).

З а м е ч а н и е . Верная цифра приближенного числа, вообще говоря, не обязана совпадать с соответствующей цифрой в записи точного числа. Таким образом, термин "верная цифра" не следует понимать буквально (см. пример 6).

Пример 6. Пусть $a = 1.00000$, $a^* = 0.99999$. Тогда $\Delta(a^*) = 0.00001$ и у числа a^* все подчеркнутые цифры - верные, хотя они и не совпадают с соответствующими цифрами числа a .

Количество верных значащих цифр числа тесно связано с величиной его относительной погрешности. Приведенные ниже утверждения позволяют в дальнейшем связывать точность числа с количеством его верных значащих цифр и трактовать потерю точности как потерю верных цифр.

Утверждение 1. Если число a^* содержит N верных значащих цифр, то справедливо неравенство

$$d(a^*) \leq (10^{N-1} - 1)^{-1} \approx 10^{-N+1}$$

Утверждение 2. Для того чтобы число a^* содержало N верных значащих цифр, достаточно, чтобы было выполнено неравенство

$$d(a^*) \leq (10^N + 1)^{-1} \approx 10^{-N}$$

Утверждение 3. Если число a^* имеет ровно N верных значащих цифр, то $10^{-N-1} < d(a^*) < 10^{-N+1}$ и поэтому $d(a^*) \approx 10^{-N}$.

Пример 7. Что можно сказать об относительной погрешности числа a^* , если оно содержит 3 верные цифры?

В силу утверждения 1 имеем $d(a^*) < 10^{-2} = 1\%$.

Пример 8. С какой относительной точностью следует найти число a^* , чтобы верными оказались 6 значащих цифр?

Из утверждения 2 следует, что достаточно найти a^* с относительной точностью $e \approx 10^{-6}$.

Границы абсолютной и относительной погрешностей принято записывать с одной или двумя значащими цифрами. Большая точность в записи этих величин, как правило, не имеет смысла, так как обычно они являются довольно грубыми оценками истинных значений погрешностей, и, кроме того, для практического использования часто бывает достаточно знать только их порядок.

Пример 9. Информация о погрешности вида $d(a^*) \approx 0.288754 \cdot 10^{-5}$ практически равноценна информации $d(a^*) \approx 3 \cdot 10^{-6}$, причем последняя вызывает больше доверия. Скорее всего, вполне удовлетворительной в данном случае является и запись $d(a^*) \approx 10^{-6}$.

Неравенство (3) эквивалентно двойному неравенству

$$a^* - \bar{\Delta}(a^*) \leq a \leq a^* + \bar{\Delta}(a^*)$$

и поэтому тот факт, что число a^* является приближенным значением числа a с верхней границей абсолютной погрешности $\bar{\Delta}(a^*)$ (с абсолютной точностью $e = \bar{\Delta}(a^*)$), принято записывать в виде

$$a = a^* \pm \bar{\Delta}(a^*)$$

Как правило, числа a^* и $\bar{\Delta}(a^*)$ указывают с одинаковым числом цифр после десятичной точки.

Пример 10. Пусть для числа a известны приближенное значение $a^*=1.648$ и граница абсолютной погрешности $\bar{\Delta}(a^*)=0.002832$. Тогда можно записать $a=1.648 \pm 0.003$. Записи вида $a=1.648 \pm 0.002832$ или $a=1.648 \pm 0.1$ являются неестественными.

Из неравенства (4) следует, что значение a заключено примерно между $a^*(1-\bar{d}(a^*))$ и $a^*(1+\bar{d}(a^*))$. Поэтому тот факт, что число a^* является приближенным значением числа a с границей относительной погрешности $\bar{d}(a^*)$ (с *относительной точностью* $e=\bar{d}(a^*)$), принято записывать в виде $a=a^*(1 \pm \bar{d}(a^*))$.

Пример 11. Оценим точность часто используемого в простейших расчетах приближения $p^*=3.14$ к числу p . Известно, что $p=3.14159\dots$, поэтому $p-p^*=0.00159\dots$. Следовательно, можно принять $\bar{\Delta}(p^*)=0.0016$ и $\bar{d}(p^*) \approx 0.0016/3.14 \approx 0.00051 = 0.051\%$. Итак, $p=3.14(1 \pm 0.00051)$.

З а м е ч а н и е . Если число a^* приводится в качестве результата без указания величины погрешности, то принято считать, что все его значащие цифры являются верными. Однако нельзя слишком доверять выводимым из ЭВМ числам. Анализ результатов вычислений и определение степени их достоверности дело совсем непростое. Одна из целей изучения вычислительных методов как раз и состоит в понимании того, что можно и чего нельзя ожидать от результатов, полученных на ЭВМ.

2.3. Округление. Часто возникает необходимость в *округлении* числа a , т.е. в замене его другим числом a^* с меньшим числом значащих цифр. Возникающая при такой замене погрешность называется *погрешностью округления*.

Существует несколько способов округления числа до n значащих цифр. Наиболее простой из них - *усечение* состоит в отбрасывании всех цифр, расположенных справа от n -й значащей цифры. Более предпочтительным является *округление по дополнению*. В простейшем варианте это правило округления состоит в следующем. Если первая слева из отбрасываемых цифр меньше 5, то сохраняемые цифры остаются без изменения. Если же она больше либо равна 5, то в младший сохраняемый разряд добавляется единица.

Абсолютная величина погрешности округления при округлении по дополнению не превышает половины единицы разряда, соответствующего последней оставляемой цифре, а при округлении усечением - единицы того же разряда.

Пример 12. При округлении числа $a=1.72631$ усечением до трех значащих цифр получится число $a^*=1.72$, а при округлении по дополнению - число $a^*=1.73$.

Границы абсолютной и относительной погрешностей принято всегда округлять в сторону увеличения.

Пример 13. Округление величин $\bar{\Delta}(a^*)=0.003721$ и $\bar{d}(a^*)=0.0005427$ до двух значащих цифр дает значения $\bar{\Delta}(a^*)=0.0038$ и $\bar{d}(a^*)=0.00055$.

При округлении числа x^* возникает погрешность, называемая погрешностью округления: $\Delta_{\text{окр}}=|x^*-x_1|$, где x^* - данное число, x_1 - результат округления. Так как число $x^*=2.5732$ задано всеми своими верными значащими цифрами в широком смысле, то абсолютная погрешность числа x^* $\Delta(x^*)=0.0001$. Поэтому абсолютная погрешность округленного числа Δx_1 вычисляется по формуле

$$\Delta x_1 = \Delta(x^*) + \Delta x_{\text{вв}}.$$

Пример 14. Округлить число $x^*=2.5732$, заданного всеми своими верными значащими цифрами в широком смысле, до 3 значащих цифр. Округлённое значение $x_1=2.57$ и погрешность округления $\Delta_{\text{окр}}=0.0032$. Выполнить округление

числа $x^*=2.5732$ с помощью программирования в системе Borland Pascal, в MATHCAD(e) и программное округление по заданному значению абсолютной погрешности с помощью специальной программы округление.

1) ПРОГРАММА ОКРУГЛЕНИЯ ЧИСЕЛ В СИСТЕМЕ ПРОГРАММИРОВАНИЯ BORLAND PASCAL.

```
program okrugchislo;
uses crt;
var x,x1,deltax,deltaokr,deltax1,otndeltax:real;
begin
clrscr;
writeln('Введите число x');
read(x);
writeln('Введите округленное число x1');
read(x1);
writeln('Введите абсолютную погрешность числа deltaokr');
read(deltax);
deltaokr:=abs(x-x1);
writeln('Погрешность округления deltaokr=',deltaokr:7:6);
deltax1:=deltax+deltaokr;
writeln('абсолютная погрешность округленного числа deltaokr= ',deltax1:7:6);
otndeltax:=deltax1/x1;
writeln('относительная погрешность округленного числа otndeltax=',otndeltax:7:6);
end.
```

Введите число x

2.5732

Введите округленное число x1

2.57

Введите абсолютную погрешность числа deltaokr

0.0001

Погрешность округления deltaokr=0.003200

абсолютная погрешность округленного числа deltaokr= 0.003300

относительная погрешность округленного числа otndeltax=0.001284

2) РЕШЕНИЕ ЗАДАЧИ ОКРУГЛЕНИЯ В MATHCAD(e)

Округлить число $x=2.5732$ до трех значащих цифр, найти абсолютную и относительную погрешности полученных приближений. Цифры верны в широком смысле.

$x := 2.5732$

Округленное число $x_1=2.57$

$$x_1 := 2.57$$

По определению верной цифры числа в широком смысле абсолютная погрешность $\Delta(x^*)=0.0001$

$$\Delta x := 0.0001$$

Погрешность округления определяется по формуле $\Delta_{\text{окр}}=|x-x_1|$

$$\Delta_{\text{окр}} := |x - x_1| \quad \Delta_{\text{окр}} = 3.2 \times 10^{-3}$$

Абсолютная погрешность округленного числа определяется по формуле $\Delta x_1 = \Delta x + \Delta_{\text{окр}}$

$$\Delta x_1 := \Delta x + \Delta_{\text{окр}} \quad \Delta x_1 = 3.3 \times 10^{-3}$$

$$\Delta x_1 = 3.3 \times 10^{-3} < 1 \cdot 10^{-2}$$

Это означает, что в числе x три цифры (2,5,7) верны в строгом смысле по абсолютной погрешности

Относительная погрешность округленного числа определяется по формуле $\delta x_1 = \Delta x_1 / x_1$

$$\delta x_1 := \frac{\Delta x_1}{x_1} \quad \delta x_1 = 1.284 \times 10^{-3}$$

$$\Delta x_1 = 3.3 \times 10^{-3} < 1 \cdot 10^{-2}$$

Это означает, что в числе x три цифры (2,5,7) верны в строгом смысле по относительной погрешности

3) ПРОГРАММНОЕ ОКРУГЛЕНИЕ ДЕСЯТИЧНЫХ ЧИСЕЛ ПО ЗАДАННОМУ ЗНАЧЕНИЮ АБСОЛЮТНОЙ ПОГРЕШНОСТИ

В некоторых языках программирования имеется функция $ROUND(x, n)$, где x - арифметическое выражение, задающее округляемое значение, а n - арифметическое выражение, задающее уровень округления. Округление с помощью функции $ROUND$ выполняется следующим образом:

- а) при $n > 0$ - происходит округление до n -й цифры после запятой;
- б) при $n = 0$ - округление до ближайшего целого;
- в) при $n < 0$ - округление до $(n+1)$ -й цифры влево от десятичной точки.

Блок-схема алгоритма округления десятичных чисел по заданному значению абсолютной погрешности приведена на рис. 1.

При отсутствии функции $ROUND$ она может быть смоделирована с помощью функции Int — «целая часть»:

$$ROUND(a, n) = \frac{sign(a)Int(|a|10^n + 0.5)}{10^n}$$

где $Sign$ — «знаковая функция», равная 1 для положительного значения аргумента, -1 для отрицательного и 0 для нулевого. Используя эту формулу, можно программным путем вести округление как по заданному уровню округления и, так и по заданной величине абсолютной погрешности.

На рис. 1 изображена блок-схема алгоритма округления числа a с предварительным определением уровня округления n по заданной величине абсолютной погрешности $\Delta(a^*)$.

При всей простоте этого алгоритма при машинном округлении могут возникать некоторые технические проблемы, часто не позволяющие получать достоверного результата. Коснемся здесь этих проблем, поскольку процедура округления десятичных чисел с последующим их выводом верными цифрами в естественной форме записи в дальнейшем будет применяться неоднократно. Ниже приведена программа *okruglenie*, которая реализует симметрическое округление десятичных чисел, оставляя лишь верные (в широком смысле) цифры в соответствии с заданной абсолютной погрешностью числа. Для вывода значения округленного числа в программе умышленно сохранена машинная (нормальная) форма представления десятичных чисел, без «усечения» неверных знаков.

В программе используется модуль *okrugl*, который, в свою очередь, обращается к процедуре *окр*. В основе этой процедуры лежит алгоритм, изображенный на рис. 1. Первый ее аргумент - подлежащее округлению число a , второй - абсолютная погрешность d . Результаты работы процедуры — параметры-переменные b и n ; первый из них - округленное число, второй показывает, на сколько знаков происходит округление (при $d > 1$ округление происходит до n -й цифры до десятичной точки, при $d < 1$ - до n -й цифры после десятичной точки). В ходе исполнения программы вначале определяется количество оставляемых в числе значащих цифр n (по разным схемам при $d < 1$ и $d > 1$) и затем выполняется округление по указанной выше формуле. В силу отсутствия в языке Pascal операции возведения в произвольную степень, для вычисления 10^n используется функция *stepen10*. Конструкция *type T* позволяет легко менять тип данных (делать это следует одновременно в модуле и в программе).

Нуждается в комментарии следующее различие между схемой на рис. 1 и программой: в циклах вместо условий $d < 10^{-n}$ и $d \geq 10^{-n}$ используются модифицированные условия $d < (10^{-n} + e)$ и $d \geq (10^{-n} + e)$ соответственно. Это связано с особенностями операций над вещественными числами: при вычислении 10^{-n} (при $n > 0$) в ЭВМ получается *приближенное* значение, и для правильной работы программы следует подстраховаться путем добавления e (точное значение e особой роли не играет, в программе положено $1 \cdot 10^{-10}$). Если этого не сделать, то при округлении числа 1.23456 при погрешностях 0.1 и 0.2 получится одинаковое округленное значение числа, в то время как оно в соответствии с правилом определения верных цифр должно быть разным.

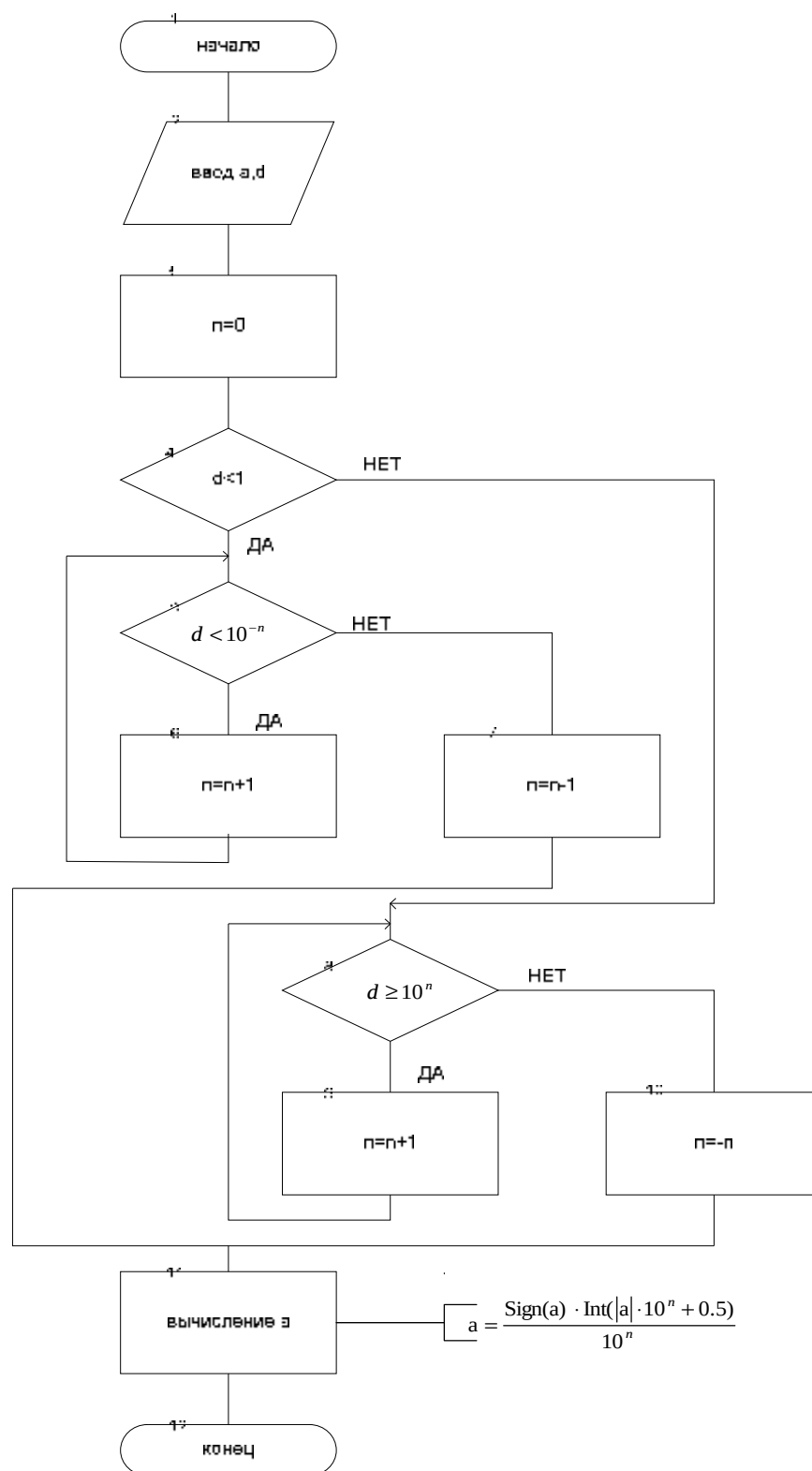


Рис. 1. Блок-схема алгоритма округления чисел по заданной величине абсолютной погрешности $\Delta(x^*)$.

Вывод значений уровня округления n позволяет, в частности, избежать ошибки в особых случаях - например, при округлении целых чисел, в изображении которых оказываются значащие нули, расположенные между значащими цифрами. При использовании программы `okruglenie`, когда в результате округления значащие и незначащие нули оказываются рядом и для их разделения используется значение n .

Выполним программное округление для числа $x = 2.5732$ с погрешностью $e = 0.0033$.

ПРОГРАММА ОКРУГЛЕНИЯ ЧИСЕЛ НА ЯЗЫКЕ BORLAND PASCAL

```
Program okruglenie;
uses okrugl,crt;
type T=real;
var a,d,b:T; n,m:integer;
begin
clrscr;
writeln;
writeln('введите число и погрешность'); read(a,d);
okr(a,d,a,n);
if a<0 then
begin
if n<0 then m:=0
else m:=n;
writeln('Округленное число a=',a-1E-17:17:m, ' n=',n);
end
else
begin
if n<0 then m:=0
else m:=n;
writeln('Округленное число a=',a+1E-17:17:m, ' n=',n);
end;
repeat until keypressed
end.
```

В программе используется модуль `okrugl`, который, в свою очередь, содержащий процедуру `okr`:

```
unit okrugl;
interface
type T=real;
procedure okr(a,d:T; var b:T; var n:integer);
implementation
function stepen 10(n:integer):T;
var i:integer; k:longint;
begin k:=10; for i:=1 to (abs(n)-1) do k:=k*10;
if n=0 then stepen 10:=1;
if n>0 then stepen 10:=k;
if n<0 then stepen 10:=1.0/k;
end;
procedure okr(a,d:T; var b:T; var n:integer);
var i,k:integer;
begin
```

```

n:=1;
if d<1 then
begin
while d<((1.0/stepen10(n))+1E-10)
do n:=n+1;
n:=n-1
end
else
begin
while d>=(stepen10(n)+1E-10)
do n:=n+1;
n:=-n
end;
b:=int((abs(a)*stepen10(n))+0.5)/stepen10(n);
if a <> 0 then b:=b*a/abs(a)
end;
end.

```

Результаты работы

введите число и погрешность

2.5732 0.003300

Округленное число a=2.57

n=2

3. Погрешности арифметических операций над приближенными числами

Исследуем влияние погрешностей исходных данных на погрешность результатов арифметических операций. Пусть a^* и b^* — приближенные значения чисел a и b . Какова соответствующая им величина неустранимой погрешности результата?

Утверждение 4. Абсолютная погрешность алгебраической суммы (суммы или разности) не превосходит суммы абсолютных погрешностей слагаемых, т.е.

$$\Delta(a^* \pm b^*) \leq \Delta(a^*) + \Delta(b^*), \quad (8)$$

Имеем

$$\Delta(a^* \pm b^*) = |(a \pm b) - (a^* \pm b^*)| = |(a - a^*) \pm (b - b^*)| \leq \Delta(a^*) + \Delta(b^*).$$

С л е д с т в и е . В силу неравенства (8) естественно положить

$$\bar{\Delta}(a^* \pm b^*) = \bar{\Delta}(a^*) + \bar{\Delta}(b^*). \quad (9)$$

Оценим относительную погрешность алгебраической суммы.

Утверждение 5. Пусть a и b — ненулевые числа одного знака. Тогда справедливы неравенства

$$\bar{d}(a^* + b^*) \leq d_{\max}, d(a^* - b^*) \leq n d_{\max}, \quad (10)$$

$$\text{где } d_{\max} = \max\{d(a^*), d(b^*)\}, n = |a + b| / |a - b|.$$

Используя формулу (2) и неравенство (8), имеем

$$|a \pm b| d(a^* \pm b^*) = \Delta(a^* \pm b^*) \leq \Delta(a^*) + \Delta(b^*) = |a| d(a^*) + |b| d(b^*) \leq (|a| + |b|) d_{\max} = |a + b| d_{\max}$$

Из полученного неравенства сразу следуют оценки (10).

Следствие. В силу неравенств (10) естественно положить

$$\bar{d}(a^* + b^*) = \bar{d}_{\max}, \bar{d}(a^* - b^*) = n \bar{d}_{\max}, \quad (11)$$

$$\text{где } \bar{d}_{\max} = \max\{\bar{d}(a^*), \bar{d}(b^*)\}, n = |a + b| / |a - b|.$$

Первое из равенств (11) означает, что при суммировании чисел одного знака не происходит потери точности, если оценивать точность в относительных единицах. При вычитании чисел одного знака граница относительной ошибки возрастает в $n > 1$ раз и возможна существенная потеря точности. Если числа a и b близки настолько, что $|a + b| \gg |a - b|$, то коэффициент $n \gg 1$ и не исключена полная или почти полная потеря точности. Когда это происходит, говорят о том, что произошла *катастрофическая потеря точности*.

Пример 15. Пусть решается инженерная задача, в которой окончательный результат y вычисляется по формуле $y = 1 - x$ с помощью предварительно определяемого значения x . Предположим, что найденное приближение $x^* = 0.999997$ к значению x содержит 6 верных значащих цифр. Тогда $y^* = 1 - 0.999997 = 0.000003$ и в процессе вычисления оказались потерянными 5 верных цифр. Если же учесть, что $d(x^*) \sim 0.0001\%$, а $d(y^*) \sim 33\%$, то следует признать, что произошла катастрофическая потеря точности.

Подчеркнем, что здесь виновником "катастрофы" является не операция вычитания, а предложенный метод решения задачи, где окончательный результат получается с помощью вычитания двух близких чисел. Выполнение этой операции лишь делает очевидным то, что действительно полезная информация о значении y уже оказалась потерянной до вычитания. Если нет другого варианта расчета, то для получения приемлемого результата следовало бы предварительно вычислить x с существенно большим числом верных знаков, учитывая, что 5 старших значащих цифр при вычитании будут потеряны.

Итак, получаем следующий важный вывод. При построении численного метода решения задачи следует избегать вычитания близких чисел одного знака. Если же такое вычитание неизбежно, то следует, вычислять аргументы с повышенной точностью, учитывая ее потерю примерно в $n = |a + b| / |a - b|$ раз.

Утверждение 6. Для относительных погрешностей произведения и частного приближенных чисел верны оценки

$$d(a^* b^*) \leq d(a^*) + d(b^*) + d(a^*) d(b^*), \quad (12)$$

$$d(a^* / b^*) \leq \frac{d(a^*) + d(b^*)}{1 - d(b^*)}, \quad (13)$$

в последней из которых считается, что $d(b^*) < 1$.

Выполним следующую цепочку преобразований:

$$\Delta(a^*b^*) = |ab|d(a^*b^*) = |ab - a^*b^*|,$$

$$|ab - a^*b^*| = |(a - a^*)b + (b - b^*)a - (a - a^*)(b - b^*)| \leq |b|\Delta(a^*) + |a|\Delta(b^*) + \Delta(a^*)\Delta(b^*)$$

$$|b|\Delta(a^*) + |a|\Delta(b^*) + \Delta(a^*)\Delta(b^*) = |ab|(d(a^*) + d(b^*) + d(a^*)d(b^*)),$$

т.е.

$$|ab|d(a^*b^*) \leq |ab|(d(a^*) + d(b^*) + d(a^*)d(b^*)).$$

Разделив обе части этого неравенства на $|ab|$, получаем оценку (12).

Для вывода второй оценки предварительно заметим, что

$$|b^*| = |b + (b^* - b)| \geq |b| - \Delta(b^*) = |b|(1 - d(b^*)). \text{ Тогда}$$

$$\begin{aligned} d(a^*/b^*) &= \frac{|a/b - a^*/b^*|}{|a/b|} = \frac{|ab^* - ba^*|}{|ab^*|} = \frac{|a(b^* - b) + b(a - a^*)|}{|ab^*|} \leq \\ &\leq \frac{|a|\Delta(b^*) + |b|\Delta(a^*)}{|ab|(1 - d(b^*))} = \frac{d(b^*) + d(a^*)}{1 - d(b^*)} \end{aligned}$$

С л е д с т в и е. Если $\bar{d}(a^*) \ll 1$ и $\bar{d}(a^*) \leq 1$, то для оценки границ относительных погрешностей можно использовать следующие приближенные равенства:

$$\bar{d}(a^*b^*) \approx \bar{d}(a^*) + \bar{d}(b^*), \quad \bar{d}(a^*/b^*) \approx \bar{d}(a^*) + \bar{d}(b^*) \quad (14)$$

Именно равенства (14) чаще всего и используют для практической оценки погрешности.

Итак, выполнение арифметических операций над приближенными числами, как правило, сопровождается потерей точности. Единственная операция, при которой потеря не происходит, - это сложение чисел одного знака. Наибольшая потеря точности может произойти при вычитании близких чисел одного знака.

Формулы для оценки абсолютных и относительных погрешностей арифметических операций: сложения, вычитания, умножения, деления, возведения в степень и извлечения корня приведены в таблице 2.

Пример 16. Вычислить абсолютные и относительные погрешности арифметических операций (сложения, умножения, деления, вычитания, возведения в степень и извлечения корня) с числами $a=0.7219$ и $b=135.347$, заданными всеми своими значащими цифрами в широком смысле системе программирования Borland Pascal и в Mathcad(e).

Таблица 2

Арифметические действия	Абсолютная погрешность	Относительная погрешность
$x + y$	$\Delta x + \Delta y$	$\frac{ x }{ x+y }dx + \frac{ y }{ x+y }dy$
$x - y$	$\Delta x + \Delta y$	$\frac{ x }{ x-y }dx + \frac{ y }{ x-y }dy$
$x*y$	$ x \Delta y + y \Delta x$	$dx + dy$
x/y	$\frac{ x \Delta y + y \Delta x}{y^2}$	$dx + dy$
x^n	$nx^{n-1}\Delta x$	ndx
$\sqrt[n]{x}$	$\frac{1}{nx^{\frac{n-1}{n}}}\Delta x$	$\frac{dx}{n}$

1) РЕШЕНИЕ В СИСТЕМЕ ПРОГРАММИРОВАНИЯ BORLAND PASCAL

```

uses crt;
var a,a1,da,d1a,b,db,d1b,c,dc,dc1,e,e1,e2,g,g1,g2,f,f1,f2,h,h1,h2,p,p1,p2:real;
function dsum(da,db:real):real;
begin
dsum:=da+db; end;
function dmult(a,da,b,db:real):real;
begin dmult:=abs(a)*db+abs(b)*da; end;
function dquot(a,da,b,db:real):real;
begin dquot:=(abs(a)*db+abs(b)*da)/sqr(b);
end;
function d1sum(a,d1a,b,d1b:real):real;
begin d1sum:=abs(a)*d1a/abs(a+b)+abs(b)*d1b/abs(a+b);
end;
function d1mult(d1a,d1b:real):real;
begin d1mult:=d1a+d1b; end;
begin
clrscr;
a:=0.7219;
da:=0.0001;
b:=135.147;
db:=0.001;
d1a:=da/a;
d1b:=db/b;
writeln('a=',a:1:8);
writeln('абсолютная погрешность a da=',da:1:8);
writeln('относительная погрешность a d1a=',d1a:1:8);
writeln('b=',b:1:8);
writeln('абсолютная погрешность b db=',db:1:8);

```

```

writeln('относительная погрешность b d1b=',d1b:1:8);
writeln;
c:=a*b;
writeln('произведение a*b=',c:1:8);
dc:=dmult(a,da,b,db);
writeln('абсолютная погрешность произведения d(a*b)=',dc:1:8);
dc1:=d1mult(d1a,d1b);
writeln('относительная погрешность d1(a*b)=',dc1:1:8);
writeln;
e:=a/b;
writeln('частное a/b=',e:1:8);
e1:=dquot(a,da,b,db);
writeln('абсолютная погрешность d(a/b)=',e1:1:8);
e2:=d1mult(d1a,d1b);
writeln('относительная погрешность частного d1(a/b)=',e2:1:8);
writeln;
f:=a+b;
writeln('сумма a+b=',f:1:8);
f1:=dsum(da,db);
writeln('абсолютная погрешность суммы d(a+b)=',f1:1:8);
f2:=d1sum(a,d1a,b,d1b);
writeln('относительная погрешность суммы d1(a+b)=',f2:1:8);
writeln;
g:=a-b;
writeln('разность a-b=',g:1:8);
g1:=dsum(da,db);
writeln('абсолютная погрешность разности d(a-b)=',g1:1:8);
g2:={g1/abs(g)}d1sum(a,d1a,-b,d1b);
writeln('относительная погрешность разности d1(a-b)=',g2:1:6);
writeln;
h:=a*a*a;
writeln('куб числа a a*a*a=',h:1:8);
h1:=3*a*a*da;
writeln('абсолютная погрешность куба числа a d(a*a)=',h1:1:8);
h2:=h1/h;
writeln('относительная погрешность куба числа a d1(a*a*a)=',h2:1:8);
writeln;
p:=sqrt(a);
writeln('корень квадратный sqrt(a)=',p:1:8);
p1:=1/(2*sqrt(a))*da;
writeln('абсолютная погрешность корня квадратного dsqrt(a) =',p1:1:8);
p2:=p1/p;
writeln('относительная погрешность корня квадратного dsqrt(a)=',p2:1:6);
writeln;
end.

```

Результаты работы программы

```

a=0.72190000
абсолютная погрешность a da=0.00010000
относительная погрешность a d1a=0.00013852
b=135.34700000
абсолютная погрешность b db=0.00100000

```

относительная погрешность b $d1b=0.00000740$

произведение $a*b=97.706999$

абсолютная погрешность произведения $d(a*b)=0.01423660$

относительная погрешность $d1(a*b)=0.00014592$

частное $a/b=0.0053337$

абсолютная погрешность $d(a/b)=0.00000078$

относительная погрешность частного $d1(a/b)=0.00014592$

сумма $a+b=136.06890000$

абсолютная погрешность суммы $d(a+b)=0.00110000$

относительная погрешность суммы $d1(a+b)=0.00000810$

разность $a-b=-134.62510000$

абсолютная погрешность разности $d(a-b)=0.00110000$

относительная погрешность разности $d1(a-b)=0.000008$

куб числа a $a*a*a=0.37621068$

абсолютная погрешность куба числа a $d(a*a)=0.00015634$

относительная погрешность куба числа a $d1(a*a*a)=0.00041557$

корень квадратный $\sqrt{a}=0.84964699$

абсолютная погрешность корня квадратного $d\sqrt{a}=0.00005885$

относительная погрешность корня квадратного $d\sqrt{a}=0.000069$

2) РЕШЕНИЕ В MATHCAD(e)

$$a := 0.7219$$

$$b := 135.34$$

$$\Delta a := 0.0001$$

$$\Delta b := 0.001$$

$$\delta a := \frac{\Delta a}{a} \quad \delta a = 0.000139$$

$$\delta b := \frac{\Delta b}{b} \quad \delta b = 0.000007$$

1) Сложение

$$c := a + b$$

$$c = 136.0689$$

$$\Delta c := \Delta a + \Delta b$$

$$\Delta c = 0.0011$$

$$\delta c := \frac{\Delta c}{c}$$

$$\delta c = 0.00000808$$

2) Вычитание

$$c1 := a - b$$

$$c1 = -134.6251$$

$$\Delta c1 := \Delta a + \Delta b$$

$$\Delta c1 = 0.0011$$

$$\delta c1 := \frac{\Delta c1}{|c1|}$$

$$\delta c1 = 0.00000817$$

3) Умножение

$$c2 := a \cdot b$$

$$c2 = 97.7070$$

$$\Delta c2 := |a| \cdot \Delta b + |b| \cdot \Delta a$$

$$\Delta c2 = 0.0143$$

$$\delta c2 := \frac{\Delta c2}{c2}$$

$$\delta c2 = 0.000146$$

4) Деление

$$c3 := \frac{a}{b}$$

$$c3 = 0.0053$$

$$\Delta c3 := \frac{|a| \cdot \Delta b + |b| \cdot \Delta a}{b^2}$$

$$\Delta c3 = 0.00000078$$

$$\delta c3 := \frac{\Delta c3}{c3}$$

$$\delta c3 = 0.00014591$$

5) Возведение в степень

$$c4 := a^3$$

$$c4 = 0.376$$

$$\Delta c4 := 3 \cdot a^2 \cdot \Delta a$$

$$\Delta c4 = 0.000156$$

$$\delta c4 := \frac{\Delta c4}{c4}$$

$$\delta c4 = 0.000416$$

6) Извлечение корня

$$c5 := \sqrt{a} \quad c5 = 0.8496$$

$$\Delta c5 := \frac{1}{2 \cdot \sqrt{a}} \cdot \Delta a \quad \Delta c5 = 0.000059$$

$$\delta c5 := \frac{\Delta c5}{c5} \quad \delta c5 = 0.000069$$

4. Погрешность функции

4.1. Погрешность функции многих переменных. Пусть $f(x) = f(x_1, x_2, \dots, x_m)$ дифференцируемая в области G функция m переменных, вычисление которой производится при приближенно заданных значениях аргументов $x^*_1, x^*_2, \dots, x^*_m$. Такая ситуация возникает, например, всякий раз, когда на ЭВМ производится расчет по формуле. Важно знать, какова величина неустранимой ошибки, вызванной тем, что вместо значения $y = f(x)$ в действительности вычисляется значение $y^* = f(x^*)$, где $x^* = (x^*_1, x^*_2, \dots, x^*_m)$.

Пусть $[x, x^*]$ - отрезок, соединяющий точки x и x^* .

Утверждение 7. Для абсолютной погрешности значения $y^* = f(x^*)$ справедлива следующая оценка:

$$\Delta(y^*) \leq \sum_{j=1}^m \max_{[x, x^*]} \left| \frac{\partial f}{\partial x_j} \right| \Delta(x_j^*) \quad (15)$$

Оценка (15) вытекает из формулы конечных приращений Лагранжа:

$$f(x) - f(x^*) = \sum_{j=1}^m \frac{\partial f(\xi)}{\partial x_j} (x_j - x_j^*), \quad \xi \in [x, x^*]$$

С л е д с т в и е. Если $x^* \approx x$, то в силу оценки (15) можно положить

$$\bar{\Delta}(y^*) \approx \sum_{j=1}^m \left| \frac{\partial f}{\partial x_j}(x^*) \right| \bar{\Delta}(x_j^*), \quad (16)$$

$$\bar{\Delta}(y^*) \approx \sum_{j=1}^m \left| \frac{\partial f}{\partial x_j}(x) \right| \bar{\Delta}(x_j^*), \quad (17)$$

Равенство (16) удобно для практических оценок, а равенством (17) мы воспользуемся в дальнейшем для теоретических построений.

Из формул (16), (17) вытекают приближенные равенства для оценки границ относительных погрешностей:

$$\bar{d}(y^*) \approx \sum_{j=1}^m n_j^* \bar{d}(x_j^*), \quad \bar{d}(y^*) \approx \sum_{j=1}^m n_j \bar{d}(x_j^*), \quad (18)$$

Здесь

$$n_j^* = \frac{\left| x_j^* \right| \left| \frac{\partial f(x^*)}{\partial x_j} \right|}{\left| f(x^*) \right|}, n_j = \frac{\left| x_j \right| \left| \frac{\partial f(x)}{\partial x_j} \right|}{\left| f(x) \right|} \quad (19)$$

Пример 17. Найти абсолютную и относительную погрешность вычислений объёма цилиндра по формуле $V = \frac{p d^2}{4} h$, если $d = 4,5 \pm 0,05$ см., $h = 10 \pm 0,08$ см., $p^* = 3,14$.

Решение. Будем считать точным значением числа p его приближение, взятое с точностью до 6 знаков $p = 3,141592$. Тогда абсолютная погрешность $\Delta p = p - p^* = 0,001592$. Рассматривая d, h, p как переменные величины, находим частные производные

$$\frac{\partial V}{\partial d} = \frac{p d h}{2} = \frac{3,14 \cdot 4,5 \cdot 10}{2} = 70,69;$$

$$\frac{\partial V}{\partial h} = \frac{p d^2}{4} = \frac{p \cdot 4,5^2}{4} = 15,896;$$

$$\frac{\partial V}{\partial p} = \frac{h d^2}{4} = \frac{10 \cdot 4,5^2}{4} = 50,625.$$

По формуле вычисления абсолютной погрешности функции трех переменных

$$\Delta V = \left| \frac{\partial V}{\partial d} \right| \Delta d + \left| \frac{\partial V}{\partial h} \right| \Delta h + \left| \frac{\partial V}{\partial p} \right| \Delta p$$

находим абсолютную погрешность объёма

$$\Delta V = \left| \frac{\partial V}{\partial d} \right| \Delta d + \left| \frac{\partial V}{\partial h} \right| \Delta h + \left| \frac{\partial V}{\partial p} \right| \Delta p = 70,69 \cdot 0,05 + 15,896 \cdot 0,08 + 50,625 \cdot 0,001592 = 4,887 \approx 4,89$$

Тогда $V = \frac{\pi d^2 h}{4} \approx (158,96 \pm 4,89) \text{ мл}^3$.

Относительная погрешность определения объёма

$$dV = \frac{4,887}{158,96} = 0,031 = 3,1\%$$

Выполним эти вычисления в MATHCAD(e).

$$v(d, h, \pi) := \frac{\pi \cdot d^2 \cdot h}{4}$$

Находим частные производные

$$\frac{d}{dd} v(d, h, \pi) \rightarrow \frac{1}{2} \cdot \pi \cdot d \cdot h \quad v1(d, h, \pi) := \frac{1}{2} \cdot \pi \cdot d \cdot h$$

$$\frac{d}{dh} v(d, h, \pi) \rightarrow \frac{1}{4} \cdot \pi \cdot d^2 \quad v2(d, h, \pi) := \frac{1}{4} \cdot \pi \cdot d^2$$

$$\frac{d}{d\pi} v(d, h, \pi) \rightarrow \frac{1}{4} \cdot d^2 \cdot h \quad v3(d, h, \pi) := \frac{1}{4} \cdot d^2 \cdot h$$

Будем считать точным значением числа π его приближение, взятое с точностью до 6 знаков после запятой $\pi = 3.141592$. Абсолютная погрешность числа $\Delta p = p - p_a = 0.001592$. Абсолютные погрешности $\Delta d = 0.05$; $\Delta h = 0.08$.

$$d := 4.5 \quad h := 10 \quad \pi\alpha := 3.14$$

$$\Delta d := 0.05 \quad \Delta h := 0.05 \quad \Delta \pi\alpha := 0.00159$$

$$v := \frac{\pi d^2 h}{4} \quad v = 159.043$$

Вычислим значения частных производных

$$v_1(d, h, \pi) = 70.686 \quad v_2(d, h, \pi) = 15.904 \quad v_3(d, h, \pi) = 50.625$$

Абсолютная погрешность вычисления объема

$$\Delta v := |v_1(d, h, \pi)| \cdot \Delta d + |v_2(d, h, \pi)| \cdot \Delta h + |v_3(d, h, \pi)| \cdot \Delta \pi\alpha$$

$$\Delta v = 4.887 \text{ см}^3$$

Тогда

$$v \approx 159.043 \pm 4.887$$

Относительная погрешность вычисления объёма

$$\delta v := \frac{\Delta v}{v} \quad \delta v = 0.031 \quad \delta v = 0.031 = 3.1\%$$

4.2. Погрешность функции одной переменной. Формулы для границ погрешностей функции $f(x)$ одной переменной являются частным случаем формул (16) — (18) при $m = 1$:

$$\bar{\Delta}(y^*) \approx |f'(x^*)| \bar{\Delta}(x^*), \quad \bar{\Delta}(y^*) \approx |f'(x)| \bar{\Delta}(x^*), \quad (20)$$

$$\bar{d}(y^*) \approx n^* \bar{d}(x^*), \quad \bar{d}(y^*) \approx n \bar{d}(x^*), \quad (21)$$

где $n^* = \frac{|x^*| |f'(x^*)|}{|f(x^*)|}$, $n = \frac{|x| |f'(x)|}{|f(x)|}$.

Формулы расчета абсолютной и относительной погрешностей основных элементарных функций приведены в таблице 3.

Таблица 3

$f(x)$	$\Delta f(x)$	$df(x)$
$\sqrt{x}$	$\frac{\Delta x}{2\sqrt{x}}$	$\frac{1}{2} dx$
$\frac{1}{x}$	$\frac{\Delta x}{x^2}$	dx
$\sin x$	$ \cos x \cdot \Delta x$	$ x \operatorname{ctg} x \cdot dx$
$\cos x$	$ \sin x \cdot \Delta x$	$ x \operatorname{tg} x \cdot dx$
$\operatorname{tg} x$	$\frac{\Delta x}{\cos^2 x}$	$\frac{2 x }{ \sin 2x } \cdot dx$
$\operatorname{ctg} x$	$\frac{ \Delta x }{ \sin^2 x }$	$\frac{2 x }{ \sin 2x } \cdot dx$
$\ln x$	$\frac{\Delta x}{x}$	$\frac{dx}{ \ln x }$
$\lg x$	$\frac{\Delta x}{x \ln 10}$	$\frac{dx}{ \ln x \cdot \ln 10}$
e^x	$e^x \Delta x$	$ x \cdot dx$
10^x	$10^x \ln 10 \cdot \Delta x$	$\ln 10 \cdot x \cdot dx$
a^x	$a^x \ln a \cdot \Delta x$	$\ln a \cdot x \cdot dx$
$\arcsin x$	$\frac{\Delta x}{\sqrt{1-x^2}}$	$\frac{ x }{ \arcsin x \sqrt{1-x^2}} \cdot dx$
$\arccos x$	$\frac{\Delta x}{\sqrt{1-x^2}}$	$\frac{ x }{ \arccos x \sqrt{1-x^2}} \cdot dx$
$\operatorname{arctg} x$	$\frac{\Delta x}{1+x^2}$	$\frac{ x }{ \operatorname{arctg} x (1+x^2)} \cdot dx$

arcctg x	$\left \frac{1}{1+x^2} \right \Delta x$	$\frac{ x }{ \text{arcctg } x \sqrt{1+x^2}} \cdot dx$
sh x	$ chx \cdot \Delta x$	$ cth x x \cdot dx$
ch x	$ shx \cdot \Delta x$	$ th x x \cdot dx$
th x	$\frac{\Delta x}{ch^2 x}$	$\frac{2 x }{ sh2x } \cdot dx$
cth x	$\left \frac{1}{sh^2 x} \right \cdot \Delta x$	$\frac{2 x }{ ch2x } \cdot dx$
x^n	$ nx^{n-1} \cdot \Delta x$	$n \cdot dx$

Пример 18. Оценить абсолютную и относительную погрешность вычисления функции $y = \sqrt{x-1} \cdot \cos(x)$ при $x^* = 2.5732$, заданного всеми своими верными значащими цифрами в строгом смысле, в системе программирования Borland Pascal и в Mathcad(e). При этом абсолютная погрешность оценивается как с использованием формулы $\Delta f = |f'(x)| \Delta x$, так и с использованием приращения функции.

1) ПРОГРАММА ВЫЧИСЛЕНИЯ АБСОЛЮТНОЙ И ОТНОСИТЕЛЬНОЙ ПОГРЕШНОСТЕЙ ФУНКЦИИ В СИСТЕМЕ ПРОГРАММИРОВАНИЯ BORLAND PASCAL.

```

program funksiya;
uses crt;
var x,x0,x1,dx,dx1,y0,y1,dyabs,dyotn,dyabs1,dyotn1,dxotn:real;
function y(x:real):real;
begin
y:=sqrt(x-1)*cos(x);
end;
begin
clrscr;
x0:=2.5732;
x1:=2.573;
dx:=0.0001/2;
y0:=y(x0);
writeln('x0=',x0:11:6,' ',y(x0)=' ',y0:11:6);
y1:=y(x1);
writeln('x1=',x1:11:6,' ',y(x1)=' ',y1:11:6);
dx1:=abs(x1-x0)+dx;
writeln(' абсолютная погрешность x: dx1=',dx1:11:6);

```

```

dxotn:=(dx1)/x1;
writeln('относительная погрешность x=',dxotn:11:6);
dyabs1:=abs(1/(2*sqrt(x0-1))*cos(x0)-sin(x0)*sqrt(x0-1))*dx1;
writeln('абсолютная погрешность функции (1способ) dyabs1=',dyabs1:11:6);
dyotn1:=dyabs1/abs(y0);
writeln('относительная погрешность функции (1 способ) dyotn1=',dyotn1:11:6);
dyabs:=abs(y0-y1);
writeln('абсолютная погрешность функции (2 способ) dyabs =', dyabs:11:6);
dyotn:=dyabs/abs(y1);
writeln('относительная погрешность функции (2 способ) dyotn =', dyotn:11:6);
repeat until keypressed;
end.

```

Результаты работы.

```

x0=2.573200   y(x0)= -1.057060
x1=2.573000   y(x1)= -1.0569
абсолютная погрешность x: dx1=0.000250
относительная погрешность x= 0.00009716
абсолютная погрешность функции (1способ) dyabs = 0.000253
относительная погрешность функции (1 способ) dyotn1= 0.000239
абсолютная погрешность функции (2 способ) dyabs = 0.000202
относительная погрешность функции (2 способ) dyotn = 0.000191

```

2) РЕШЕНИЕ В MATHCAD(E)

Исходная функция: $y(x) := \sqrt{x-1} * \cos(x)$

$x := 2.5732$

округлим число до 4 значащих цифр: $x1 := 2.573$

найдем абсолютную погрешность: $\Delta x := \frac{0.0001}{2}$ $\Delta x = 0.00005$

найдем погрешность округления: $xokr := |x - x1|$ $xokr = 0.0002$

Погрешность округления числа равна сумме погрешности исходного числа и погрешности округления:

$\Delta x1 := xokr + \Delta x$ $\Delta x1 = 0.00025$

найдем относительную погрешность x: $dx := \frac{\Delta x1}{x1}$ $dx = 0.00009716$

найдем абсолютную погрешность функции: $\Delta y(x) := \left| \frac{d}{dx} y(x) \right| \Delta x1$ $\Delta y(x) = 0.000253$

найдем относительную погрешность функции: $dy(x) := \frac{\Delta y(x)}{|y(x)|}$ $dy(x) = 0.000239$

$y(x) = -1.05706$

$$y(x_1) = -1.0569$$

Другой способ нахождения абсолютной и относительной погрешности:

$$\Delta y := |y(x) - y(x_1)| \quad \Delta y = 0.000202$$

$$d y_1 := \frac{\Delta y}{|y(x_1)|} \quad d y_1 = 0.000191$$

4.3. Погрешность неявной функции. Часто функция $y = f(x_1, x_2, \dots, x_m)$ задается не явной формулой, а как решение нелинейного уравнения $F(y, x_1, x_2, \dots, x_m) = 0$, т.е. неявно. Если для такой неявной функции воспользоваться известными формулами вычисления производных

$$\frac{\partial f}{\partial x_j} = \left(- \frac{\frac{\partial F}{\partial x_j}}{\frac{\partial F}{\partial y}} \right) \Big|_{y=f(x)}, \quad j = 1, 2, \dots, m, \quad (22)$$

то исследование неустранимой погрешности неявной функции сразу же сводится к рассмотренному выше случаю.

5. Особенности машинной арифметики

Знание основных особенностей машинной арифметики необходимо для грамотного использования ЭВМ при решении научно-технических задач. Пользователь, не учитывающий эти особенности, не может рассчитывать на высокую точность и эффективность вычислений. Невнимание к ним часто приводит к неверным результатам. Подчеркнем, что в основе причин появления вычислительной погрешности лежит сам способ представления чисел на ЭВМ.

5.1. Системы счисления. Принятый способ записи чисел состоит в представлении их упорядоченным набором цифр. В привычной нам десятичной позиционной системе счисления вещественное число x представляют последовательностью символов, которая начинается со знака (+ или -) и продолжается цепочкой десятичных цифр a_i и b_i , разделенных десятичной точкой:

$$x = \pm a_n \dots a_1 a_0 . b_1 b_2 \dots b_m \dots \quad (23)$$

Здесь каждой позиции (разряду), которую занимает цифра относительно десятичной точки, отвечает определенная степень числа 10. По существу, равенство (23) представляет собой принятое сокращение полной записи числа x , имеющей вид

$$x = \pm (a_n \cdot 10^n + \dots + a_1 \cdot 10^1 + a_0 \cdot 10^0 + b_1 \cdot 10^{-1} + b_2 \cdot 10^{-2} + \dots + b_m \cdot 10^{-m} + \dots).$$

Пример 19. Запись $x = 20.5$ означает, что $x = 2 \cdot 10^1 + 0 \cdot 10^0 + 5 \cdot 10^{-1}$

Для представления чисел в вычислительных машинах также используют позиционные системы счисления, однако основаниями систем служат, как правило, степени числа 2. Это вызвано способом хранения чисел в устройствах памяти ЭВМ, каждое из которых можно рассматривать как набор одноступенчатых

элементов, способных находиться только в одном из двух возможных устойчивых состояний - "включен" или "выключен". Эти состояния интерпретируются соответственно как 0 или 1 - значения двоичного числа. Наиболее распространены системы счисления с основанием 2 (базисная двоичная система счисления), 8 и 16.

С точностью до малосущественных деталей будем считать, что все вычислительные машины работают в двоичной системе счисления. В ней вещественное число x по-прежнему записывается в виде (23), однако a_i и b_i - уже двоичные цифры (0 или 1). В этом случае полная запись выглядит так:

$$x = \pm(a_n \cdot 2^n + \dots + a_1 \cdot 2^1 + a_0 \cdot 2^0 + b_1 \cdot 2^{-1} + b_2 \cdot 2^{-2} + \dots + b_m \cdot 2^{-m} + \dots)$$

Пример 20. Запишем число $x = 20.5$ в двоичной системе счисления. Для этого разложим его в сумму по степеням двойки: $x = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1}$. . Опуская степени двойки, получаем $x = (10100.1)_2$. Здесь нижний индекс 2 указывает на основание 2 системы счисления.

Для хранения числа в памяти ЭВМ отводится поле стандартной длины (*машинное слово*), в котором число записывают в виде последовательности двоичных цифр. По форме представления, способу хранения и реализации арифметических операций существенно различаются два типа используемых на вычислительных машинах чисел: целые числа и вещественные числа.

5.2. Представление целых чисел. Целое число n представляют в виде

$$x = \pm(a_L \cdot 2^L + \dots + a_1 \cdot 2^1 + a_0 \cdot 2^0) \quad (24)$$

где L - некоторое стандартное для ЭВМ целое число, a_i — двоичные цифры. Всего для хранения числа n отводят $L+2$ разрядов (один из них для хранения знака).

Из представления (24) видно, что максимальное по модулю целое число, представимое в ЭВМ, есть $n_{\max} = 2^L + \dots + 2^1 + 2^0 = 2^{L+1} - 1$. Обычно это значение не очень велико. Например, при стандартном формате записи целых чисел на компьютерах типа IBM, PC $n_{\max} = 2^{31} - 1 \approx 2 \cdot 10^9$.

Операции сложения, вычитания и умножения над целыми числами реализованы так, что если результат не превышает по модулю число $n_{\max}$, то он получается точным. Отметим, однако, следующую неприятную особенность. Если модуль результата превышает $n_{\max}$, то на большинстве вычислительных машин эта ситуация не доводится до сведения пользователя, происходит присвоение результату некоторого значения (меньшего $n_{\max}$ по модулю) и вычисления продолжают далее.

5.3. Представление вещественных чисел. В большинстве современных ЭВМ для вещественных чисел принята форма представления с плавающей точкой, когда каждое число представляют в виде

$$x = \pm(g_1 \cdot 2^{-1} + g_2 \cdot 2^{-2} + \dots + g_t \cdot 2^{-t}) 2^p. \quad (25)$$

Здесь $g_1, g_2, \dots, g_t$ — двоичные цифры. Число x нормализуется так, чтобы $g_1 = 1$, поэтому в памяти ЭВМ хранятся только значащие цифры. Число $m = \pm(g_1 \cdot 2^{-1} + g_2 \cdot 2^{-2} + \dots + g_t \cdot 2^{-t})$ называется *мантиссой* числа x . Количество t цифр, которое отводится для записи мантиссы, называемое *разрядностью* мантиссы, зависит от конструктивных особенностей конкретной вычислительной машины, но всегда является конечным. В представлении (25) p — целое число, называемое *двоичным порядком*. Порядок также записывают как двоичное целое число $p = \pm(s_1 s_{l-1} \dots s_0)_2$, для хранения которого в машинном слове отводится $l+2$ двоичных разрядов. На рис. 2 схематически представлена структура машинного слова для хранения вещественного числа.

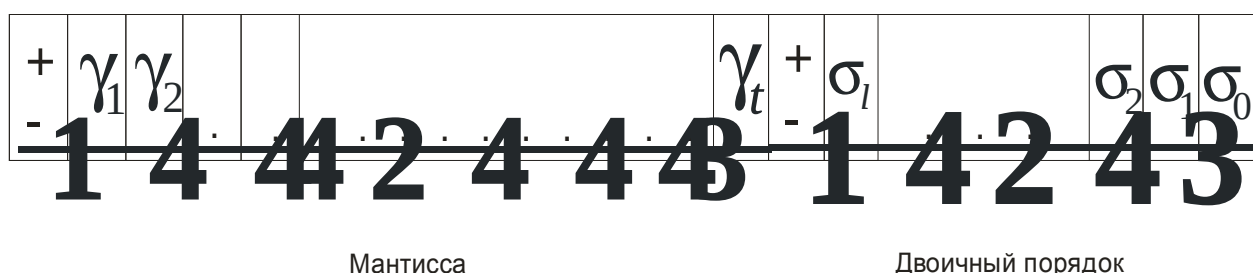


Рис. 2. Схематическое представление структуры машинного слова для хранения вещественного числа.

Поскольку нуль - ненормализуемое число (его нельзя представить в виде (25) при $g_1 \neq 0$), для его хранения предусматривают особый способ записи.

На основании имеющихся сведений о представлении чисел в ЭВМ можно сделать ряд важных выводов.

1°. На ЭВМ представимы не все числа, а лишь конечный набор рациональных чисел специального вида. Эти числа образуют *представимое множество* вычислительной машины. Для всех остальных чисел x возможно лишь их приближенное представление с ошибкой, которую принято называть *ошибкой представления* (или *ошибкой округления*). Обычно приближенное представление числа x в ЭВМ обозначают как $x^* = fl(x)$. Если округление производят по дополнению, то граница относительной погрешности представления равна единице первого отброшенного разряда мантиссы, т.е. $\bar{d}(x^*) = e_m = 2^{-t}$ (порядок числа не влияет на относительную погрешность представления).

Если же округление производят усечением, то $\bar{d}(x^*) = e_m = 2^{1-t}$. Величина e_m играет в вычислениях на ЭВМ фундаментальную роль; ее называют *относительной точностью ЭВМ*, а также *машинной точностью* (или *машинным эпсилон*). Всюду в дальнейшем e_m — это относительная точность ЭВМ. Заметим, что значение этой величины определяется разрядностью мантиссы и способом округления.

Важно с самого начала иметь четкое представление о том, что почти наверняка в представимом множестве ЭВМ нет числа y , являющегося решением поставленной задачи. Лучшее, что можно попытаться сделать, — это найти его представление $y^* = fl(x)$ с относительной точностью порядка e_m .

Полезно отметить, что среди представимых на ЭВМ чисел нет не только ни одного иррационального (в том числе и таких важных постоянных, как p , e , $\sqrt{2}$), но и даже такого широко используемого в вычислениях числа, как 0.1. Поэтому это число всегда представляется на ЭВМ приближенно, с погрешностью, вызванной необходимостью округления.

2°. Диапазон изменения чисел в ЭВМ ограничен. В самом деле, так как $g_1 = 1$, то для мантиссы m справедливы оценки $0.5 \leq |m| < 1$. В то же время для представления в ЭВМ порядка p используется конечное число $(l+1)$ двоичных цифр и поэтому $|p| \leq p_{\max} = 2^{l+1} - 1$. Таким образом, для всех представимых на ЭВМ чисел x (за исключением нуля) имеем $0 < X_0 \leq |x| < X_\infty$, где $X_0 = 2^{-(p_{\max}+1)}$, $X_\infty = 2^{p_{\max}}$. Заметим, что диапазон представления чисел на ЭВМ всецело определяется разрядностью порядка.

3°. Все числа x , по модулю большие X_∞ , не представимы на ЭВМ и могут рассматриваться как *машинная бесконечность*. Попытка получить такое число приводит к аварийному останову ЭВМ по *переполнению*. Все числа x , по модулю меньшие X_0 , для вычислительной машины не различимы и представляются как нуль (*машинный нуль*). Получение числа x такого, что $|x| < X_\infty$, называют *исчезновением порядка* (или *антипереполнением*). Обычно при исчезновении порядка автоматически полагается $fl(x) = 0$ и вычисления продолжаются.

З а м е ч а н и е . Не следует смешивать машинную точность e_M с минимальным положительным представимым на ЭВМ числом X_0 . Это совершенно разные числа, причем $X_0 \ll e_M$.



Рис. 3. Расположение чисел на машинной числовой оси

4°, На машинной числовой оси (рис. 3) числа расположены неравномерно. Плотность их возрастает по мере приближения к нулю и падает с удалением от нуля. Чтобы убедиться в этом, заметим, что расстояние от одного представимого на ЭВМ числа x до другого ближайшего представимого равно единице последнего разряда мантииссы, умноженной на 2^p , т.е. равно 2^{p-t} . Так как t фиксировано, то расстояние уменьшается с уменьшением порядка p и возрастает с увеличением p .

Для компьютеров типа IBM PC (при вычислениях на ФОРТРАНЕ) представимые числа расположены в диапазоне $10^{-38} < |x| < 3 \cdot 10^{38}$, вполне достаточном для большинства приложений. Однако разрядность мантииссы невелика ($t=23$) и e_M и $e_m \approx 10^{-7}$; в десятичной арифметике это эквивалентно тому, что мантиисса содержит 7 десятичных цифр. Заметим, что для большинства ЭВМ, используемых для научно-технических расчетов, мантиисса имеет разрядность, эквивалентную 7-14 десятичным разрядам.

5.3.1. Особенности представления чисел с фиксированной и плавающей запятой в памяти ЭВМ в различных форматах данных. В ЭВМ используются в основном два способа представления чисел: с естественным размещением (фиксированной) запятой и в форме с плавающей запятой. Характерной особенностью машинного представления чисел является конечность разрядной сетки, в которой хранятся числа в машине.

На рис. 4 изображена схема представления числа с зафиксированной запятой, или естественным размещением запятой, на примере 8-разрядного калькулятора. Десятичная запятая в этом случае может оказаться в любом цифровом разряде сетки.

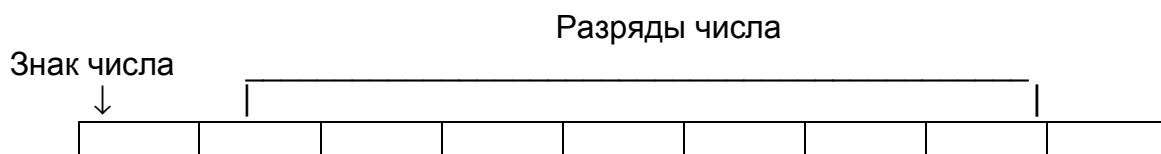


Рис. 4. Схема 8- разрядного числа с фиксированной запятой

В форме с естественной запятой 8-разрядная сетка может размещать лишь числа из диапазона $[-(1 \cdot 10^8 - 1); -1 \cdot 10^{-7}; +(1 \cdot 10^8 - 1)]$. Важно заметить, что это множество конечно. Оно покрывает числовую ось двумя участками равномерно

расположенных чисел (с шагом $1 \cdot 10^{-7}$) с границами от $-(1 \cdot 10^8 - 1)$ до $-1 \cdot 10^{-7}$ и от $1 \cdot 10^{-7}$ до $1 \cdot 10^8 - 1$ (рис.5).



Рис. 5. Множество чисел, представимых в ЭВМ в форме с фиксированной запятой

Числа из интервала $(-1 \cdot 10^{-7}; 1 \cdot 10^{-7})$ могут быть помещены в такой разрядной сетке (кроме нуля), так что если в результате произведенной операции получится любое ненулевое число из этого интервала, то в ЭВМ оно будет отождествлено с нулем. По этой причине числа из этого интервала называют машинным нулем. Не могут быть помещены в этой сетке и числа, большие по модулю, чем $1 \cdot 10^8 - 1$. При появлении в процессе счета таких чисел возникает ситуация, называемая переполнением разрядной сетки.

В форме с плавающей запятой число представляется в виде $x = \pm M \cdot 10^p$, где M - мантисса; p - порядок числа, т.е. показатель степени, в которую нужно возвести 10, чтобы, умножив результат на мантиссу и присвоив нужный знак, получить данное число (рис. 6).



Рис. 6. Пример разрядной сетки с представлением десятичных чисел в форме с плавающей запятой

Абсолютная величина порядка при представлении, изображенном на рис. 6, может выражаться максимальным двузначным числом 99. Это дает огромные диапазоны представления действительных чисел. Важной особенностью этой системы представления чисел является неравномерность их расположения на числовой оси.

В учебной литературе часто встречается представление мантиссы в виде $M = \frac{d_1}{b} + \frac{d_2}{b^2} + \dots + \frac{d_t}{b^t}$, где β - основание системы счисления; t - количество разрядов в записи мантиссы; $d_1, d_2, \dots, d_t$ - целые числа, удовлетворяющие неравенствам $0 \leq d_i \leq \beta - 1$ ($i = 1, 2, \dots, t$).

Определенная таким образом мантисса оказывается в диапазоне $0 < M < 1$. Значение порядка p берется из интервала $[p_1, p_2]$, границы которого обусловлены способом физического представления чисел в машине. Используя эти представления можно вывести, что в системе с плавающей запятой ровно $2(\beta - 1)^{t-1} (p_2 - p_1 + 1) + 1$ чисел.

Для хранения двоичных вещественных чисел в современных ЭВМ соответствующие форматы предписаны стандартом, не зависящим от архитектуры ЭВМ и конкретных языков программирования (IEEE 754 Floating Points Numbers).

Существует несколько типов вещественных величин. Характеристики некоторых из этих типов, используемых в языке Pascal, приведены в таблице 4.

Таблица 4

Тип	Длина записи числа, байт	Диапазон значений	Количество сохраняемых значащих цифр мантиссы в десятичном представлении числа
Real	6	$2.9 \cdot 10^{-39} - 1.7 \cdot 10^{38}$	11-12
Single	4	$1.5 \cdot 10^{-45} - 3.4 \cdot 10^{38}$	7-8
Double	8	$5 \cdot 10^{-324} - 1.7 \cdot 10^{308}$	15-16
Extended	10	$3.4 \cdot 10^{-4932} - 1.1 \cdot 10^{4932}$	19-20

Опишем форматы хранения чисел типов single и double. Отметим предварительно следующее обстоятельство. В указанном выше стандарте число хранится в такой форме, что его мантисса заключена в диапазоне $1 < M < 2$, т.е. первая цифра мантиссы всегда равна 1. Эта дополнительная операция сдвига на один разряд («нормализация») делает нормализованный порядок на единицу меньше, чем тот, который использован в предыдущем рассмотрении.

Величина типа single занимает в памяти ЭВМ 4 байта (рис. 7). Знак числа хранится в крайнем левом разряде с номером 31 (как обычно, разряды нумеруются с 0 и располагаются справа налево). Порядок в смещенной форме занимает разряды с 23 по 30. Смещенная форма находится так: к настоящему значению порядка прибавляется число 127 (так называемое «смещение») для того, чтобы значение смещенного порядка всегда было положительным. Мантисса занимает все остальные разряды, т.е. с 0 по 22, причем целая часть мантиссы вообще не хранится (она всегда равна единице и при вычислениях это учитывается).

Величина типа double занимает в памяти ЭВМ 8 байт (рис. 8). Знак числа хранится в разряде номер 63. Порядок в смещенной форме занимает разряды с 52 по 62 (смещение в данном случае равно 1023). Мантисса занимает разряды с 0 по 51 (как и раньше, целая часть мантиссы, автоматически равная единице, не хранится).

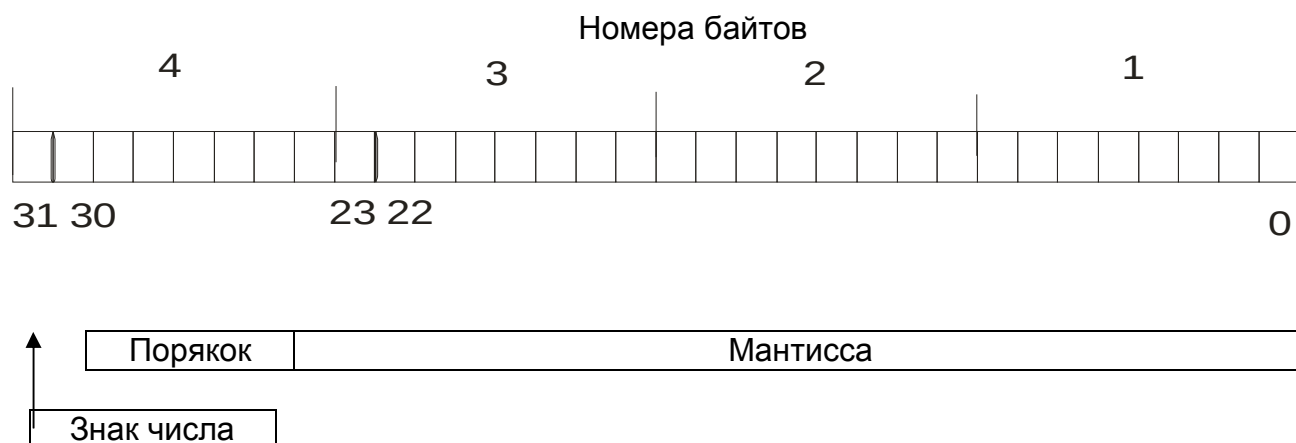


Рис. 7. Способ хранения числа типа single в ЭВМ

Опишем формат хранения числа типа double. Величина типа double занимает в памяти ЭВМ 8 байт (рис. 8). Знак числа хранится в разряде номер 63. Порядок в смещенной форме занимает разряды с 52 по 62 (смещение в данном случае равно

1023). Мантисса занимает разряды с 0 по 51, целая часть мантиссы автоматически равная единице не хранится.

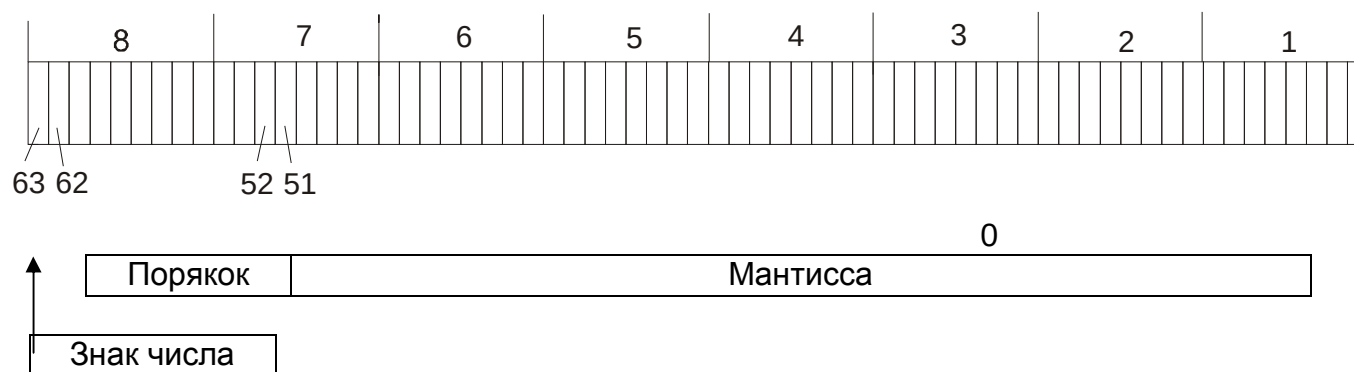


Рис. 8. Способ хранения числа типа double в ЭВМ.

5.3.2. Погрешности вычисления элементарных функций с учетом представления в ЭВМ вещественных чисел в различных форматах записи данных и особенностей процедуры вывода writeln в системе программирования Borland Pascal. Характеристики некоторых используемых в языке программирования Pascal типов вещественных величин приведены в таблице 1: длина записи в байтах, диапазон значений, количество сохраняемых значащих цифр мантиссы в десятичном представлении числа. Будем считать, что вычисляемое значение имеет наибольшую точность (что вполне естественно), Из таблицы 1 следует, что при использовании величины типа real в мантиссе лишь 12 верных значащих цифр, типа double - 17, типа single - 7 (верные цифры подчеркнуты, а сомнительные выделены жирным шрифтом). Возможности стандартной процедуры вывода в языке Borland Pascal writeln позволяют сохранить в мантиссе отображения лишь 18 значащих цифр. Поэтому при использовании типа extended не все верные цифры результата выводятся процедурой writeln. Поэтому в программе использован форматированный вывод с заведомо большим, чем необходимо, количеством значащих цифр, чтобы не потерять в отображении результата части цифр мантиссы.

Пример 21. Оценить точность вычисления элементарной функции $y = \sqrt{x-1} * \cos(x)$ в системе программирования BORLAND PASCAL в точке $x=2.5732$.

ПРОГРАММА ОЦЕНКИ ТОЧНОСТИ ВЫЧИСЛЕНИЯ ЭЛЕМЕНТАРНЫХ ФУНКЦИЙ В СИСТЕМЕ ПРОГРАММИРОВАНИЯ BORLAND PASCAL

```
program s;
uses crt;
var x,y: double;
begin
clrscr;
writeln('vvedite x');
read(x);
```

```
writeln('x=',x:6:5);
y:=sqrt(x-1)*cos(x);
writeln('sqrt(x-1)*cos(x)=' ,y:40);
repeat until keypressed
end.
```

РЕЗУЛЬТАТЫ РАБОТЫ

Vvedite x

2.5732

double: sqrt(x-1)*cos(x)=	<u>-1.05705996827185778E+0000</u>
single: sqrt(x-1)*cos(x)=	<u>-1.05706000328063965E +0000</u>
rea1: sqrt(x-1)*cos(x)=	<u>-1.05705996827055060E +0000</u>
extended: sqrt(x-1)*cos(x)=	<u>-1.05705996827185794E +0000</u>

Верные цифры мантисс подчеркнуты

Сомнительные цифры мантиссы выделены

5.4. Арифметические операции над числами с плавающей точкой.

Правила выполнения арифметических операций в двоичной системе счисления чрезвычайно просты и легко реализуются на ЭВМ.

Однако в силу ограниченной разрядности мантиссы операции сложения, вычитания, умножения и деления над представимыми в ЭВМ вещественными числами не могут быть реализованы точно. Дело в том, что арифметические операции над числами, мантиссы которых содержат t разрядов, приводят, как правило, к результатам, содержащим более t разрядов. Округление результата до t разрядов и служит главным источником погрешности. Для того чтобы отличать машинные арифметические операции от идеальных математических операций $+$, $-$, $\times$, $:$, будем обозначать их через $\oplus$, $\ominus$, $\otimes$, $\oslash$. Игнорируя несущественные детали, можно считать, что результат машинной арифметической операции совпадает с результатом точного выполнения той же операции с погрешностью, приближенно равной погрешности округления. Таким образом,

$$\overline{\Delta}(a \oplus b) \approx (a + b)e_m, \quad \overline{\Delta}(a \ominus b) \approx (a - b)e_m,$$

$$\overline{\Delta}(a \otimes b) \approx (a \times b)e_m, \quad \overline{\Delta}(a \oslash b) \approx (a : b)e_m$$

Конечно, в некоторых ситуациях округление может отсутствовать. Полезно знать, что умножение и деление числа на целую степень двойки выполняется на ЭВМ точно, так как в этом случае мантисса не меняется.

Машинные арифметические операции обладают иными свойствами в отличие от обычных математических операций. Например, не выполняется известное правило арифметики «от перемены мест слагаемых сумма не меняется».

5.5. Удвоенная точность. На многих вычислительных машинах возможна реализация арифметических действий над числами, разрядность мантисс которых примерно вдвое превосходит стандартную разрядность t , что приводит к существенному повышению машинной точности.

На тех ЭВМ, где арифметика удвоенной точности реализована аппаратно, время исполнения программ возрастает не более чем вдвое (часто этот коэффициент ближе к единице). Если же реализация вычислений с удвоенной мантиссой осуществляется программным образом, то время счета увеличивается в несколько раз. Поскольку для хранения числа с удвоенной мантиссой отводится два машинных слова, в этом случае вдвое возрастает используемая память.

Отметим, что удвоенная точность не ликвидирует ошибки округления, а только уменьшает их величину.

5.6.Вычисление машинного эпсилон. Для приближенного вычисления величины e_M удобно пользоваться следующим определением. Машинное эпсилон – это минимальное из представимых на ЭВМ чисел e , для которых $1 \oplus e > 1$.

В дальнейшем все конкретные числовые примеры мы будем рассматривать в десятичной арифметике, которая привычна для большинства людей, т.к. основные закономерности поведения ошибок округления не зависят от основания используемой системы счисления.

Одной из важных характеристик точности арифметики числа с плавающей запятой ЭВМ является значение машинного эпсилон, т.е. такого наименьшего числа e_i из системы чисел с плавающей запятой, что добавление его к единице дает на этой ЭВМ значение, большее единицы. Величина e_M играет в вычислениях на ЭВМ фундаментальную роль; ее называют *относительной точностью ЭВМ*, а также *машинной точностью* (или *машинным эпсилон*). Всюду в дальнейшем e_M — это относительная точность ЭВМ. Заметим, что значение этой величины определяется разрядностью мантииссы и способом округления.

Важно с самого начала иметь четкое представление о том, что почти наверняка в представимом множестве ЭВМ нет числа y , являющегося решением поставленной задачи. Лучшее, что можно попытаться сделать, — это найти его представление $y^* = fl(x)$ с относительной точностью порядка e_M .

Поскольку числа, по абсолютной величине меньшие $e_i X_0 = e_i 2^{-(p_{\max}+1)}$, где $p_{\max} = 2^{l+1} - 1$ - максимально возможный порядок числа с плавающей запятой, имеющий в разрядной сетке данной ЭВМ конечное число $(l+1)$ двоичных цифр, не входят в множество представимых чисел этой ЭВМ, то все эти числа автоматически заменяются нулём с абсолютной погрешностью $e_i 2^{-(p_{\max}+1)}$ и относительной погрешностью e_i . Числа, по абсолютной величине большие $e_i X_0 = e_i 2^{-(p_{\max}+1)}$, входят в множество представимых чисел этой ЭВМ, но они также представляются с ошибкой округления, вызванной отбрасыванием двоичных разрядов мантииссы, номера которых превышают номер двоичного разряда, соответствующего e_i . Поэтому важно исследовать зависимость машинного эпсилон e_i от типов вещественных величин. типов вещественных величин (форматов представления данных), приведенных в таблице 4.

Значение машинного эпсилон для конкретной ЭВМ можно определить программным путем. Полагая $e^{(0)} = 1$, следует вычислять последовательно $e^{(1)} = 0.5e^{(0)}$, $e^{(2)} = 0.5e^{(1)}$, ..., $e^{(n)} = 0.5e^{(n-1)}$, ..., проверяя каждый раз выполнение неравенства $1 \oplus e^{(n)} > 1$. Как только при некотором n окажется, что $1 \oplus e^{(n)} = 1$, следует положить $e_M = e^{(n-1)}$ перейти к следующему этапу вычислений. Хотя полученное таким способом значение может отличаться от e_M в 2 раза, обычно оно используется так, что эта погрешность не имеет значения. Блок-схема алгоритма представлена на рис. 9. Далее приводится программа нахождения машинного эпсилон e_M в различных форматах представления данных. В программе использован форматированный вывод eps:40 с заведомо большим, чем необходимо, количеством цифр, чтобы не потерять в отображении результата части цифр мантииссы. Показано, что $e_M = 5.42101086242752217E-0020$ одинаково во всех форматах представления вещественных чисел в языке Pascal single, rea1, double, extended.

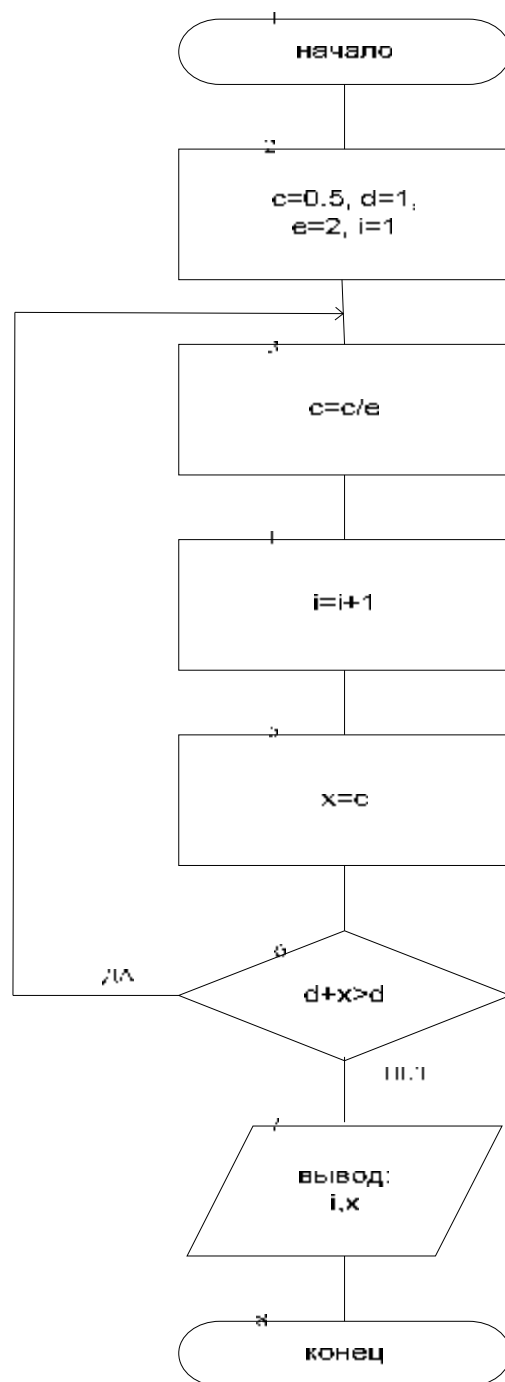


Рис. 9. Блок-схема алгоритма нахождения машинного эпсилон

ПРОГРАММА НАХОЖДЕНИЯ МАШИННОГО ЭПСИЛОН В СИСТЕМЕ ПРОГРАММИРОВАНИЯ BORLAND PASCAL.

```
program epsilon;  
uses crt;  
const c:single=0.5; d:single=1; e:single=2;  
var a,eps:single; i:integer;  
begin  
  clrscr;  
  a:=d; eps:=c; i:=1;  
  while a+eps>a do  
  begin  
    eps:=eps/e; i:=i+1;  
  end;  
  writeln('i=',i,'eps:single, mashinnoe epsilon=',eps:40);  
  repeat until keypressed  
end.
```

i=64eps: single, машинное эпсилон=5.42101086242752217E-0020

Это число одинаково во всех форматах представления данных.

5.7. Универсальная формула оценки погрешности вычислений. В практике вычислений часто используется универсальная формула для меры погрешности вычисления приближенного числа a^*

$$\Delta = \frac{|a - a^*|}{1 + |a|},$$

которая объединяет в себе черты абсолютной и относительной погрешности. Она близка к $\Delta(a^*)$ при $|a| \ll 1$ и практически совпадает с $d(a^*)$ при $|a| \gg 1$.

6. ПРИБЛИЖЕННЫЕ ВЫЧИСЛЕНИЯ ПО ФОРМУЛЕ

Наиболее распространенный вид вычислений – это вычисления по готовой формуле. В компьютере вычисление при любой громоздкости формулы обеспечивается, как правило, одной командой (оператором). Если при этом программно не предусматривается контроль за вычислительными погрешностями, то вычислитель анализирует результат в конце счета.

Рассмотрим вычисления по правилам подсчета цифр, вычисления со строгим учетом предельных абсолютных погрешностей, вычисления по методу границ. При этом часто приходится применять пооперационный подход, который позволяет исключать сомнительные цифры в результатах последующих вычислений.

Пример 22. Вычислить значение величины Z по формуле $z = \frac{\ln(b) - a}{a^2 + 12 * c}$ при заданных значениях параметров $a=0.7219$, $b=135.347$, $c=0.013$ следующими методами:

6.1. вычислениями по правилам подсчета цифр;
6.2. вычислениями со строгим учетом предельных абсолютных погрешностей;
6.3. вычислениями по методу границ
с пошаговой и итоговой регистрацией результатов вычислений, составляя программу в системе программирования Borland Pascal и используя Mathcad. Числа a , b , c заданы всеми своими верными значащими цифрами в широком смысле.

6.1. Вычисления по правилам подсчета цифр. При вычислении этим методом явного учета погрешностей не ведется, правила подсчета цифр показывают лишь, какое количество значащих цифр или десятичных знаков в результате можно считать надежными.

Сами эти правила основываются на выводах, вытекающих из формул для оценки погрешностей арифметических действий и функций. Приведем эти правила в систематизированном виде.

1. При сложении и вычитании приближенных чисел младший из сохраняемых десятичных разрядов результата должен являться наибольшим среди десятичных разрядов, выражаемых последними верными значащими цифрами исходных данных.

Следует избегать вычитания близких по величине чисел, а так же при пооперационном применении правила для сложения и вычитания нескольких чисел подряд стараться производить действия над числами в порядке возрастания их абсолютных величин.

2. При умножении и делении приближенных чисел нужно выбрать число с наименьшим количеством значащих цифр и округлить остальные числа так, чтобы в них было лишь на одну значащую цифру больше, чем в наименее точно числе.

В результате следует считать верными столько значащих цифр, сколько их в числе с наименьшим количеством значащих цифр.

3. При определении количества верных цифр в значениях элементарных функций от приближенных значений аргумента следует грубо оценить значение модуля производной функции. Если это значение не превосходит единицы или близко к ней, то в значении функции можно считать верными столько знаков после запятой, сколько их имеет значение аргумента. Если же модуль производной функции в окрестности приближенного значения аргумента превосходит единицу, то количество верных десятичных знаков в значении функции меньше, чем в значении аргумента на величину k , где k – наименьший показатель степени, при котором имеет место $|f'(x)| < 10^k$.

4. При записи промежуточных результатов следует сохранять на одну цифру больше, чем рекомендуют правила 1-3. В окончательном результате это запасная цифра округляется.

Правила подсчета цифр носят оценочный характер и не являются методом строгого учета точности вычислений. Обычно их применяют тогда, когда быстро и без особых затрат нужно получить результат, не особо беспокоясь о его достоверности. Между тем практическая надежность этих правил достаточно высока в результате вычислительной вероятности взаимопогашения ошибок, не учитываемой при строгом подсчете предельных погрешностей.

При операционном учете ошибок вычислений используется обычная расчетная таблица – так называемая *расписка формулы*.

Для заданной формулы и исходных данных выполним в MATHCAD(e) расчет по правилам подсчета цифр.

ВЫЧИСЛЕНИЯ ПО ПРАВИЛАМ ПОДСЧЕТА ЦИФР ПО ФОРМУЛЕ

$$z = \frac{\ln(b) - a}{a^2 + 12 \cdot c}$$

$$a := 0.721\}$$

$$b := 135.34\}$$

$$c := 0.01\}$$

$$\Delta a := 0.0001$$

$$\Delta b := 0.001$$

$$\Delta c := 0.001$$

Вычисления проводим в следующей последовательности:

1) вычисляем $\ln(b)$ $\frac{d}{db} \ln(b) \rightarrow 7.388416440704263855 \cdot 10^{-3}$

$$7.388416440704263855 \cdot 10^{-3} < 1$$

В значении функции $\ln(b)$ берем 3 знака после запятой, как в числе b

$$\ln(b) = 4.908$$

2) вычисляем $d = \ln(b) - a$

$$d := \ln(b) - a$$

$$d = 4.186$$

В значении числа d берем 3 знака после запятой, как и в функции $\ln(b)$, поскольку при сложении и вычитании приближенных чисел младший из сохраняемых десятичных разрядов результата должен являться наибольшим среди десятичных разрядов, выражаемых последними верными значащими цифрами исходных данных.

3) вычисляем $e = a^2$

$$e := a^2$$

$$\frac{d}{da} a^2 \rightarrow 1.443\}$$

$$1.4438 > 1$$

В значении функции $e = a^2$ берем 3 знака после запятой, то есть на одну десятичную цифру меньше, чем в числе a

$$e = 0.521$$

4) вычисляем $f = 12 \cdot c$

$$f := 12 \cdot c$$

$$f = 0.156$$

В значении числа f берем 3 знака после запятой, как и в в числе c

5) вычисляем $g = e + f = a^2 + 12 \cdot c$

$$g := e + f$$

$$g = 0.677$$

В значении числа g берем 3 знака после запятой, как и в в числах e и f

6) вычисляем $z = \frac{d}{g} = \frac{\ln(b) - a}{a^2 + 12 \cdot c}$

$$z := \frac{d}{g} \quad z = 6.182$$

В значении числа z берем 3 знака после запятой, как и в числах d и g

Окончательно $z = 6.182$

6.2. Вычисления со строгим учетом предельных абсолютных погрешностей.

Этот метод предусматривает использование правил вычислений предельных абсолютных погрешностей результатов выполнения арифметических действий и формул оценки погрешности значений функций, изложенных в разделах 3,4. При пооперационном учете погрешностей выполняются пошаговые вычисления со строгим учетом предельных абсолютных погрешностей каждой операции. Промежуточные результаты округляются до одной запасной цифры. Значение погрешностей округляются с возрастанием до двух значащих цифр.

1) РЕШЕНИЕ В СИСТЕМЕ ПРОГРАММИРОВАНИЯ BORLAND PASCAL

```
uses crt;
var a,da,b,db,c,dc,z,z0,dz,z1,d,dd,d1,e,e1,de,f,f1,df,g,g1,dg,h,h1,dh:real;
begin
  clrscr;
  a:=0.7219; da:=0.0001;
  b:=135.347; db:=0.001;
  c:=0.013; dc:=0.001;
  writeln('Вычисление по формуле Z=(ln(b)-a)/(a*a+12*c) методом ');
  writeln('строгого учета предельных абсолютных погрешностей');
  writeln('a=',a:8:4,' da=',da:8:4);
  writeln('b=',b:8:4,' db=',db:8:4);
  writeln('c=',c:8:4,' dc=',dc:8:4);
  d:=ln(b); dd:=db/b;
  writeln('d=',d:1:6,' dd=',dd:1:7);
  write('Введите округленное значение d d1=');read(d1);
  e:=d1-a;de:=dd+da;
  writeln('e=',e:1:6,' de=',de:1:7);
  write('Введите округленное значение e e1=');read(e1);
  f:=a*a;df:=2*a*da;
  writeln('f=',f:1:6,' df=',df:1:7);
  write('Введите округленное значение f f1=');read(f1);
  g:=12*c;dg:=12*dc;
  writeln('g=',g:1:6,' dg=',dg:1:7);
  write('Введите округленное значение g g1=');read(g1);
  h:=f1+g1;dh:=df+dg;
  writeln('h=',h:1:6,' dh=',dh:1:7);
```

```

write('Введите округленное значение h h1=');read(h1);
z:=e1/h1;dz:=(e1*dh+h1*de)/sqr(h1);
writeln('z=',z:1:6,' dz=',dz:1:7);
write('Введите округленное значение z z1=');read(z1);
z0:=(ln(b)-a)/(a*a+12*c);
writeln('Точное значение z=',z0:1:6);
end.

```

Вычисление по формуле $Z=(\ln(b)-a)/(a^2+12c)$ методом строгого учета предельных абсолютных погрешностей

```

a= 0.7219 da= 0.0001
b=135.3470 db= 0.0010
c= 0.0130 dc= 0.0010
d=4.907842 dd=0.0000074
Введите округленное значение d d1=4.90784
e=4.185940 de=0.0001074
Введите округленное значение e e1=4.1859
f=0.521140 df=0.0001444
Введите округленное значение f f1=0.5211
g=0.156000 dg=0.0120000
Введите округленное значение g g1=0.15
h=0.671100 dh=0.0121444

```

```

Введите округленное значение h h1=0.67
z=6.247612 dz=0.1134041
Введите округленное значение z z1=6.2
Точное значение z=6.181800

```

2) РЕШЕНИЕ В MATHCAD(e).

Вычисление со строгим учетом предельных абсолютных погрешностей

по формуле $z = \frac{\ln(b) - a}{a^2 + 12c}$

```

a := 0.7219
b := 135.34
c := 0.013

```

```

Δa := 0.0001
Δb := 0.001
Δc := 0.001

```

```

d := ln(b)
d = 4.907842
Δd := Δb/b
Δd = 0.0000074

```

Округляем число $d = 4.907842$ по абсолютной погрешности $\Delta d = 0.0000074$ в широком смысле $d1=4.90784$

```

d1 := 4.9078

```

```

e := d1 - a
e = 4.185940
Δe := Δd + Δa
Δe = 0.0001074

```

Округляем число $e = 4.185940$ по абсолютной погрешности $\Delta e = 0.0001074$ в широком смысле $e1=4.1859$

$$e1 := 4.1859$$

$$f := a^2 \quad f = 0.521140 \quad \Delta f := 2 \cdot a \cdot \Delta a \quad \Delta f = 0.0001444$$

Округляем число $f = 0.521140$ по абсолютной погрешности $\Delta f = 0.0001444$ в широком смысле $f1=0.5211$

$$f1 := 0.5211$$

$$g := 12 \cdot c \quad g = 0.156000 \quad \Delta g := 12 \cdot \Delta c \quad \Delta g = 0.012$$

Округляем число $g = 0.156000$ по абсолютной погрешности $\Delta g = 0.0120000$ в широком смысле $g1=0.15$

$$g1 := 0.15$$

$$h := f1 + g1 \quad h = 0.6711 \quad \Delta h := \Delta f + \Delta g \quad \Delta h = 0.0121444$$

Округляем число $h = 0.671100$ по абсолютной погрешности $\Delta h = 0.0121444$ в широком смысле $h1=0.67$

$$h1 := 0.67$$

$$z := \frac{e1}{h1} \quad z = 6.248 \quad \Delta z := \frac{e1 \cdot \Delta h + h1 \cdot \Delta e}{h1^2} \quad \Delta z = 0.113404$$

Округляем число $z = 6.247612$ по абсолютной погрешности $\Delta z = 0.1134041$ в широком смысле $z1=6.2$

$$z1 := 6.2$$

$$z := \frac{\ln(b) - a}{a^2 + 12 \cdot c} \quad z = 6.181800$$

6.3. Вычисления по методу границ. Если нужно иметь абсолютно гарантированные границы возможных значений вычисляемой величины, используют специальный метод вычислений – метод границ.

Пусть $z=f(x,y)$ – функция непрерывная и монотонная в некоторой области допустимых значений аргументов x и y . Нужно получить её значения $f(a,b)$, где a и b – приближенные значения аргументов, причем достоверно известно, что $a_1 < a < a_2$, $b_1 < b < b_2$. Здесь a_1, b_1 и a_2, b_2 – обозначения соответственно нижних и верхних границ значений параметров a и b .

Итак, вопрос состоит в том, чтобы найти строгие границы значения $f(a,b)$ при известных границах значений a и b , используя свойства монотонности функции $z=f(x,y)$ по соответствующим аргументам. Для этого приходится вычислять частные производные по этим аргументам.

1) РЕШЕНИЕ В СИСТЕМЕ ПРОГРАММИРОВАНИЯ BORLAND PASCAL

```
uses crt;
var a,da,a1,a2,b,db,b1,b2,c,dc,c1,c2,z,z1,z2:real;
begin
clrscr;
a:=0.7219; da:=0.0001;
b:=135.347; db:=0.001;
c:=0.013; dc:=0.001;
writeln('Вычисление по формуле  $Z=(\ln(b)-a)/(a*a+12*c)$  методом границ');
writeln('a=',a:8:4,' da=',da:8:4);
writeln('b=',b:8:4,' db=',db:8:4);
writeln('c=',c:8:4,' dc=',dc:8:4);
a1:=a-da; a2:=a+da;
b1:=b-db; b2:=b+db;
c1:=c-dc; c2:=c+dc;
writeln('нижняя граница a=',a1:1:5);
writeln('верхняя граница a=',a2:1:5);
writeln('нижняя граница b=',b1:1:5);
writeln('верхняя граница b=',b2:1:5);
writeln('нижняя граница c=',c1:1:5);
writeln('верхняя граница c=',c2:1:5);
z:=(ln(b)-a)/(a*a+12*c);
writeln('число z= ',z:1:5);
z1:=(ln(b1)-a2)/(a2*a2+12*c2);
writeln('нижняя граница z1= ',z1:1:5);
z2:=(ln(b2)-a1)/(a1*a1+12*c1);
writeln('верхняя граница z2= ',z2:1:5);
end.
```

Вычисление по формуле $Z=(\ln(b)-a)/(a*a+12*c)$ методом границ

a= 0.7219 da= 0.0001
b=135.3470 db= 0.0010
c= 0.0130 dc= 0.0010
нижняя граница a=0.72180
верхняя граница a=0.72200

нижняя граница b=135.34600
 верхняя граница b=135.34800
 нижняя граница c=0.01200
 верхняя граница c=0.01400
 число z= 6.18180
 нижняя граница z1= 6.07273
 верхняя граница z2= 6.29486

2) РЕШЕНИЕ В MATHCAD(e)

ВЫЧИСЛЕНИЕ МЕТОДОМ ГРАНИЦ ПО ФОРМУЛЕ $z = \frac{\ln(b) - a}{a^2 + 12 \cdot c}$,

ГДЕ ЧИСЛА a=0.7219, b=135.347, c=0.013 ЗАДАНЫ ВСЕМИ СВОИМИ ВЕРНЫМИ ЗНАЧАЩИМИ ЦИФРАМИ В ШИРОКОМ СМЫСЛЕ

ИССЛЕДУЕМ ФУНКЦИЮ $z = \frac{\ln(b) - a}{a^2 + 12 \cdot c}$ НА ВОЗРАСТАНИЕ И УБЫВАНИЕ ПО ПАРАМЕТРАМ a,b,c, ДЛЯ ЧЕГО ВЫЧИСЛЯЕМ ЕЁ ЧАСТНЫЕ ПРОИЗВОДНЫЕ И ОПРЕДЕЛЯЕМ ИХ ЗНАК ПРИ ЗАДААННЫХ ЗНАЧЕНИЯХ ПАРАМЕТРОВ

$$z = \frac{\ln(b) - a}{a^2 + 12 \cdot c}$$

$$\frac{d}{da} \left(\frac{\ln(b) - a}{a^2 + 12 \cdot c} \right) \rightarrow \frac{-1}{a^2 + 12 \cdot c} - 2 \cdot \frac{\ln(b) - a}{(a^2 + 12 \cdot c)^2} \cdot a$$

$$\frac{d}{da} \left(\frac{\ln(b) - a}{a^2 + 12 \cdot c} \right) = \frac{-1}{a^2 + 12 \cdot c} - 2 \cdot \frac{\ln(b) - a}{(a^2 + 12 \cdot c)^2} \cdot a < 0$$

$$\frac{d}{db} \left(\frac{\ln(b) - a}{a^2 + 12 \cdot c} \right) \rightarrow \frac{1}{b \cdot (a^2 + 12 \cdot c)}$$

$$\frac{d}{db} \left(\frac{\ln(b) - a}{a^2 + 12 \cdot c} \right) = \frac{1}{b \cdot (a^2 + 12 \cdot c)} > 0$$

$$\frac{d}{dc} \left(\frac{\ln(b) - a}{a^2 + 12 \cdot c} \right) \rightarrow -12 \cdot \frac{\ln(b) - a}{(a^2 + 12 \cdot c)^2}$$

$$\frac{d}{dc} \left(\frac{\ln(b) - a}{a^2 + 12 \cdot c} \right) = -12 \cdot \frac{\ln(b) - a}{(a^2 + 12 \cdot c)^2} < 0$$

$$a := 0.7219$$

$$b := 135.347$$

$$c := 0.013$$

$$\Delta a := 0.0001$$

$$\Delta b := 0.001$$

$$\Delta c := 0.001$$

$$a1 := a - \Delta a$$

$$a1 = 0.7218$$

$$b1 := b - \Delta b$$

$$b1 = 135.346$$

$$a2 := a + \Delta a$$

$$a2 = 0.722$$

$$b2 := b + \Delta b$$

$$b2 = 135.348$$

$$\begin{aligned}c1 &:= c - \Delta c & c1 &= 0.012 \\c2 &:= c + \Delta c & c2 &= 0.014\end{aligned}$$

$$z := \frac{\ln(b) - a}{a^2 + 12 \cdot c} \quad z = 6.181800$$

$$z1 := \frac{\ln(b1) - a2}{a2^2 + 12 \cdot c2} \quad z1 = 6.072728$$

$$z2 := \frac{\ln(b2) - a1}{a1^2 + 12 \cdot c1} \quad z2 = 6.294856$$

ВЫЧИСЛЕНИЕ ПО ФОРМУЛЕ МЕТОДОМ ГРАНИЦ С ПОШАГОВОЙ РЕГИСТРАЦИЕЙ ВЫЧИСЛЕНИЙ

$$\begin{aligned}a &:= 0.7218 & b &:= 135.34 & c &:= 0.013 \\ \Delta a &:= 0.0001 & \Delta b &:= 0.001 & \Delta c &:= 0.001\end{aligned}$$

$$\begin{aligned}a1 &:= a - \Delta a & a1 &= 0.7218 & b1 &:= b - \Delta b & b1 &= 135.346 \\ a2 &:= a + \Delta a & a2 &= 0.722 & b2 &:= b + \Delta b & b2 &= 135.348\end{aligned}$$

$$\begin{aligned}c1 &:= c - \Delta c & c1 &= 0.012 \\ c2 &:= c + \Delta c & c2 &= 0.014\end{aligned}$$

$$\begin{aligned}d1 &:= \ln(b1) & d1 &= 4.90783 \\ d2 &:= \ln(b2) & d2 &= 4.90785\end{aligned}$$

$$\begin{aligned}e1 &:= d1 - a2 & e1 &= 4.18583 \\ e2 &:= d2 - a1 & e2 &= 4.18605\end{aligned}$$

$$f1 := (a1)^2 + 12 \cdot c1 \quad f1 = 0.664995$$

$$f2 := (a2)^2 + 12 \cdot c2 \quad f2 = 0.68928$$

$$g1 := \frac{e1}{f2} \quad g1 = 6.07273$$

$$g2 := \frac{e2}{f1} \quad g2 = 6.29486$$

$$z := \frac{\ln(b) - a}{a^2 + 12 \cdot c} \quad z = 6.18180$$

$$6.07273 < z < 6.29486$$

$$z1 = 6.07273 \quad z2 = 6.29486 \quad z = 6.18180$$

7. Контрольные вопросы

1. Из каких основных этапов состоит процесс решения задачи с помощью ЭВМ? Дайте характеристику каждого этапа.
2. Из каких частей складывается общая погрешность решения задачи?
3. Какие основные способы представления чисел используются в ЭВМ?
4. Какие числа называют машинным нулём?
5. Каковы причины возникновения ошибок при проведении расчетов в системе чисел с плавающей запятой, порождаемых структурой этой системы?
6. Что такое абсолютная погрешность приближенного значения величины? граница абсолютной погрешности?
7. Как с помощью границы абсолютной погрешности Δx известного приближенного значения x можно указать возможные значения его нижней и верхней границ?
8. Каким образом определяется граница абсолютной погрешности Δx для приближенного значения x , получаемого в результате многократных измерений?
9. Что такое относительная погрешность приближенного значения величины? граница относительной погрешности?
10. Как можно вычислить абсолютную погрешность приближения x , если известна его относительная погрешность?
11. Какие цифры в записи приближенного числа называются верными в широком смысле? верными в строгом смысле?
12. Верно ли утверждение, что компьютер, округляющий числа по методу отбрасывания, всегда выдает результаты, записанные только верными цифрами? Почему?
13. Какие цифры в записи приближенного числа называются значащими?
14. Что такое округление числа? погрешность округления?
15. Какие различаются погрешности округления методом отбрасывания?
16. Из чего складывается полная погрешность округленного числа?
17. Почему в функции `stepen 10` переменная `k` описана как `longint`?
18. Какие ограничения существуют в приведенной выше программе относительно значений величины n при $d < 1$ и $d \geq 1$ и как эти ограничения изменятся, если описать переменную `k` как `integer`?
19. В значениях x и Δx десятичная запятая одновременно перемещена на одинаковое количество разрядов влево или вправо. Изменится ли при этом количество верных цифр в числе x ?
20. Как устанавливается количество верных в строгом смысле цифр по величине относительной погрешности приближенного числа, если первая цифра относительной погрешности меньше 5? больше или равна 5?
21. Как объясняется целесообразность сохранения излишних десятичных знаков в более точных слагаемых при сложении нескольких чисел?
22. Каким способом можно быстро оценить относительную погрешность суммы нескольких слагаемых одного знака, если известны относительные погрешности каждого слагаемого?
23. По какой причине в вычислениях следует избегать вычитания близких по величине чисел?
24. Как объясняется нецелесообразность сохранения излишних значащих цифр в более точных данных при умножении или делении нескольких чисел?
25. В какой зависимости находятся абсолютная погрешность значения функции одной переменной от абсолютной погрешности значения аргумента?
26. Как объясняется резкая потеря точности значения $f(x)$ для приближенных значений аргумента x на участках с большими по модулю значениями $f'(x)$?
27. Как формулируется обратная задача теории погрешностей?

28. Как формулируются правила подсчета цифр?
29. В каких случаях рекомендуется применять правила подсчета цифр?
30. Какие два способа применения правил подсчета цифр возможны в вычислениях на МК и ЭВМ?
31. Какова последовательность действий на каждом промежуточном этапе расчетной таблицы в вычислениях по правилам подсчета цифр с пооперационным учетом ошибок? на заключительном этапе?
32. Как оформляются вычисления со строгим учетом предельных погрешностей при пооперационном учете ошибок?
33. Какова последовательность действий на каждом промежуточном этапе расчетной таблицы в вычислениях по методу строгого учета предельных погрешностей с пооперационным учетом ошибок? на заключительном этапе?
34. Как вычисляются предельные погрешности результата при использовании методики итоговой оценки ошибки вычислений?
35. В чем основное отличие метода границ от вычислений по методу строго учета границ погрешностей?
36. Какова последовательность действий в вычислениях по методу границ с пооперационным учетом ошибок на каждом промежуточном этапе расчетной таблицы? на заключительном этапе?
37. Какие встроенные функции округления могут быть использованы в процессе вычислений с помощью инструментальных средств?
38. Какие инструментальные программные средства являются более эффективными при использовании различных методов вычисления?
39. Как работает механизм завышения оценки погрешности путем её замены предельной погрешностью в случае вычисления произведения большого числа сомножителей?

8. Варианты заданий для лабораторных и курсовых работ

Таблица 5

№	Функция	x
1	$y=2^{\sqrt[3]{x}-1}$	2.02653
2	$y=\sqrt[3]{x^3-8}$	3.1174
3	$y=\sqrt{\lg(x)}$	0.9753
4	$y=\log_{\sqrt{7}}(x)$	1.43172
5	$y=\sin(\lg(x))$	0.7123
6	$y=\operatorname{ch}(x)$	0.36481
7	$y=\sqrt[3]{\operatorname{ch}(x)}$	0.5746
8	$y=\cos(\sqrt{x})$	0.67812
9	$y=2^{\sqrt{x^2-1}}$	2.112
10	$y=2^{\sqrt{\cos(x)}}$	0.64513
11	$y=\sqrt{\arctan(x)}$	1.2317
12	$y=\sqrt[3]{\arctan(x)}$	0.93721
13	$y=\operatorname{cth}(x)$	0.53745
14	$y=\operatorname{th}(x)$	0.28873
15	$y=e^{\sin(x)}$	0.23931
16	$y=\sqrt[3]{\operatorname{sh}(x)}$	0.86129
17	$y=2^{\cos(x)}$	0.74033
18	$y=\sqrt{\operatorname{ctg}(x)}$	0.1495
19	$y=e^{\sin(x)}$	0.44745
20	$y=\sin(x)$	0.70415
21	$y=e^{x^2-1}$	1.70423
22	$y=\operatorname{arcctg} \sqrt[3]{x}$	0.8437
23	$y=\arctg \sqrt[3]{x}$	2.39247
24	$y=\sin(x) \sqrt{x}$	0.26352
25	$y=\cos(x) \sqrt{x}$	0.76532
26	$y=\lg(x)$	0.628
27	$y=\operatorname{ctg}^3(x)$	0.4942
28	$y=\lg(\sin(x))$	1.2371
29	$y=\exp(\lg^2(x))$	0.6852
30	$y=\exp(\sqrt{\operatorname{ctg}(x)})$	0.932

Таблица 6

№	Функция	x
1	$y=\text{th}(x)$	1.1732
2	$y=\text{cth}(x)$	1.697
3	$y=\sqrt[3]{\text{tg}^2(x)-1}$	0.926
4	$y=\cos(e^x)$	0.517
5	$y=\arctan(\sqrt{x})$	1.7632
6	$y=\sqrt{x-1} \cdot \sin(x)$	1.69311
7	$y=3^x$	0.9137
8	$y=2x^2-2x+4$	2.61131
9	$y=\text{tg}(\sqrt{x})$	0.3913
10	$y=x^2\sqrt{x}$	2.172
11	$y=\arcsin(x)$	0.79536
12	$y=x^3-1$	2.1123
13	$y=\sqrt{\ln(x)}$	2.3132
14	$y=\ln(\sqrt{x^3-1})$	3.7451
15	$y=2^{\text{tg}(x)}$	0.752
16	$y=\text{tg}^2(x)$	0.6341
17	$y=\sin(x)$	1.8714
18	$y=\sqrt{x-1} \cdot \cos(x)$	1.9531
19	$y=x \cdot \cos(\sqrt{x-1})$	2.34175
20	$y=\text{eps}(x^2)$	1.234
21	$y=\cos^2(x)$	2.1324
22	$y=x \cdot \sin(\sqrt{x-1})$	2.6372
23	$y=x \cdot \text{tg}(\sqrt{x-1})$	1.5414
24	$y=x \cdot \text{ctg}(\sqrt{x-1})$	2.9327
25	$y=x \cdot \sec(\sqrt{x-1})$	2.9163
26	$y=\text{tg}(\sin(x))$	0.1385
27	$y=\sin(\sqrt{x})$	0.47942
28	$y=\text{tg}(\sin^2(x))$	0.7921
29	$y=\text{ctg}(\sqrt{x})$	0.332
30	$y=\sin^2(x)$	1.1745

Таблица 7

№	Функция	x
1	$y=2^{\sqrt{x^2-1}}$	2.975
2	$y=5^{\sqrt{x}}$	1.6413
3	$y=2^{\sqrt[3]{x^2-1}}$	2.37411
4	$y=5^{\sqrt[3]{x}}$	1.9247
5	$y=\exp(x)$	1.112
6	$y=5^{\arctg(x)}$	0.9174
7	$y=\sin(e^x)$	-0.5714
8	$y=\exp(\sqrt{\sin(x)})$	1.2351
9	$y=\operatorname{ctg}(x)$	1.2409
10	$y=\arctg(e^x)$	0.413
11	$y=\operatorname{tg}(e^x)$	0.2233
12	$y=\sin(\sqrt{e^x})$	0.538
13	$y=\operatorname{ctg}(3^x)$	0.1725
14	$y=\sqrt{\sin(x)}$	1.432
15	$y=\sqrt[3]{x^2-1}$	2.974
16	$y=\sec(\sqrt{x})$	1.5124
17	$y=\cos(\sqrt{x})$	0.8413
18	$y=\sqrt[3]{x}$	0.6611
19	$y=\operatorname{tg}(\sin(x))$	0.4374
20	$y=\operatorname{tg}(\cos(x))$	1.8214
21	$y=\sin(\operatorname{tg}(x))$	0.7193
22	$y=\sqrt[4]{x^3-1}$	1.6925
23	$y=\sqrt[3]{x^2-1}$	2.753
24	$y=\sqrt[5]{x^3-2}$	3.1745
25	$y=\sqrt[4]{x^3+2}$	2.291
26	$y=\arctg(x)$	0.4714
27	$y=\operatorname{arcctg}(\sqrt{x})$	1.4456
28	$y=\operatorname{acrtg}(\sqrt{x})$	2.112
29	$y=\arctan(\sqrt[3]{x})$	1.934
30	$y=\operatorname{tg}^2(x)$	1.347

Таблица 8

№	Функция	x
1	$y=\sin(\sqrt[3]{x})$	0.802
2	$y=\cos(x^2)$	1.4437
3	$y=x \cdot \sqrt{(x+1)}$	3.1415
4	$y=\operatorname{sh}(x)$	1.2314
5	$y=\cos(\sqrt[3]{x})$	-0.572
6	$y=\sin(\operatorname{ctg}(x))$	0.645
7	$y=\operatorname{ch}(x)$	2.1371
8	$y=\sec(x)$	0.7851
9	$y=\operatorname{th}^2(x)$	1.632
10	$y=\sqrt{\operatorname{th}(x)}$	1.4945
11	$y=\operatorname{tg} \sqrt[5]{x}$	8.2416
12	$y=\sin(\operatorname{th}(x))$	0.38725
13	$y=\cos \sqrt[7]{x}$	10.8441
14	$y=\operatorname{ctg} \sqrt[5]{x}$	0.805637
15	$y=\operatorname{tg} \sqrt[7]{x}$	2.3485
16	$y=\sqrt[3]{\operatorname{ch} x}$	0.088748
17	$y=\arcsin \frac{x}{\sqrt{x^2+10^2}}$	38.4258
18	$y=\sqrt[3]{\operatorname{sh} x}$	1.784
19	$y=\sqrt{x^2-\operatorname{tg} x}$	6.4257
20	$y=\sqrt[3]{x^2+\operatorname{ctg} x}$	3.87683
21	$y=\sec(\operatorname{th} x)$	0.12356
22	$y=\operatorname{cosec}(\operatorname{cth} x)$	0.75244
23	$y=\sqrt[3]{x^2+2x}$	45.7832
24	$y=\sec(\operatorname{ch} x)$	0.66835
25	$y=\operatorname{cosec}(\operatorname{sh} x)$	0.5748
26	$y=\frac{1}{\sqrt{x^2-\sqrt{x^3}}}$	27.1541
27	$y=\frac{x+20}{\sqrt{20^2-x}}$	83.736
28	$y=\frac{\sqrt[3]{x^2-1}}{\sqrt{x^2+1}}$	5.435
29	$y=\frac{\sqrt{x^2+1}}{\sqrt[3]{x^2-1}}$	7.3272
30	$y=\frac{1}{\sqrt[3]{x^3-\sqrt{x}}}$	3.92215

Таблица 9

№	a	b	c	Формула
1	1.6531	3.78	0.158	$z = \frac{a^2 b}{c^3}$
2	0.7568	21.7	2.65	$z = \sqrt{\frac{ac}{b}}$
3	10.82	2.786	0.28	$z = \frac{\sqrt[3]{a-b}}{c(c-a)}$
4	8.53	6.271	12.48	$z = \frac{(a+b)c^3}{4}$
5	8.51	6.1705	12.32	$z = \frac{(a+b)c}{12}$
6	3.236	15.8	0.64	$z = \frac{(a+b)c}{(c-a)^2}$
7	0.562	0.2518	0.68	$z = \frac{(a+b)^2}{2c}$
8	0.834	0.3523	0.747	$z = \frac{(a^2 + b^2)c}{5}$
9	1.245	0.121	2.34	$z = \frac{b^2 a}{c}$
10	23.16	8.23	145.5	$z = \frac{(a+b)c}{\sqrt{c-d}}$
11	0.142	1.71	3.727	$z = \frac{a^2 b^3}{c}$
12	15.71	3.28	7.2	$z = \frac{(a-b)c}{\sqrt{b+c}}$
13	17.8	32.42	11.42	$z = \frac{a^2 + 4ab + b^2}{(a-b)^3}$
14	12.751	3.71	0.285	$z = \frac{ac}{b^2}$
15	2.456	1.76	-	$z = \frac{1}{6}pb(3a^2 + b)$
16	2.348	4.37	0.235	$z = \frac{c\sqrt{b-a}}{c+a}$
17	3.14	71.513	3.274	$z = \frac{1}{64}p\sqrt{b^4 - a^4}$
18	0.8345	13.8	1.84	$z = \sqrt{\frac{ac}{b}}$
19	0.0976	2.371	1.15874	$z = \frac{a - \sin b}{13c + b}$

Продолжение таблицы 9

20	3.49	0.845	0.0037	$z = \frac{2ac + b}{\sqrt{b - c}}$
21	0.11587	4.25	3.00971	$z = \frac{a^3 - b^2}{\sqrt{ac + b}}$
22	82.3574	34.1	7.00493	$z = \frac{\ln(a - b)}{\sqrt{b - c}}$
23	7.345	0.31	0.09872	$z = \frac{4a + \ln b}{ac - b}$
24	3.672	4.63	0.0278	$z = \frac{b - \cos a}{3a + c}$
25	0.1756	3.71	0.285	$z = \frac{a + b}{\sqrt{(b - c)a}}$
26	13.57	3.7	4.226	$z = \frac{a^3(c + b)}{a - b}$
27	0.317	13.57	0.751	$z = \frac{\sqrt{ab}}{c} + a$
28	0.317	33.827	14.85	$z = \frac{a + b}{a\sqrt{b - c}}$
29	5.52	3.27	14.123	$z = \frac{ab}{\cos(c - 2a)}$
30	2.25	0.02	9.87	$z = \frac{\sin(a + \sqrt{b})}{a - \sin c}$

Таблица 10

№	a	b	c	Формула
1	3.4	6.22	0.149	$z = \frac{\sqrt{ab}}{b - 2c}$
2	4.05	6.723	0.03254	$z = \frac{(b - c)^2}{2a + b}$
3	0.7219	135.347	0.013	$z = \frac{\ln b - a}{a^2 + 12c}$
4	3.672	4.63	0.0278	$z = \frac{b - \sin a}{a + 3c}$
5	1.24734	0.346	0.051	$z = \frac{10c + \sqrt{b}}{a^2 - b}$
6	11.7	0.0937	5.081	$z = \frac{(a - c)^2}{\sqrt{a} + 3b}$
7	1.75	1.21	0.041	$z = \frac{a - \sin b}{b^2 + 6c}$
8	18.0354	3.7251	0.071	$z = \frac{\sqrt{b - c}}{\ln a + b}$
9	0.113	0.1056	89.4	$z = \frac{\ln c - 10a}{\sqrt{bc}}$
10	0.317	3.27	4.7561	$z = \frac{c + \sin b}{c - a^2}$
11	0.0399	4.83	0.072	$z = \frac{\ln(b + c)}{b - ac}$
12	1.574	1.40	1.1236	$z = \frac{\sqrt{a + b}}{3a - c}$
13	12.72	0.34	0.0290	$z = \frac{ab - 4c}{\ln a + b}$
14	3.49	0.845	0.0037	$z = \frac{a - \cos b}{13c + b}$
15	0.0976	2.371	1.15874	$z = \frac{ac + b}{\sqrt{b - c}}$
16	82.3574	34.1	7.00493	$z = \frac{\lg(a - b)}{\sqrt{b - c}}$
17	0.11587	4.25	3.00971	$z = \frac{b + \cos c}{b + 2a}$
18	3.71452	3.03	0.765	$z = \frac{a^2 - b}{\sqrt{ab + c}}$
19	7.345	0.31	0.09872	$z = \frac{\ln a + 4b}{ab - c}$

Продолжение таблицы 10

20	0.038	3.9353	5.75	$z = \frac{b^2 + \ln c}{\sqrt{c - a}}$
21	9.542	3.128	0.17	$z = \frac{a\sqrt{b - c}}{a + c}$
22	9.79	2.3327	4.198	$z = \frac{\sqrt{abc}}{\ln a - \operatorname{tg} b}$
23	6.66	3.5	1.141	$z = \ln(a + 3\sqrt{c})^b$
24	5.14	6.22	0.0072	$z = \frac{\operatorname{tg}(a + 4c)}{b^3}$
25	29.49	87.878	4.403	$z = \frac{a \ln b}{\sin(\sqrt{a + c})}$
26	74.079	5.3091	6.234	$z = \frac{0.8 \ln b}{\sqrt{a + bc}}$
27	3.4	3.128	0.17	$z = \frac{a\sqrt{b - c}}{a + c}$
28	1.75	1.215	0.041	$z = \frac{a - \sin c}{b^2 + 6c}$
29	3.672	4.63	0.0278	$z = \frac{a^3(c + b)}{a - b}$
30	13.57	3.7	4.226	$z = \frac{a^3(b + c)}{b^2 + 6c}$

Таблица 11

№	a	b	c	Формула
1	0.317	3.27	4.7561	$z = \frac{4\sqrt{a+b}}{ab-c}$
2	12.72	0.34	0.0290	$z = \frac{ab-4c}{\ln a+3b}$
3	3.49	0.845	0.0037	$z = \frac{a-tg(b)}{13c-b}$
4	0.0976	2.371	1.15887	$z = \frac{ac+3b}{\sqrt{b-c}}$
5	82.3574	34.12	7.00493	$z = \frac{\ln(a-b)}{\sqrt{b+c}}$
6	3.71452	3.03	0.756	$z = \frac{a^2-b}{\sqrt{ab-c}}$
7	0.11587	4.256	3.00971	$z = \frac{b+\cos c}{3b+2a}$
8	4.05	6.723	0.03254	$z = \frac{(b-c)}{2a+b}$
9	0.7219	135.347	0.013	$z = \frac{\ln b-a}{a^2-12c}$
10	11.7	0.0937	5.081	$z = \frac{(a-c)^2}{\sqrt{a+3b}}$
11	5.7568	21.7	2.65	$z = \frac{\sqrt[3]{a-c}}{b+2\ln a}$
12	5.33	23.123	8.802	$z = \frac{(a+4c)^3}{b-5a^2}$
13	6.0031	1.005	1.6135	$z = \left(\frac{a+6b}{\ln c-a} \right)^3$
14	0.037	5.777	3.441	$z = \left(\frac{2a+\cos b}{c-3b} \right)^2$
15	8.317	13.521	6.123	$z = a^3 + 2\sin(b - \ln c)$
16	7.345	0.31	0.09871	$z = \frac{\ln a+3b}{ab-c}$
17	0.2471	0.0948	4.378	$z = \frac{2tg(a-b)}{a^2c+b}$
18	1.284	4.009	3.2175	$z = \frac{4\sqrt{a+c}}{ab-c}$
19	18.407	149.12	2.3078	$z = \frac{\sin(a-\sqrt{b})}{c+\ln b}$

Продолжение таблицы 11

20	29.49	87.878	4.403	$z = \frac{a \ln b}{\sin(\sqrt{a+c})}$
21	74.079	5.3091	6.234	$z = \frac{0.8 \ln b}{\sqrt{a+bc}}$
22	3.4	6.22	0.149	$z = \frac{\sqrt{a}}{bc - \ln c}$
23	5.387	13.527	0.7565	$z = \frac{\sqrt{ab}}{b-2c}$
24	1.75	1.215	0.041	$z = \frac{a - \sin c}{b^2 + 6c}$
25	0.317	13.57	0.751	$z = \frac{\sqrt{ab}}{c} + a$
26	0.317	33.827	14.85	$z = \frac{a+b}{a\sqrt{b-c}}$
27	5.52	3.27	14.123	$z = \frac{ab}{\cos(c-2a)}$
28	9.542	3.128	0.17	$z = \frac{a\sqrt{b-c}}{a+c}$
29	9.79	2.3327	4.198	$z = \frac{\sqrt{abc}}{\ln a - \operatorname{tg} b}$
30	6.67	3.5	1.141	$z = \ln(a + 3\sqrt{c})^b$

Таблица 12

№	a	b	c	Формула
1	5.14	6.22	0.007	$z = \frac{tg(a + 4c)}{b^3}$
2	2.258	0.027	9.87	$z = \frac{\sin(a + \sqrt{b})}{a - \sin c}$
3	0.113	0.1056	89.4	$z = \frac{\ln c - 10a}{\sqrt{bc}}$
4	2.674	31.48	-	$z = \frac{ab}{\sqrt{a + b^2}}$
5	1.3172	2.675	-	$z = \frac{a + \sqrt{b}}{\ln(a^2 + b^2)}$
6	63.173	32.967	-	$z = \frac{e^a - \sqrt[3]{b}}{\ln(1 + a^2)}$
7	2.674	32.317	-	$z = \lg \frac{\cos^2 a + b}{a^{\sqrt{b}} + b^{\sqrt{a}}}$
8	2.156	0.927	-	$z = \frac{0.62a + \sqrt{16.9b}}{\lg(a + b)}$
9	3.85	2.0435	1.217	$z = \frac{ab}{\sqrt[3]{c}}$
10	1.141	3.156	-	$z = \frac{a^2 + 4ab + b^2}{(a + b)^2}$
11	4.3	17.21	8.221	$z = \left(\frac{(a + b)c}{b - a} \right)^2$
12	228.6	86.422	68.71	$z = \frac{\sqrt{a} \cdot b}{c}$
13	2.754	11.7	10.536	$z = \frac{(a + b)c}{(c - a)^2}$
14	3.456	0.642	7.12	$z = \frac{a^2 b}{c}$
15	0.643	2.17	5.8432	$z = \frac{ab^3}{c}$
16	0.3575	2.63	0.854	$z = \frac{ab}{c^2}$
17	2.456	1.76	-	$z = \frac{1}{6}pb(3a^2 + b^2)$
18	32.1471	2.31435	-	$z = \ln(a^2 + b)$
19	7.544	4.3251	-	$z = \frac{1}{\sqrt{a - \sqrt{b}}}$
20	0.2538	10.2117	-	$z = \frac{a}{\sqrt{b}}$

Продолжение таблицы 12

21	129.361	7.3251	-	$z = \frac{a}{b^2}$
22	3.00971	1.15874	-	$z = \frac{1}{(a-b)^2}$
23	4.75617	3.2175	-	$z = \frac{1}{\sqrt{a^2 - b^2}}$
24	12.935	53.3211	-	$z = \arctg \frac{a}{b}$
25	92.451	31.1425	-	$z = \frac{a}{\sqrt{a^2 + b^2}}$
26	4.14005	6.23587	-	$z = e^{\sin \frac{a}{b}}$
27	3.0975	5.64115	-	$z = \frac{a^2 - b^2}{a^2 + b^2}$
28	5.6421	4.636	-	$z = \ln \left(1 + \frac{a}{b} \right)$
29	2.4341	3.4		$z = \ln \operatorname{tg} \frac{a}{b}$
30	0.374	0.6781	-	$z = \left(ab + \frac{a}{b} \right)^2$

9. Образец оформления курсовой работы по дисциплинам «Информатика» и «Вычислительная математика».

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Российский государственный технологический университет
им. К.Э. Циолковского «РГТУ-МАТИ»**

КАФЕДРА ВЫСШЕЙ МАТЕМАТИКИ

Курсовая работа
по дисциплине «ИНФОРМАТИКА»

Часть 1. Приближенные вычисления

Вариант 11

Факультет	4
Группа	4МТМ-2ДБ-010
Выполнила	С.С. Тихомирова
Преподаватель	Е.К. Гурьев

МОСКВА 2007

СОДЕРЖАНИЕ

ЗАДАНИЕ.....	3
1. ПРЕДСТАВЛЕНИЕ ЧИСЕЛ С ФИКСИРОВАННОЙ И ПЛАВАЮЩЕЙ ТОЧКОЙ В ПАМЯТИ ЭВМ.....	4
2. ПОГРЕШНОСТИ ВЫЧИСЛЕНИЯ ЭЛЕМЕНТАРНЫХ ФУНКЦИЙ С УЧЕТОМ ПРЕДСТАВЛЕНИЯ В ЭВМ ВЕЩЕСТВЕННЫХ ЧИСЕЛ В РАЗЛИЧНЫХ ФОРМАТАХ ЗАПИСИ ДАННЫХ И ОСОБЕННОСТЕЙ ПРОЦЕДУРЫ ВЫВОДА WRITELN В ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PASCAL.....	7
3. НАХОЖДЕНИЕ МАШИННОГО ЭПСИЛОН.....	8
4. АБСОЛЮТНАЯ И ОТНОСИТЕЛЬНАЯ ПОГРЕШНОСТИ ЧИСЛА.....	10
4.1. РЕШЕНИЕ С ПОМОЩЬЮ ПРОГРАММИРОВАНИЯ В СИСТЕМЕ ПРОГРАММИРОВАНИЯ BORLAND PASCAL	11
4.2. РЕШЕНИЕ В MATHCAD(e)	11
5. ПРАВИЛЬНАЯ ЗАПИСЬ И ОКРУГЛЕНИЕ ЧИСЕЛ.....	12
5.1. ОКРУГЛЕНИЕ ЧИСЕЛ ПО ЗАДАННОМУ ЗНАЧЕНИЮ АБСОЛЮТНОЙ ПОГРЕШНОСТИ В СИСТЕМЕ ПРОГРАММИРОВАНИЯ BORLAND PASCAL.....	12
5.2. РЕШЕНИЕ В MATHCAD(e).....	13
5.3. ПРОГРАММНОЕ ОКРУГЛЕНИЕ ДЕСЯТИЧНЫХ ЧИСЕЛ ПО ЗАДАННОМУ ЗНАЧЕНИЮ АБСОЛЮТНОЙ ПОГРЕШНОСТИ.....	14
6. ВЫЧИСЛЕНИЕ АБСОЛЮТНОЙ И ОТНОСИТЕЛЬНОЙ ПОГРЕШНОСТИ ФУНКЦИИ ОДНОЙ ПЕРЕМЕННОЙ.....	18
6.1. ПРОГРАММА ВЫЧИСЛЕНИЯ АБСОЛЮТНОЙ И ОТНОСИТЕЛЬНОЙ ПОГРЕШНОСТИ ФУНКЦИИ В СИСТЕМЕ ПРОГРАММИРОВАНИЯ BORLAND PASCAL.....	18
6.2. РЕШЕНИЕ В MATHCAD(e)	19
7. ОЦЕНКА ПОГРЕШНОСТИ ФУНКЦИИ НЕСКОЛЬКИХ ПЕРЕМЕННЫХ.....	20
8. ОЦЕНКА ПОГРЕШНОСТИ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ.....	23
8.1. РЕШЕНИЕ В СИСТЕМЕ ПРОГРАММИРОВАНИЯ BORLAND PASCAL.....	23
8.2. РЕШЕНИЕ В MATHCAD(e).....	26
9. ПРИБЛИЖЕННЫЕ ВЫЧИСЛЕНИЯ ПО ФОРМУЛЕ.....	28
9.1. ВЫЧИСЛЕНИЯ ПО ПРАВИЛАМ ПОДСЧЕТА ЦИФР.....	28
9.2. ВЫЧИСЛЕНИЯ СО СТРОГИМ УЧЕТОМ ПРЕДЕЛЬНЫХ АБСОЛЮТНЫХ ПОГРЕШНОСТЕЙ.....	30
9.2.1. РЕШЕНИЕ В СИСТЕМЕ ПРОГРАММИРОВАНИЯ BORLAND PASCAL.....	30
9.2.2. РЕШЕНИЕ В MATHCAD(e)	32
9.3. ВЫЧИСЛЕНИЯ ПО МЕТОДУ ГРАНИЦ.....	33
9.3.1. РЕШЕНИЕ В СИСТЕМЕ ПРОГРАММИРОВАНИЯ BORLAND PASCAL.....	33
9.3.2. РЕШЕНИЕ В MATHCAD(e)	35
ЛИТЕРАТУРА.....	38

ЗАДАНИЕ

1. Исследовать реальную точность вычисления элементарной функции $y = \sqrt{x-1} * \cos(x)$ в различных форматах представления данных и соотнести её с возможностями стандартных процедур вывода при значении аргумента $x=2.5732$, заданного всеми своими верными цифрами в узком смысле.
2. Установить значение машинного эпсилон для системы программирования Borland Pascal в различных форматах представления данных.
3. Для заданного числа $x=2.5732$, заданного всеми своими верными цифрами в строгом смысле, найти абсолютную $\Delta(x^*)$ и относительную $d(x^*)$ погрешности.
4. Для числа $x=2.5732$, заданного всеми своими верными значащими цифрами в широком смысле, выполнить округление до трех значащих цифр и найти на микрокалькуляторе абсолютную и относительную погрешности. Написать программу и выполнить программное округление этого числа по заданному значению его абсолютной погрешности в системе программирования Borland Pascal и в Mathcad.
4. Оценить абсолютную и относительную погрешность вычисления функции $y = \sqrt{x-1} * \cos(x)$ при $x=2.5732$ (в задании 1) в системе программирования Borland Pascal и в Mathcad.
5. Вычислить абсолютную и относительную погрешность арифметических операций (сложение, умножение, деление, вычитание, возведение в степень и извлечение корня) с числами $a=0.7219$, $b=135.347$, заданными всеми своими значащими цифрами в широком смысле в системе программирования Borland Pascal и в Mathcad.
6. Вычислить значение величины Z по формуле $z = \frac{\ln(b) - a}{a^2 + 12 * c}$ при заданных значениях параметров $a=0.7219$, $b=135.347$, $c=0.013$ следующими методами:
 - 6.1. вычислениями по правилам подсчета цифр;
 - 6.2. вычислениями со строгим учетом предельных абсолютных погрешностей;
 - 6.3. вычислениями по методу границс пошаговой и итоговой регистрацией результатов вычислений, составляя программу в системе программирования Borland Pascal и используя Mathcad. Числа a , b , c заданы всеми своими верными значащими цифрами в широком смысле.

1. ПРЕДСТАВЛЕНИЕ ЧИСЕЛ С ФИКСИРОВАННОЙ И ПЛАВАЮЩЕЙ ТОЧКОЙ В ПАМЯТИ

В ЭВМ используются в основном два способа представления чисел: с естественным размещением (фиксированной) запятой и в форме с плавающей запятой. Характерной особенностью машинного представления чисел является конечность разрядной сетки, в которой хранятся числа в машине.

На рис. 1 изображена схема представления числа с зафиксированной запятой, или естественным размещением запятой, на примере 8-разрядного калькулятора. Десятичная запятая в этом случае может оказаться в любом цифровом разряде сетки.

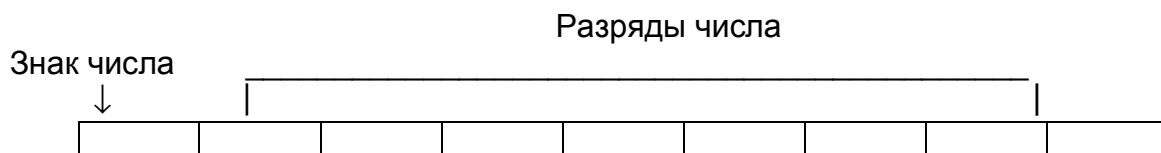


Рис. 1. Схема 8-разрядного числа с фиксированной запятой

В форме с естественной запятой 8-разрядная сетка может размещать лишь числа из диапазона $[-(1 \cdot 10^8 - 1); -1 \cdot 10^{-7}; +(1 \cdot 10^8 - 1)]$. Важно заметить, что это множество конечно. Оно покрывает числовую ось двумя участками равномерно расположенных чисел (с шагом $1 \cdot 10^{-7}$) с границами от $-(1 \cdot 10^8 - 1)$ до $-1 \cdot 10^{-7}$ и от $1 \cdot 10^{-7}$ до $1 \cdot 10^8 - 1$ (рис.2).

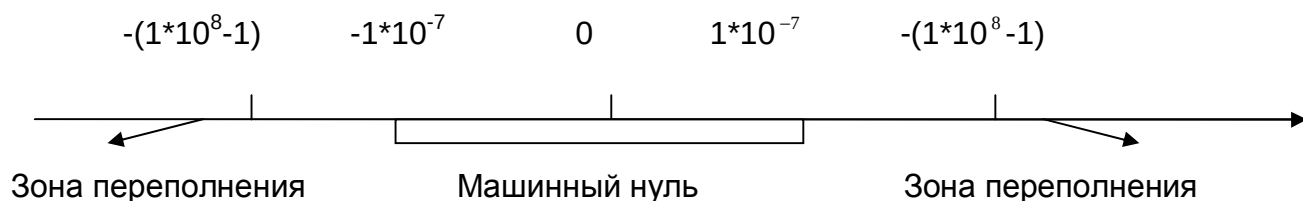


Рис. 2. Множество чисел, представимых в ЭВМ в форме с фиксированной запятой

Числа из интервала $(-1 \cdot 10^{-7}; 1 \cdot 10^{-7})$ могут быть помещены в такой разрядной сетке (кроме нуля), так что если в результате произведенной операции получится любое ненулевое число из этого интервала, то в ЭВМ оно будет отождествлено с нулем. По этой причине числа из этого интервала называют машинным нулем. Не могут быть помещены в этой сетке и числа, большие по модулю, чем $1 \cdot 10^8 - 1$. При появлении в процессе счета таких чисел возникает ситуация, называемая переполнением разрядной сетки.

В форме с плавающей запятой число представляется в виде $x = \pm M \cdot 10^p$, где M - мантисса; p - порядок числа, т.е. показатель степени, в которую нужно возвести 10, чтобы, умножив результат на мантиссу и присвоив нужный знак, получить данное число (рис. 3).

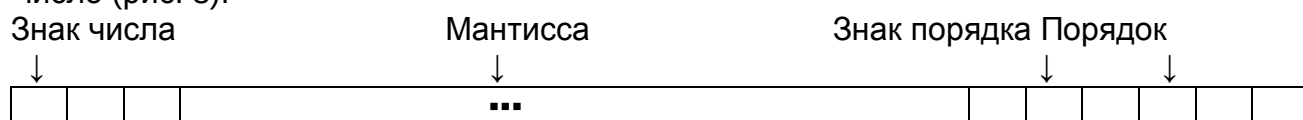


Рис. 3. Пример разрядной сетки с представлением десятичных чисел в форме с плавающей запятой

Абсолютная величина порядка при представлении, изображенном на рис. 3, может выражаться максимальным двузначным числом 99. Это дает огромные диапазоны представления действительных чисел. Важной особенностью этой системы представления чисел является неравномерность их расположения на числовой оси.

В учебной литературе часто встречается представление мантииссы в виде $M = \frac{d_1}{b} + \frac{d_2}{b^2} + \dots + \frac{d_t}{b^t}$, где β - основание системы счисления; t – количество разрядов в записи мантииссы; $d_1, d_2, \dots, d_t$ - целые числа, удовлетворяющие неравенствам $0 \leq d_i \leq \beta - 1$ ($i=1, 2, \dots, t$).

Определенная таким образом мантиисса оказывается в диапазоне $0 < M < 1$. Значение порядка p берется из интервала $[p_1, p_2]$, границы которого обусловлены способом физического представления чисел в машине. Используя эти представления можно вывести, что в системе с плавающей запятой ровно $2(\beta - 1)^{t-1} (p_2 - p_1 + 1) + 1$ чисел.

Для хранения двоичных вещественных чисел в современных ЭВМ соответствующие форматы предписаны стандартом, не зависящим от архитектуры ЭВМ и конкретных языков программирования (IEEE 754 Floating Points Numbers).

Существует несколько типов вещественных величин. Характеристики некоторых из этих типов, используемых в языке Pascal, приведены в таблице 1.

Таблица 1

Тип	Длина записи числа, байт	Диапазон значений	Количество сохраняемых значащих цифр мантииссы в десятичном представлении числа
Real	6	$2.9 \cdot 10^{-39} - 1.7 \cdot 10^{38}$	11-12
Single	4	$1.5 \cdot 10^{-45} - 3.4 \cdot 10^{38}$	7-8
Double	8	$5 \cdot 10^{-324} - 1.7 \cdot 10^{308}$	15-16
Extended	10	$3.4 \cdot 10^{-4932} - 1.1 \cdot 10^{4932}$	19-20

Опишем форматы хранения чисел типов single и double. Отметим предварительно следующее обстоятельство. В указанном выше стандарте число хранится в такой форме, что его мантиисса заключена в диапазоне $1 < M < 2$, т.е. первая цифра мантииссы всегда равна 1. Эта дополнительная операция сдвига на один разряд («нормализация») делает нормализованный порядок на единицу меньше, чем тот, который использован в предыдущем рассмотрении.

Величина типа single занимает в памяти ЭВМ 4 байта (рис. 4). Знак числа хранится в крайнем левом разряде с номером 31 (как обычно, разряды нумеруются с 0 и располагаются справа налево). Порядок в смещенной форме занимает разряды с 23 по 30. Смещенная форма находится так: к настоящему значению порядка прибавляется число 127 (так называемое «смещение») для того, чтобы значение смещенного порядка всегда было положительным. Мантиисса занимает все остальные разряды, т.е. с 0 по 22, причем целая часть мантииссы вообще не хранится (она всегда равна единице и при вычислениях это учитывается).

Величина типа double занимает в памяти ЭВМ 8 байт (рис. 5). Знак числа хранится в разряде номер 63. Порядок в смещенной форме занимает разряды с 52 по 62 (смещение в данном случае равно 1023). Мантиисса занимает разряды с 0 по 51 (как и раньше, целая часть мантииссы, автоматически равная единице, не хранится).

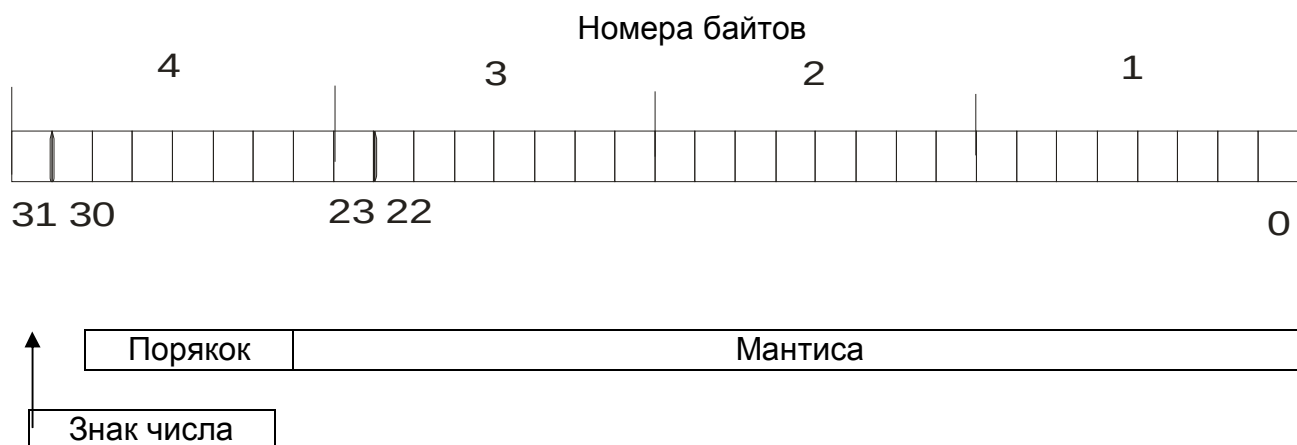


Рис. 4. Способ хранения числа типа single в ЭВМ

Опишем формат хранения числа типа double. Величина типа double занимает в памяти ЭВМ 8 байт (рис. 5). Знак числа хранится в разряде номер 63. Порядок в смещенной форме занимает разряды с 52 по 62 (смещение в данном случае равно 1023). Мантисса занимает разряды с 0 по 51, целая часть мантиссы автоматически равная единице не хранится.

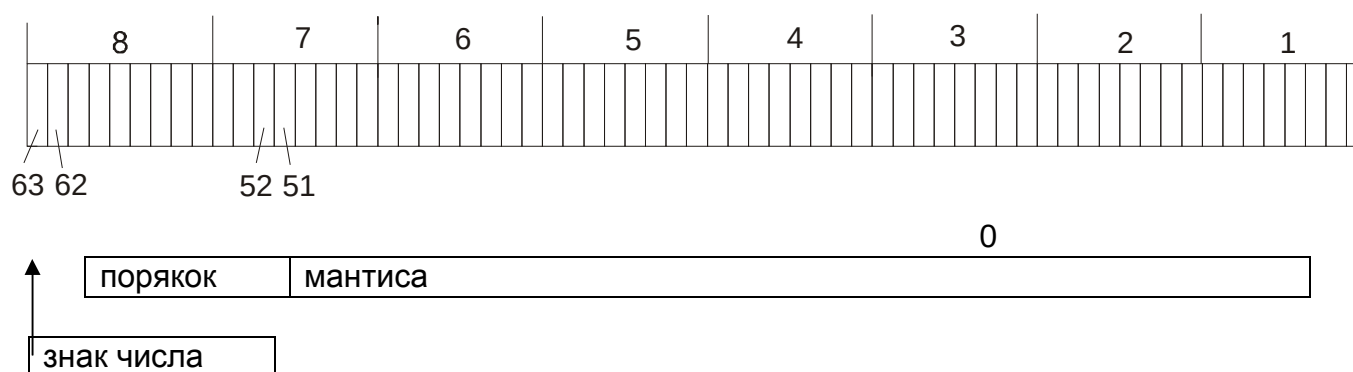


Рис. 5. Способ хранения числа типа double в ЭВМ.

2. ПОГРЕШНОСТИ ВЫЧИСЛЕНИЯ ЭЛЕМЕНТАРНЫХ ФУНКЦИЙ С УЧЕТОМ ПРЕДСТАВЛЕНИЯ В ЭВМ ВЕЩЕСТВЕННЫХ ЧИСЕЛ В РАЗЛИЧНЫХ ФОРМАТАХ ЗАПИСИ ДАННЫХ И ОСОБЕННОСТЕЙ ПРОЦЕДУРЫ ВЫВОДА WRITELN В ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PASCAL.

Характеристики некоторых используемых в языке программирования Pascal типов вещественных величин приведены в таблице 1: длина записи в байтах, диапазон значений, количество сохраняемых значащих цифр мантиссы в десятичном представлении числа. Будем считать, что вычисляемое значение имеет наибольшую точность (что вполне естественно). Из таблицы 1 следует, что при использовании величины типа real в мантиссе лишь 12 верных значащих цифр, типа double - 17, типа single - 7 (верные цифры подчеркнуты, а сомнительные выделены жирным шрифтом). Возможности стандартной процедуры вывода в языке Borland Pascal writeln позволяют сохранить в мантиссе отображения лишь 18 значащих цифр. Поэтому при использовании типа extended не все верные цифры результата выводятся процедурой writeln. Поэтому в программе использован

форматированный вывод с заведомо большим, чем необходимо, количеством значащих цифр, чтобы не потерять в отображении результата части цифр мантииссы.

ПРОГРАММА ОЦЕНКИ ТОЧНОСТИ ВЫЧИСЛЕНИЯ ЭЛЕМЕНТАРНЫХ ФУНКЦИЙ В СИСТЕМЕ ПРОГРАММИРОВАНИЯ BORLAND PASCAL

```
program s;  
uses crt;  
var x,y: double;  
begin  
clrscr;  
writeln('vvedite x');  
read(x);  
writeln('x=',x:6:5);  
y:=sqrt(x-1)*cos(x);  
writeln('sqrt(x-1)*cos(x)=',y:40);  
repeat until keypressed  
end.
```

РЕЗУЛЬТАТЫ РАБОТЫ

Vvedite x
2.5732
double: sqrt(x-1)*cos(x)= -1.05705996827185778E+0000
single: sqrt(x-1)*cos(x)= -1.05706000328063965E +0000
rea1: sqrt(x-1)*cos(x)= -1.05705996827055060E +0000
extended: sqrt(x-1)*cos(x)= -1.05705996827185794E +0000
Верные цифры мантиисс подчеркнуты
Сомнительные цифры мантииссы выделены

3. НАХОЖДЕНИЕ МАШИННОГО ЭПСИЛОН.

Одной из важных характеристик точности арифметики числа с плавающей запятой ЭВМ является значение машинного эпсилон, т.е. такого наименьшего числа e_i из системы чисел с плавающей запятой, что добавление его к единице дает на этой ЭВМ значение, большее единицы. Величина e_M играет в вычислениях на ЭВМ фундаментальную роль; ее называют *относительной точностью ЭВМ*, а также *машинной точностью* (или *машинным эпсилоном*). Всюду в дальнейшем e_M — это относительная точность ЭВМ. Заметим, что значение этой величины определяется разрядностью мантииссы и способом округления.

Важно с самого начала иметь четкое представление о том, что почти наверняка в представимом множестве ЭВМ нет числа y , являющегося решением поставленной задачи. Лучшее, что можно попытаться сделать, — это найти его представление $y^* = fl(x)$ с относительной точностью порядка e_M .

Поскольку числа, по абсолютной величине меньшие $e_i X_0 = e_i 2^{-(p_{\max}+1)}$, где $p_{\max} = 2^{l+1} - 1$ — максимально возможный порядок числа с плавающей запятой, имеющий в разрядной сетке данной ЭВМ конечное число $(l+1)$ двоичных цифр, не входят в множество представимых чисел этой ЭВМ, то все эти числа автоматически заменяются нулём с абсолютной погрешностью $e_i 2^{-(p_{\max}+1)}$ и относительной погрешностью e_i . Числа, по абсолютной величине большие $e_i X_0 = e_i 2^{-(p_{\max}+1)}$, входят в множество представимых чисел этой ЭВМ, но они также представляются с ошибкой округления, вызванной отбрасыванием двоичных разрядов мантииссы, номера которых превышают номер двоичного разряда, соответствующего e_i .

Поэтому важно исследовать зависимость машинного эпсилон e_i от типов вещественных величин. типов вещественных величин (форматов представления данных), приведенных в таблице 1.

Значение машинного эпсилон для конкретной ЭВМ можно определить программным путем. Полагая $e^{(0)} = 1$, следует вычислять последовательно $e^{(1)} = 0.5e^{(0)}$, $e^{(2)} = 0.5e^{(1)}$, ..., $e^{(n)} = 0.5e^{(n-1)}$, ..., проверяя каждый раз выполнение неравенства $1 \oplus e^{(n)} > 1$. Как только при некотором n окажется, что $1 \oplus e^{(n)} = 1$, следует положить $e_M = e^{(n-1)}$ перейти к следующему этапу вычислений. Хотя полученное таким способом значение может отличаться от e_M в 2 раза, обычно оно используется так, что эта погрешность не имеет значения. Блок-схема алгоритма представлена на рис. 6. Далее приводится программа нахождения машинного эпсилон e_M в различных форматах представления данных. В программе использован форматированный вывод eps:40 с заведомо большим, чем необходимо, количеством цифр, чтобы не потерять в отображении результата части цифр мантиисы. Показано, что $e_M = 5.42101086242752217E-0020$ одинаково во всех форматах представления вещественных чисел в языке Pascal single, rea1, double, extended.

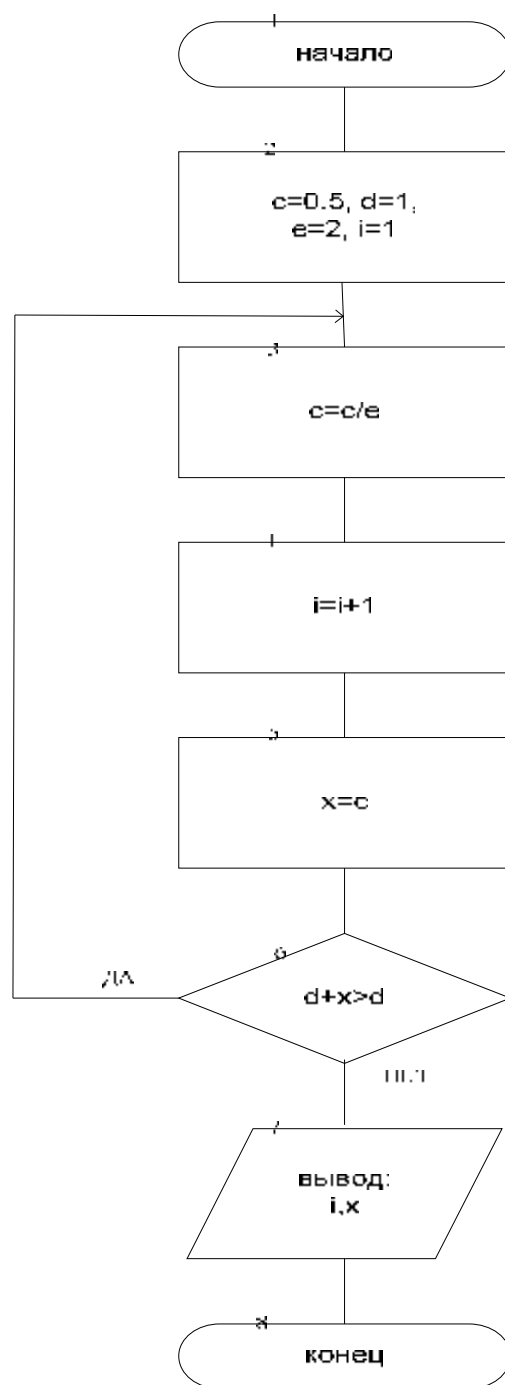


Рис. 6. Блок-схема алгоритма нахождения машинного эпсилон

ПРОГРАММА НАХОЖДЕНИЯ МАШИННОГО ЭПСИЛОН В СИСТЕМЕ ПРОГРАММИРОВАНИЯ BORLAND PASCAL.

```
program epsilon;
uses crt;
const c:single=0.5; d:single=1; e:single=2;
var a,eps:single; i:integer;
begin
clrscr;
a:=d; eps:=c; i:=1;
while a+eps>a do
begin
eps:=eps/e; i:=i+1;
end;
writeln('i=',i,'eps:single, mashinnoe epsilon=',eps:40);
repeat until keypressed
end.
```

i=64eps: single, машинное эпсилон=5.42101086242752217E-0020

Это число одинаково во всех форматах представления данных.

4. АБСОЛЮТНАЯ И ОТНОСИТЕЛЬНАЯ ПОГРЕШНОСТИ

Простейшей количественной мерой ошибки является *абсолютная погрешность*

$$\Delta(x^*) = |x - x^*|,$$

где x - точное значение некоторой величины, x^* - известное приближенное значение той же величины (*приближенное число*). *Ошибкой* (или *погрешностью*) приближенного числа x^* называют разность $x - x^*$ между точным и приближенным значениями.

Чтобы соотнести погрешность величины и ее значение, вводится понятие *относительной погрешности* (при $x \neq 0$)

$$d(x^*) = \frac{|x - x^*|}{|x|} = \frac{\Delta(x^*)}{|x|}$$

Относительные погрешности не зависят от масштабов величин и единиц измерения.

Вычислим абсолютную и относительную погрешности числа $x^*=2.5732$, заданного всеми своими верными цифрами в строгом (узком) смысле. Абсолютная погрешность $\Delta(x^*)=0,00005$. Относительная погрешность

$$d(x^*) = \frac{\Delta(x^*)}{x^*} = \frac{0.00005}{2.5732} = 1.94310586040802E-0005$$

Вычислим эти погрешности с помощью программирования в системе Borland Pascal и в MATHCAD(e).

4.1. РЕШЕНИЕ С ПОМОЩЬЮ ПРОГРАММИРОВАНИЯ В СИСТЕМЕ BORLAND PASCAL

```
program pogrchisla;
uses crt;
var x,dx,deltax: real;
begin
clrscr;
writeln('vvedite x');
read(x);
writeln('vvedite dx');
read(dx);
writeln('x=',x:5:5);
writeln('dx=',dx:5:5);
deltax:=dx/x;
writeln('deltax=',deltax);
repeat until keypressed
end.
```

Результат

```
Введите x      2.5732

Введите dx     0.00005

deltax=        1.94310586040802E-0005
```

4.2. РЕШЕНИЕ В MATHCAD(e)

$x := 2.5732$

$\Delta x := 0.00005$

$\delta(x) := \frac{\Delta x}{x}$

$\delta(x) = 1.943 \times 10^{-5}$

5. ПРАВИЛЬНАЯ ЗАПИСЬ И ОКРУГЛЕНИЕ ЧИСЕЛ

При округлении числа x^* возникает погрешность, называемая погрешностью округления: $\Delta_{\text{окр}} = |x^* - x_1|$, где x^* – данное число, x_1 – результат округления. Так как число $x^* = 2.5732$ задано всеми своими верными значащими цифрами в широком смысле, то абсолютная погрешность числа x^* $\Delta(x^*) = 0.0001$. Поэтому абсолютная погрешность округленного числа Δx_1 вычисляется по формуле

$$\Delta x_1 = \Delta(x^*) + \Delta x_{\text{вв}}.$$

Округлим число $x^* = 2.5732$ до 3 значащих цифр. Тогда округлённое значение $x_1 = 2.57$ и погрешность округления $\Delta_{\text{окр}} = 0.01$. Выполним округление числа $x^* = 2.5732$ с помощью программирования в системе Borland Pascal и в MATHCAD(e).

5.1. ПРОГРАММА ОКРУГЛЕНИЯ ЧИСЕЛ В СИСТЕМЕ ПРОГРАММИРОВАНИЯ BORLAND PASCAL.

```
program okrugchislo;
uses crt;
var x,x1,deltax,deltaokr,deltax1,otndeltax:real;
begin
clrscr;
writeln('Введите число x');
read(x);
writeln('Введите округленное число x1');
read(x1);
writeln('Введите абсолютную погрешность числа deltaokr');
read(deltax);
deltaokr:=abs(x-x1);
writeln('Погрешность округления deltaokr=',deltaokr:7:6);
deltax1:=deltax+deltaokr;
writeln('абсолютная погрешность округленного числа deltaokr= ', deltax1:7:6);
otndeltax:=deltax1/x1;
writeln('относительная погрешность округленного числа otndeltax=',otndeltax:7:6);
end.
```

Введите число x

2.5732

Введите округленное число x1

2.57

Введите абсолютную погрешность числа deltaokr

0.0001

Погрешность округления deltaokr=0.003200

абсолютная погрешность округленного числа deltaokr= 0.003300

относительная погрешность округленного числа otndeltax=0.001284

5.2. РЕШЕНИЕ ЗАДАЧИ ОКРУГЛЕНИЯ В MATHCAD(e).

Округлить число $x=2.5732$ до трех значащих цифр, найти абсолютную и относительную погрешности полученных приближений. Цифры верны в широком смысле.

$$x := 2.5732$$

Округленное число $x_1=2.57$

$$x_1 := 2.57$$

По определению верной цифры числа в широком смысле абсолютная погрешность $\Delta(x^*) = 0.0001$

$$\Delta x := 0.0001$$

Погрешность округления определяется по формуле $\Delta \hat{\epsilon} = |x - x_1|$

$$\Delta_{\text{окр}} := |x - x_1| \quad \Delta_{\text{окр}} = 3.2 \times 10^{-3}$$

Абсолютная погрешность округленного числа определяется по формуле $\Delta x_1 = \Delta x + \Delta \hat{\epsilon}$

$$\Delta x_1 := \Delta x + \Delta_{\text{окр}} \quad \Delta x_1 = 3.3 \times 10^{-3}$$

$$\Delta x_1 = 3.3 \times 10^{-3} < 1 \cdot 10^{-2}$$

Это означает, что в числе x три цифры (2,5,7) верны в строгом смысле по абсолютной погрешности

Относительная погрешность округленного числа определяется по формуле

$$\delta x_1 = \Delta x_1 / x_1$$

$$\delta x_1 := \frac{\Delta x_1}{x_1} \quad \delta x_1 = 1.284 \times 10^{-3}$$

$$\delta x_1 = 1.284 \times 10^{-3} < 0.5 \cdot 10^{-2}$$

Это означает, что в числе x три цифры (2,5,7) верны в строгом смысле по относительной погрешности

5.3. ПРОГРАММНОЕ ОКРУГЛЕНИЕ ДЕСЯТИЧНЫХ ЧИСЕЛ ПО ЗАДАННОМУ ЗНАЧЕНИЮ АБСОЛЮТНОЙ ПОГРЕШНОСТИ

В некоторых языках программирования имеется функция $ROUND(x, n)$, где x - арифметическое выражение, задающее округляемое значение, а n - арифметическое выражение, задающее уровень округления. Округление с помощью функции $ROUND$ выполняется следующим образом:

- а) при $n > 0$ - происходит округление до n -й цифры после запятой;
- б) при $n = 0$ - округление до ближайшего целого;
- в) при $n < 0$ - округление до $(n+1)$ -й цифры влево от десятичной точки.

Блок-схема алгоритма округления десятичных чисел по заданному значению абсолютной погрешности приведена на рис. 7.

При отсутствии функции $ROUND$ она может быть смоделирована с помощью функции Int — «целая часть»:

$$ROUND(a, n) = \frac{sign(a)Int(|a|10^n + 0.5)}{10^n}$$

где $Sign$ — «знаковая функция», равная 1 для положительного значения аргумента, -1 для отрицательного и 0 для нулевого. Используя эту формулу, можно программным путем вести округление как по заданному уровню округления и, так и по заданной величине абсолютной погрешности.

На рис. 7 изображена блок-схема алгоритма округления числа a с предварительным определением уровня округления n по заданной величине абсолютной погрешности $\Delta(a^*)$.

При всей простоте этого алгоритма при машинном округлении могут возникать некоторые технические проблемы, часто не позволяющие получать достоверного результата. Коснемся здесь этих проблем, поскольку процедура округления десятичных чисел с последующим их выводом верными цифрами в естественной форме записи в дальнейшем будет применяться неоднократно. Ниже приведена программа *okruglenie*, которая реализует симметрическое округление десятичных чисел, оставляя лишь верные (в широком смысле) цифры в соответствии с заданной абсолютной погрешностью числа. Для вывода значения округленного числа в программе умышленно сохранена машинная (нормальная) форма представления десятичных чисел, без «усечения» неверных знаков.

В программе используется модуль *okrugl*, который, в свою очередь, обращается к процедуре *окр*. В основе этой процедуры лежит алгоритм, изображенный на рис. 9. Первый ее аргумент - подлежащее округлению число a , второй - абсолютная погрешность d . Результаты работы процедуры — параметры-переменные b и n ; первый из них - округленное число, второй показывает, на сколько знаков происходит округление (при $d > 1$ округление происходит до n -й цифры до десятичной точки, при $d < 1$ - до n -й цифры после десятичной точки). В ходе исполнения программы вначале определяется количество оставляемых в числе значащих цифр n (по разным схемам при $d < 1$ и $d > 1$) и затем выполняется округление по указанной выше формуле. В силу отсутствия в языке Pascal операции возведения в произвольную степень, для вычисления 10^n используется функция *stepen10*. Конструкция *type T* позволяет легко менять тип данных (делать это следует одновременно в модуле и в программе).

Нуждается в комментарии следующее различие между схемой на рис. 9 и программой: в циклах вместо условий $d < 10^{-n}$ и $d \geq 10^{-n}$ используются модифицированные условия $d < (10^{-n} + e)$ и $d \geq (10^{-n} + e)$ соответственно. Это связано с особенностями операций над вещественными числами: при вычислении 10^{-n} (при $n > 0$) в ЭВМ получается *приближенное* значение, и для правильной работы программы следует подстраховаться путем добавления e (точное значение e особой роли не играет, в программе положено $1 \cdot 10^{-10}$). Если этого не сделать, то при округлении числа 1.23456 при погрешностях 0.1 и 0.2 получится одинаковое округленное значение числа, в то время как оно в соответствии с правилом определения верных цифр должно быть разным.

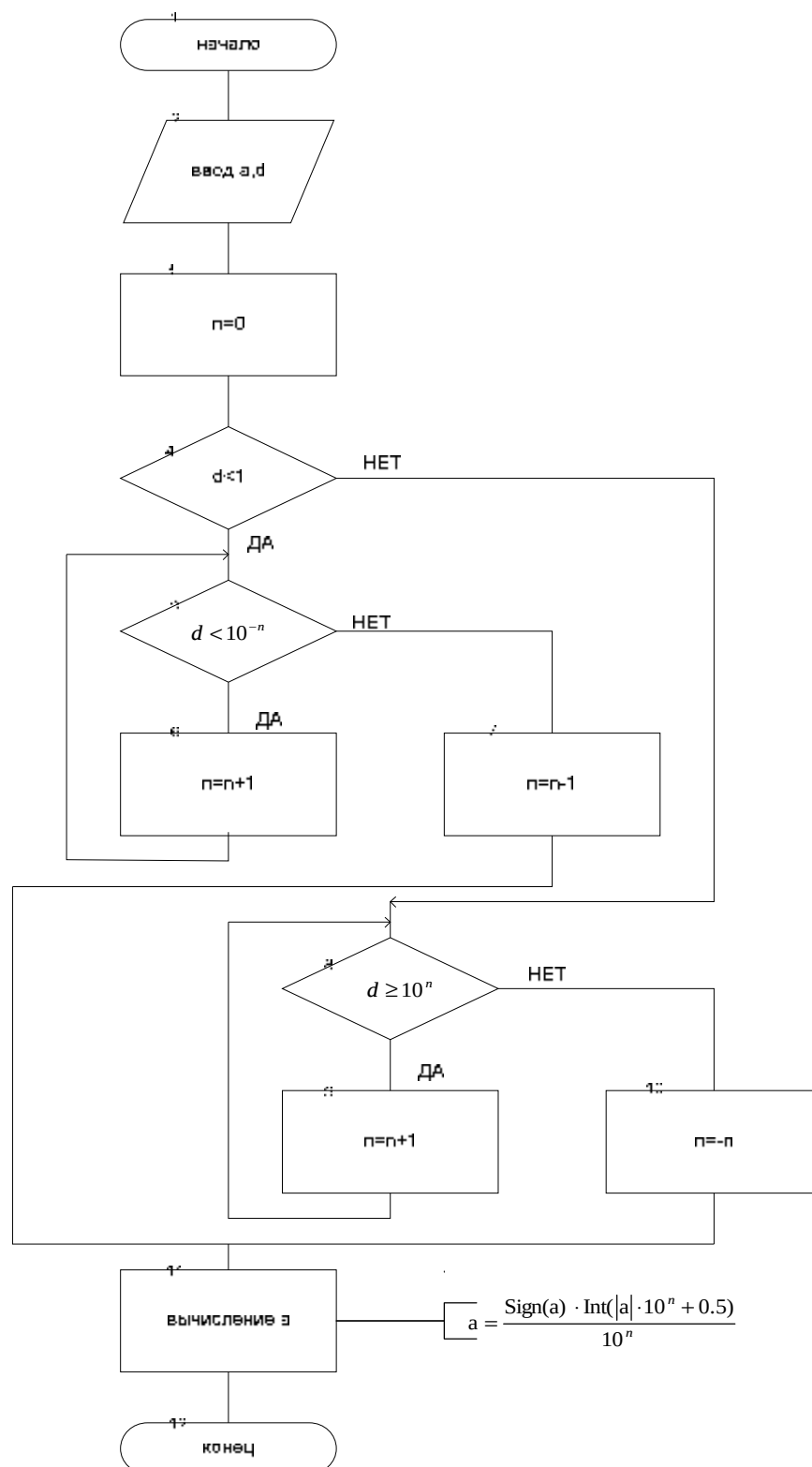


Рис. 7. Блок-схема алгоритма округления чисел по заданной величине абсолютной погрешности $\Delta(x^*)$.

Вывод значений уровня округления n позволяет, в частности, избежать ошибки в особых случаях - например, при округлении целых чисел, в изображении которых оказываются значащие нули, расположенные между значащими цифрами. При использовании программы `okruglenie`, когда в результате округления значащие и незначащие нули оказываются рядом и для их разделения используется значение n .

Выполним программное округление для числа $x = 2.5732$ с погрешностью $e = 0.0033$

ПРОГРАММА ОКРУГЛЕНИЯ ЧИСЕЛ НА ЯЗЫКЕ BORLAND PASCAL

```
Program okruglenie;
uses okrugl,crt;
type T=real;
var a,d,b:T; n,m:integer;
begin
clrscr;
writeln;
writeln('введите число и погрешность'); read(a,d);
okr(a,d,a,n);
if a<0 then
begin
if n<0 then m:=0
else m:=n;
writeln('Округленное число a=',a-1E-17:17:m, ' n=',n);
end
else
begin
if n<0 then m:=0
else m:=n;
writeln('Округленное число a=',a+1E-17:17:m, ' n=',n);
end;
repeat until keypressed
end.
```

В программе используется модуль `okrugl`, который, в свою очередь, содержащий процедуру `okr`:

```
unit okrugl;
interface
type T=real;
procedure okr(a,d:T; var b:T; var n:integer);
implementation
function stепен 10(n:integer):T;
var i:integer; k:longint;
begin k:=10; for i:=1 to (abs(n)-1) do k:=k*10;
if n=0 then stепен 10:=1;
if n>0 then stепен 10:=k;
if n<0 then stепен 10:=1.0/k;
end;
procedure okr(a,d:T; var b:T; var n:integer);
var i,k:integer;
begin
```

```

n:=1;
if d<1 then
begin
while d<((1.0/stepen10(n))+1E-10)
do n:=n+1;
n:=n-1
end
else
begin
while d>=(stepen10(n)+1E-10)
do n:=n+1;
n:=-n
end;
b:=int((abs(a)*stepen10(n))+0.5)/stepen10(n);
if a <> 0 then b:=b*a/abs(a)
end;
end.

```

Результаты работы

введите число и погрешность

2.5733 0.0033

Округленное число a=2.57

n=2

6. ВЫЧИСЛЕНИЕ АБСОЛЮТНОЙ И ОТНОСИТЕЛЬНОЙ ПОГРЕШНОСТИ ФУНКЦИИ.

Абсолютной погрешностью Δx приближенного числа x^* называется выражение: $\Delta x = |x - x^*|$. Относительной погрешностью δx приближенного числа называется отношение абсолютной погрешности к абсолютному значению приближения x :

$$\delta x = \frac{\Delta x}{|x|}.$$

Абсолютная погрешность функции $y = f(x)$ вычисляется по формуле:

$\Delta f = |f'(x)| \Delta x$, а относительная погрешность функции – по формуле

$$df(x) = \frac{|f'(x)| \cdot |x|}{|f(x)|} \cdot d(x).$$

Оценим абсолютную и относительную погрешность вычисления функции $y = \sqrt{x-1} * \cos(x)$ при $x^* = 2.5732$, заданного всеми своими верными значащими цифрами в строгом смысле, в системе программирования Borland Pascal и в Mathcad. При этом абсолютная погрешность оценивается как с использованием формулы $\Delta f = |f'(x)| \Delta x$, так и с использованием приращения функции.

6.1. ПРОГРАММА ВЫЧИСЛЕНИЯ АБСОЛЮТНОЙ И ОТНОСИТЕЛЬНОЙ ПОГРЕШНОСТИ ФУНКЦИИ В СИСТЕМЕ ПРОГРАММИРОВАНИЯ BORLAND PASCAL.

```
program funksiya;
uses crt;
var x,x0,x1,dx,dx1,y0,y1,dyabs,dyotn,dyabs1,dyotn1,dxotn:real;
function y(x:real):real;
begin
y:=sqrt(x-1)*cos(x);
end;
begin
clrscr;
x0:=2.5732;
x1:=2.573;
dx:=0.0001/2;
y0:=y(x0);
writeln('x0=',x0:11:6,' ',y(x0)=',y0:11:6);
y1:=y(x1);
writeln('x1=',x1:11:6,' ',y(x1)=',y1:11:6);
dx1:=abs(x1-x0)+dx;
writeln(' абсолютная погрешность x: dx1=',dx1:11:6);
dxotn:=(dx1)/x1;
writeln('относительная погрешность x=',dxotn:11:6);
dyabs1:=abs(1/(2*sqrt(x0-1))*cos(x0)-sin(x0)*sqrt(x0-1))*dx1;
writeln('абсолютная погрешность функции (1способ) dyabs1=',dyabs1:11:6);
dyotn1:=dyabs1/abs(y0);
writeln('относительная погрешность функции (1 способ) dyotn1=',dyotn1:11:6);
dyabs:=abs(y0-y1);
writeln('абсолютная погрешность функции (2 способ) dyabs =', dyabs:11:6);
dyotn:=dyabs/abs(y1);
writeln('относительная погрешность функции (2 способ) dyotn =', dyotn:11:6);
repeat until keypressed;
end.
```

Результаты работы.

```
x0=2.573200   y(x0)= -1.057060
x1=2.573000   y(x1)= -1.0569
абсолютная погрешность x: dx1=0.000250
относительная погрешность x= 0.00009716
абсолютная погрешность функции (1способ) dyabs = 0.000253
относительная погрешность функции (1 способ) dyotn1= 0.000239
абсолютная погрешность функции (2 способ) dyabs = 0.000202
относительная погрешность функции (2 способ) dyotn = 0.000191
```

6.2. РЕШЕНИЕ В MATHCAD(E)

Исходная функция: $y(x) := \sqrt{x-1} * \cos(x)$

$x := 2.5732$

округлим число до 4 значащих цифр: $x1 := 2.573$

найдем абсолютную погрешность: $\Delta x := \frac{0.0001}{2}$ $\Delta x = 0.00005$

найдем погрешность округления: $x_{okr} := |x - x1|$ $x_{okr} = 0.0002$

Погрешность округления числа равна сумме погрешности исходного числа и погрешности округления:

$\Delta x1 := x_{okr} + \Delta x$ $\Delta x1 = 0.00025$

найдем относительную погрешность x : $dx := \frac{\Delta x1}{x1}$ $dx = 0.00009716$

найдем абсолютную погрешность функции: $\Delta y(x) := \left| \frac{d}{dx} y(x) \right| \Delta x1$

$\Delta y(x) = 0.000253$

найдем относительную погрешность функции: $dy(x) := \frac{\Delta y(x)}{|y(x)|}$

$dy(x) = 0.000239$

$y(x) = -1.05706$

$y(x1) = -1.0569$

Другой способ нахождения абсолютной и относительной погрешности:

$\Delta y := |y(x) - y(x1)|$ $\Delta y = 0.000202$

$d y1 := \frac{\Delta y}{|y(x1)|}$ $d y1 = 0.000191$

7. ОЦЕНКА ПОГРЕШНОСТИ ФУНКЦИИ НЕСКОЛЬКИХ ПЕРЕМЕННЫХ

Абсолютная погрешность этой функции нескольких переменных определяется по формуле:

$$\Delta f = \sum_{i=1}^n \left| \frac{\partial f}{\partial x_i}(x) \right| \Delta x_i.$$

Относительная погрешность этой функции вычисляется следующим образом:

$$d f = \frac{\Delta f}{|f(x_1, x_2, \dots, x_n)|} = \sum_{i=1}^n \left| \frac{\partial f}{\partial x_i}(x) \frac{1}{f} \right| \Delta x_i = \sum_{i=1}^n \left| \frac{\partial \ln f}{\partial x_i}(x) \right| \Delta x_i.$$

Пример 8. Найти абсолютную и относительную погрешность вычислений объёма цилиндра по формуле $V = \frac{\pi d^2}{4} h$, если $d = 4,5 \pm 0,05$ см., $h = 10 \pm 0,08$ см., $\pi^* = 3,14$.

Решение. Будем считать точным значением числа π его приближение, взятое с точностью до 6 знаков $\pi = 3,141592$. Тогда абсолютная погрешность $\Delta \pi = \pi - \pi^* = 0,001592$. Рассматривая d, h, π как переменные величины, находим частные производные

$$\frac{\partial V}{\partial d} = \frac{\pi d h}{2} = \frac{3,14 \cdot 4,5 \cdot 10}{2} = 70,69;$$

$$\frac{\partial V}{\partial h} = \frac{\pi d^2}{4} = \frac{\pi \cdot 4,5^2}{4} = 15,896;$$

$$\frac{\partial V}{\partial \pi} = \frac{h d^2}{4} = \frac{10 \cdot 4,5^2}{4} = 50,625.$$

По формуле вычисления абсолютной погрешности функции трех переменных

$$\Delta V = \left| \frac{\partial V}{\partial d} \right| \Delta d + \left| \frac{\partial V}{\partial h} \right| \Delta h + \left| \frac{\partial V}{\partial \pi} \right| \Delta \pi$$

находим абсолютную погрешность объёма

$$\Delta V = \left| \frac{\partial V}{\partial d} \right| \Delta d + \left| \frac{\partial V}{\partial h} \right| \Delta h + \left| \frac{\partial V}{\partial \pi} \right| \Delta \pi = 70,69 \cdot 0,05 + 15,896 \cdot 0,08 + 50,625 \cdot 0,001592 = 4,887 \approx 4,89$$

$$\text{Тогда } V = \frac{\pi d^2 h}{4} \approx (158,96 \pm 4,89) \text{ см}^3.$$

Относительная погрешность определения объёма

$$\frac{\Delta V}{V} = \frac{4,887}{158,96} = 0,031 = 3,1\%$$

Выполним эти вычисления в MATHCAD(e).

Найти абсолютную и относительную погрешность вычислений объёма

цилиндра по формуле $V = \frac{\pi d^2 h}{4}$, если $d = 4,5 \pm 0,05$ см., $h = 10 \pm 0,08$ см.,

$$\pi^* = 3,14$$

$$v(d, h, \pi) := \frac{\pi \cdot d^2 \cdot h}{4}$$

Находим частные производные

$$\frac{d}{dd}v(d, h, \pi) \rightarrow \frac{1}{2} \cdot \pi \cdot d \cdot h \quad v1(d, h, \pi) := \frac{1}{2} \cdot \pi \cdot d \cdot h$$

$$\frac{d}{dh}v(d, h, \pi) \rightarrow \frac{1}{4} \cdot \pi \cdot d^2 \quad v2(d, h, \pi) := \frac{1}{4} \cdot \pi \cdot d^2$$

$$\frac{d}{d\pi}v(d, h, \pi) \rightarrow \frac{1}{4} \cdot d^2 \cdot h \quad v3(d, h, \pi) := \frac{1}{4} \cdot d^2 \cdot h$$

Будем считать точным значением числа π его приближение, взятое с точностью до 6 знаков после запятой $\pi = 3.141592$. Абсолютная погрешность числа $\Delta\pi = \pi - \pi_a = 0.001592$. Абсолютные погрешности $\Delta d = 0.05$; $\Delta h = 0.08$.

$$d := 4.5 \quad h := 10 \quad \pi_a := 3.14$$

$$\Delta d := 0.05 \quad \Delta h := 0.08 \quad \Delta\pi_a := 0.001592$$

$$v := \frac{\pi d^2 h}{4} \quad v = 159.043$$

Вычислим значения частных производных

$$v1(d, h, \pi) = 70.686 \quad v2(d, h, \pi) = 15.904 \quad v3(d, h, \pi) = 50.625$$

Абсолютная погрешность вычисления объема

$$\Delta v := |v1(d, h, \pi)| \cdot \Delta d + |v2(d, h, \pi)| \cdot \Delta h + |v3(d, h, \pi)| \cdot \Delta\pi_a$$

$$\Delta v = 4.887 \text{ см}^3$$

Абсолютная погрешность вычисления объема

$$\Delta v := |v_1(d, h, \pi)| \cdot \Delta d + |v_2(d, h, \pi)| \cdot \Delta h + |v_3(d, h, \pi)| \cdot \Delta \pi$$

$$\Delta v = 4.887 \text{ см}^3$$

Тогда

$$v \approx 159.043 + 4.887$$

Относительная погрешность вычисления объёма

$$\delta v := \frac{\Delta v}{v} \quad \delta v = 0.031 \quad \delta v = 0.031 = 3.1\%$$

8. ОЦЕНКА ПОГРЕШНОСТИ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ

Формулы для оценки абсолютных и относительных погрешностей арифметических операций: сложения, вычитания, умножения, деления, возведения в степень и извлечения корня приведены в таблице 2.

Таблица 2

Арифметические операции	Абсолютная погрешность	Относительная погрешность
$x + y$	$\Delta x + \Delta y$	$\frac{ x }{ x+y } dx + \frac{ y }{ x+y } dy$
$x - y$	$\Delta x + \Delta y$	$\frac{ x }{ x-y } dx + \frac{ y }{ x-y } dy$
$x \cdot y$	$ x \Delta y + y \Delta x$	$dx + dy$
x/y	$\frac{ x \Delta y + y \Delta x}{y^2}$	$dx + dy$
x^n	$n x^{n-1} \Delta x$	$n dx$
$\sqrt[n]{x}$	$\frac{1}{n x^{\frac{n-1}{n}}} \Delta x$	$\frac{dx}{n}$

Вычислим абсолютные и относительные погрешности арифметических операций (сложения, умножения, деления, вычитания, возведения в степень и извлечения корня) с числами $a=0.7219$ и $b=135.347$, заданными всеми своими

значащими цифрами в широком смысле в системе программирования Borland Pascal и в Mathcad.

8.1. РЕШЕНИЕ В СИСТЕМЕ ПРОГРАММИРОВАНИЯ BORLAND PASCAL

```
uses crt;
var a,a1,da,d1a,b,db,d1b,c,dc,dc1,e,e1,e2,g,g1,g2,f,f1,f2,h,h1,h2,p,p1,p2:real;
function dsum(da,db:real):real;
begin
dsum:=da+db; end;
function dmult(a,da,b,db:real):real;
begin dmult:=abs(a)*db+abs(b)*da; end;
function dquot(a,da,b,db:real):real;
begin dquot:=(abs(a)*db+abs(b)*da)/sqr(b);
end;
function d1sum(a,d1a,b,d1b:real):real;
begin d1sum:=abs(a)*d1a/abs(a+b)+abs(b)*d1b/abs(a+b);
end;
function d1mult(d1a,d1b:real):real;
begin d1mult:=d1a+d1b; end;
begin
clrscr;
a:=0.7219;
da:=0.0001;
b:=135.347;
db:=0.001;
d1a:=da/a;
d1b:=db/b;
writeln('a=',a:1:8);
writeln('абсолютная погрешность a da=',da:1:8);
writeln('относительная погрешность a d1a=',d1a:1:8);
writeln('b=',b:1:8);
writeln('абсолютная погрешность b db=',db:1:8);
writeln('относительная погрешность b d1b=',d1b:1:8);
writeln;
c:=a*b;
writeln('произведение a*b=',c:1:8);
dc:=dmult(a,da,b,db);
writeln('абсолютная погрешность произведения d(a*b)=',dc:1:8);
dc1:=d1mult(d1a,d1b);
writeln('относительная погрешность d1(a*b)=',dc1:1:8);
writeln;
e:=a/b;
writeln('частное a/b=',e:1:8);
e1:=dquot(a,da,b,db);
writeln('абсолютная погрешность d(a/b)=',e1:1:8);
e2:=d1mult(d1a,d1b);
writeln('относительная погрешность частного d1(a/b)=',e2:1:8);
writeln;
f:=a+b;
writeln('сумма a+b=',f:1:8);
f1:=dsum(da,db);
```

```

writeln('абсолютная погрешность суммы d(a+b)=' ,f1:1:8);
f2:=d1sum(a,d1a,b,d1b);
writeln('относительная погрешность суммы d1(a+b)=' ,f2:1:8);
writeln;
g:=a-b;
writeln('разность a-b=' ,g:1:8);
g1:=dsum(da,db);
writeln('абсолютная погрешность разности d(a-b)=' ,g1:1:8);
g2:={g1/abs(g)}d1sum(a,d1a,-b,d1b);
writeln('относительная погрешность разности d1(a-b)=' ,g2:1:6);
writeln;
h:=a*a*a;
writeln('куб числа a a*a*a=' ,h:1:8);
h1:=3*a*a*da ;
writeln('абсолютная погрешность куба числа a d(a*a*a)=' ,h1:1:8);
h2:=h1/h;
writeln('относительная погрешность куба числа a d1(a*a*a)=' ,h2:1:8);
writeln;
p:=sqrt(a);
writeln('корень квадратный sqrt(a)=' ,p:1:8);
p1:=1/(2*sqrt(a))*da;
writeln('абсолютная погрешность корня квадратного dsqrt(a) =' ,p1:1:8);
p2:=p1/p;
writeln('относительная погрешность корня квадратного dsqrt(a)=' ,p2:1:6);
writeln;
end.

```

Результаты работы программы

```

a=0.72190000
абсолютная погрешность a da=0.00010000
относительная погрешность a d1a=0.00013852
b=135.34700000
абсолютная погрешность b db=0.00100000
относительная погрешность b d1b=0.00000740

произведение a*b=97.706999
абсолютная погрешность произведения d(a*b)=0.01423660
относительная погрешность d1(a*b)=0.00014592

частное a/b=0.0053337
абсолютная погрешность d(a/b)=0.00000078
относительная погрешность частного d1(a/b)=0.00014592

сумма a+b=136.06890000
абсолютная погрешность суммы d(a+b)=0.00110000
относительная погрешность суммы d1(a+b)=0.00000810

разность a-b=-134.62510000
абсолютная погрешность разности d(a-b)=0.00110000
относительная погрешность разности d1(a-b)=0.000008

куб числа a a*a*a=0.37621068

```

абсолютная погрешность куба числа a $d(a^3)=0.00015634$
 относительная погрешность куба числа a $d1(a^3)=0.00041557$

корень квадратный $\sqrt{a}=0.84964699$
 абсолютная погрешность корня квадратного $dsqrt(a) = 0.00005885$
 относительная погрешность корня квадратного $dsqrt(a)=0.000069$

8.2 РЕШЕНИЕ В MATHCAD(e).

ПОГРЕШНОСТИ АРИФМЕТИЧЕСКИХ ДЕЙСТВИЙ

Задание. Найти абсолютные и относительные погрешности результатов арифметических действий с числами $a=0.7219$ и $b=135.147$, заданных всеми своими верными значащими цифрами в широком смысле

$$a := 0.7219$$

$$b := 135.147$$

$$\Delta a := 0.0001$$

$$\Delta b := 0.001$$

$$\delta a := \frac{\Delta a}{a}$$

$$\delta a = 0.000139$$

$$\delta b := \frac{\Delta b}{b}$$

$$\delta b = 0.000007$$

1) Сложение

$$c := a + b$$

$$c = 136.0689$$

$$\Delta c := \Delta a + \Delta b$$

$$\Delta c = 0.0011$$

$$\delta c := \frac{\Delta c}{c}$$

$$\delta c = 0.00000808$$

2) Вычитание

$$c1 := a - b$$

$$c1 = -134.6251$$

$$\Delta c1 := \Delta a + \Delta b$$

$$\Delta c1 = 0.0011$$

$$\delta c1 := \frac{\Delta c1}{|c1|}$$

$$\delta c1 = 0.00000817$$

3) Умножение

$$c2 := a \cdot b$$

$$c2 = 97.706999$$

$$\Delta c2 := |a| \cdot \Delta b + |b| \cdot \Delta a$$

$$\Delta c2 = 0.0143$$

$$\delta c2 := \frac{\Delta c2}{c2}$$

$$\delta c2 = 0.000146$$

4) Деление

$$c3 := \frac{a}{b} \quad c3 = 0.0053337$$

$$\Delta c3 := \frac{|a| \cdot \Delta b + |b| \cdot \Delta a}{b^2} \quad \Delta c3 = 0.00000078$$

$$\delta c3 := \frac{\Delta c3}{c3} \quad \delta c3 = 0.00014591$$

5) Возведение в степень

$$c4 := a^3 \quad c4 = 0.376$$

$$\Delta c4 := 3 \cdot a^2 \cdot \Delta a \quad \Delta c4 = 0.000156$$

$$\delta c4 := \frac{\Delta c4}{c4} \quad \delta c4 = 0.000416$$

6) Извлечение корня

$$c5 := \sqrt{a} \quad c5 = 0.8496$$

$$\Delta c5 := \frac{1}{2 \cdot \sqrt{a}} \cdot \Delta a \quad \Delta c5 = 0.000059$$

$$\delta c5 := \frac{\Delta c5}{c5} \quad \delta c5 = 0.000069$$

9. ПРИБЛИЖЕННЫЕ ВЫЧИСЛЕНИЯ ПО ФОРМУЛЕ

9.1. ВЫЧИСЛЕНИЯ ПО ПРАВИЛАМ ПОДСЧЕТА ЦИФР

При вычислении этим методом явного учета погрешностей не ведется, правила подсчета цифр показывают лишь, какое количество значащих цифр или десятичных знаков в результате можно считать надежными.

Сами эти правила основываются на выводах, вытекающих из формул для оценки погрешностей арифметических действий и функций. Приведем эти правила в систематизированном виде.

1. При сложении и вычитании приближенных чисел младший из сохраняемых десятичных разрядов результата должен являться наибольшим среди десятичных разрядов, выражаемых последними верными значащими цифрами исходных данных.

Следует избегать вычитания близких по величине чисел, а так же при пооперационном применении правила для сложения и вычитания нескольких чисел подряд стараться производить действия над числами в порядке возрастания их абсолютных величин.

2. При умножении и делении приближенных чисел нужно выбрать число с наименьшим количеством значащих цифр и округлить остальные числа так, чтобы в них было лишь на одну значащую цифру больше, чем в наименее точно числе.

В результате следует считать верными столько значащих цифр, сколько их в числе с наименьшим количеством значащих цифр.

3. При определении количества верных цифр в значениях элементарных функций от приближенных значений аргумента следует грубо оценить значение модуля производной функции. Если это значение не превосходит единицы или близко к ней, то в значении функции можно считать верными столько знаков после запятой, сколько их имеет значение аргумента. Если же модуль производной функции в окрестности приближенного значения аргумента превосходит единицу, то количество верных десятичных знаков в значении функции меньше, чем в значении аргумента на величину k , где k – наименьший показатель степени, при котором имеет место $|f'(x)| < 10^k$.

4. При записи промежуточных результатов следует сохранять на одну цифру больше, чем рекомендуют правила 1-3. В окончательном результате это запасная цифра округляется.

Правила подсчета цифр носят оценочный характер и не являются методом строгого учета точности вычислений. Обычно их применяют тогда, когда быстро и без особых затрат нужно получить результат, не особо беспокоясь о его достоверности. Между тем практическая надежность этих правил достаточно высока в результате вычислительной вероятности взаимопогашения ошибок, не учитываемой при строгом подсчете предельных погрешностей.

При операционном учете ошибок вычислений используется обычная расчетная таблица – так называемая *расписка формулы*.

Для заданной формулы и исходных данных выполним в MATHCAD(e) расчет по правилам подсчета цифр.

ВЫЧИСЛЕНИЯ ПО ПРАВИЛАМ ПОДСЧЕТА ЦИФР ПО ФОРМУЛЕ

$$z = \frac{\ln(b) - a}{a^2 + 12c}$$

$$a := 0.721$$

$$b := 135.34$$

$$c := 0.01$$

$$\Delta a := 0.0001$$

$$\Delta b := 0.001$$

$$\Delta c := 0.001$$

Вычисления проводим в следующей последовательности:

1) вычисляем $\ln(b)$ $\frac{d}{db} \ln(b) \rightarrow 7.388416440704263855 \cdot 10^{-3}$

$$7.388416440704263855 \cdot 10^{-3} < 1$$

В значении функции $\ln(b)$ берем 3 знака после запятой, как в числе b

$$\ln(b) = 4.908$$

2) вычисляем $d = \ln(b) - a$

$$d := \ln(b) - a$$

$$d = 4.186$$

В значении числа d берем 3 знака после запятой, как и в функции $\ln(b)$, поскольку при сложении и вычитании приближенных чисел младший из сохраняемых десятичных разрядов результата должен являться наибольшим среди десятичных разрядов, выражаемых последними верными значащими цифрами исходных данных.

3) вычисляем $e = a^2$

$$e := a^2$$

$$\frac{d}{da} a^2 \rightarrow 1.443$$

$$1.4438 > 1$$

В значении функции $e = a^2$ берем 3 знака после запятой, то есть на одну десятичную цифру меньше, чем в числе a

$$e = 0.521$$

4) вычисляем $f = 12c$

$$f := 12c$$

$$f = 0.156$$

В значении числа f берем 3 знака после запятой, как и в в числе c

5) вычисляем $g = e + f = a^2 + 12c$

$$g := e + f$$

$$g = 0.677$$

В значении числа g берем 3 знака после запятой, как и в в числах e и f

6) вычисляем $z = \frac{d}{g} = \frac{\ln(b) - a}{a^2 + 12 \cdot c}$

$$z := \frac{d}{g} \quad z = 6.182$$

В значении числа z берем 3 знака после запятой, как и в числах d и g

Окончательно $z = 6.182$

9.2. ВЫЧИСЛЕНИЯ СО СТРОГИМ УЧЕТОМ ПРЕДЕЛЬНЫХ АБСОЛЮТНЫХ ПОГРЕШНОСТЕЙ

9.2.1. РЕШЕНИЕ В СИСТЕМЕ ПРОГРАММИРОВАНИЯ BORLAND PASCAL

```
uses crt;
var a,da,b,db,c,dc,z,z0,dz,z1,d,dd,d1,e,e1,de,f,f1,df,g,g1,dg,h,h1,dh:real;
begin
clrscr;
a:=0.7219; da:=0.0001;
b:=135.347; db:=0.001;
c:=0.013; dc:=0.001;
writeln('Вычисление по формуле Z=(ln(b)-a)/(a*a+12*c) методом ');
writeln('строгого учета предельных абсолютных погрешностей');
writeln('a=',a:8:4,' da=',da:8:4);
writeln('b=',b:8:4,' db=',db:8:4);
writeln('c=',c:8:4,' dc=',dc:8:4);
d:=ln(b); dd:=db/b;
writeln('d=',d:1:6,' dd=',dd:1:7);
write('Введите округленное значение d d1=');read(d1);
e:=d1-a;de:=dd+da;
writeln('e=',e:1:6,' de=',de:1:7);
write('Введите округленное значение e e1=');read(e1);
f:=a*a;df:=2*a*da;
writeln('f=',f:1:6,' df=',df:1:7);
write('Введите округленное значение f f1=');read(f1);
g:=12*c;dg:=12*dc;
writeln('g=',g:1:6,' dg=',dg:1:7);
write('Введите округленное значение g g1=');read(g1);
h:=f1+g1;dh:=df+dg;
writeln('h=',h:1:6,' dh=',dh:1:7);
write('Введите округленное значение h h1=');read(h1);
z:=e1/h1;dz:=(e1*dh+h1*de)/sqr(h1);
writeln('z=',z:1:6,' dz=',dz:1:7);
```

```

write('Введите округленное значение z z1=');read(z1);
z0:=(ln(b)-a)/(a*a+12*c);
writeln('Точное значение z=',z0:1:6);
end.

```

Вычисление по формуле $Z=(\ln(b)-a)/(a^2+12c)$ методом строгого учета предельных абсолютных погрешностей

```

a= 0.7219   da= 0.0001
b=135.3470  db= 0.0010
c= 0.0130   dc= 0.0010
d=4.907842  dd=0.0000074
Введите округленное значение d d1=4.90784
e=4.185940  de=0.0001074
Введите округленное значение e e1=4.1859
f=0.521140  df=0.0001444
Введите округленное значение f f1=0.5211
g=0.156000  dg=0.0120000
Введите округленное значение g g1=0.15
h=0.671100  dh=0.0121444

```

```

Введите округленное значение h h1=0.67
z=6.247612  dz=0.1134041
Введите округленное значение z z1=6.2
Точное значение z=6.181800

```

9.2.2. РЕШЕНИЕ В MATHCAD(e).

Вычисление со строгим учетом предельных абсолютных погрешностей

по формуле $z = \frac{\ln(b) - a}{a^2 + 12 \cdot c}$

$a := 0.7219$

$b := 135.34$

$c := 0.013$

$\Delta a := 0.0001$

$\Delta b := 0.001$

$\Delta c := 0.001$

$d := \ln(b)$

$d = 4.907842$

$\Delta d := \frac{\Delta b}{b}$

$\Delta d = 0.0000074$

Округляем число $d = 4.907842$ по абсолютной погрешности $\Delta d = 0.0000074$ в широком смысле $d1=4.90784$

$d1 := 4.9078$

$$e := d1 - a \quad e = 4.185940 \quad \Delta e := \Delta d + \Delta a$$

$$\Delta e = 0.0001074$$

Округляем число $e = 4.185940$ по абсолютной погрешности $\Delta e = 0.0001074$ в широком смысле

$$e1 = 4.1859$$

$$e1 := 4.1859$$

$$f := a^2 \quad f = 0.521140 \quad \Delta f := 2 \cdot a \cdot \Delta a \quad \Delta f = 0.0001444$$

Округляем число $f = 0.521140$ по абсолютной погрешности $\Delta f = 0.0001444$ в широком смысле $f1 = 0.5211$

$$f1 := 0.5211$$

$$g := 12 \cdot c \quad g = 0.156000 \quad \Delta g := 12 \cdot \Delta c \quad \Delta g = 0.012$$

Округляем число $g = 0.156000$ по абсолютной погрешности $\Delta g = 0.0120000$ в широком смысле $g1 = 0.15$

$$g1 := 0.15$$

$$h := f1 + g1 \quad h = 0.6711 \quad \Delta h := \Delta f + \Delta g \quad \Delta h = 0.0121444$$

Округляем число $h = 0.671100$ по абсолютной погрешности $\Delta h = 0.0121444$ в широком смысле $h1 = 0.67$

$$h1 := 0.67$$

$$z := \frac{e1}{h1} \quad z = 6.248 \quad \Delta z := \frac{e1 \cdot \Delta h + h1 \cdot \Delta e}{h1^2} \quad \Delta z = 0.113404$$

Округляем число $z = 6.247612$ по абсолютной погрешности $\Delta z = 0.1134041$ в широком смысле $z1 = 6.2$

$$z1 := 6.2$$

$$z := \frac{\ln(b) - a}{a^2 + 12 \cdot c} \quad z = 6.181800$$

9.3. ВЫЧИСЛЕНИЯ ПО МЕТОДУ ГРАНИЦ

9.3.1. РЕШЕНИЕ В СИСТЕМЕ ПРОГРАММИРОВАНИЯ BORLAND PASCAL

```
uses crt;
var a,da,a1,a2,b,db,b1,b2,c,dc,c1,c2,z,z1,z2:real;
begin
clrscr;
a:=0.7219; da:=0.0001;
b:=135.347; db:=0.001;
c:=0.013; dc:=0.001;
writeln('Вычисление по формуле  $Z=(\ln(b)-a)/(a*a+12*c)$  методом границ');
writeln('a=',a:8:4,' da=',da:8:4);
writeln('b=',b:8:4,' db=',db:8:4);
writeln('c=',c:8:4,' dc=',dc:8:4);
a1:=a-da; a2:=a+da;
b1:=b-db; b2:=b+db;
c1:=c-dc; c2:=c+dc;
writeln('нижняя граница a=',a1:1:5);
writeln('верхняя граница a=',a2:1:5);
writeln('нижняя граница b=',b1:1:5);
writeln('верхняя граница b=',b2:1:5);
writeln('нижняя граница c=',c1:1:5);
writeln('верхняя граница c=',c2:1:5);
z:=(ln(b)-a)/(a*a+12*c);
writeln('chislo z= ',z:1:5);
z1:=(ln(b1)-a2)/(a2*a2+12*c2);
writeln('нижняя граница z1= ',z1:1:5);
z2:=(ln(b2)-a1)/(a1*a1+12*c1);
writeln('верхняя граница z2= ',z2:1:5);
end.
```

Вычисление по формуле $Z=(\ln(b)-a)/(a*a+12*c)$ методом границ

```
a= 0.7219 da= 0.0001
b=135.3470 db= 0.0010
c= 0.0130 dc= 0.0010
нижняя граница a=0.72180
верхняя граница a=0.72200
нижняя граница b=135.34600
верхняя граница b=135.34800
нижняя граница c=0.01200
верхняя граница c=0.01400
chislo z= 6.18180
нижняя граница z1= 6.07273
верхняя граница z2= 6.29486
```

9.3.2. РЕШЕНИЕ В MATHCAD(e)

По методу границ

ВЫЧИСЛЕНИЕ МЕТОДОМ ГРАНИЦ ПО ФОРМУЛЕ $z = \frac{\ln(b) - a}{a^2 + 12 \cdot c}$,

ГДЕ ЧИСЛА $a=0.7219$, $b=135.347$, $c=0.013$ ЗАДАНЫ ВСЕМИ СВОИМИ ВЕРНЫМИ ЗНАЧАЩИМИ ЦИФРАМИ В ШИРОКОМ СМЫСЛЕ

ИССЛЕДУЕМ ФУНКЦИЮ $z = \frac{\ln(b) - a}{a^2 + 12 \cdot c}$ НА ВОЗРАСТАНИЕ И УБЫВАНИЕ ПО

ПАРАМЕТРАМ a, b, c , ДЛЯ ЧЕГО ВЫЧИСЛЯЕМ ЕЁ ЧАСТНЫЕ ПРОИЗВОДНЫЕ И ОПРЕДЕЛЯЕМ ИХ ЗНАК ПРИ ЗАДАННЫХ ЗНАЧЕНИЯХ ПАРАМЕТРОВ

$$z = \frac{\ln(b) - a}{a^2 + 12 \cdot c}$$

$$\frac{d}{da} \left(\frac{\ln(b) - a}{a^2 + 12 \cdot c} \right) \rightarrow \frac{-1}{a^2 + 12 \cdot c} - 2 \cdot \frac{\ln(b) - a}{(a^2 + 12 \cdot c)^2} \cdot a$$

$$\frac{d}{da} \left(\frac{\ln(b) - a}{a^2 + 12 \cdot c} \right) = \frac{-1}{a^2 + 12 \cdot c} - 2 \cdot \frac{\ln(b) - a}{(a^2 + 12 \cdot c)^2} \cdot a < 0$$

$$\frac{d}{db} \left(\frac{\ln(b) - a}{a^2 + 12 \cdot c} \right) \rightarrow \frac{1}{b \cdot (a^2 + 12 \cdot c)}$$

$$\frac{d}{db} \left(\frac{\ln(b) - a}{a^2 + 12 \cdot c} \right) = \frac{1}{b \cdot (a^2 + 12 \cdot c)} > 0$$

$$\frac{d}{dc} \left(\frac{\ln(b) - a}{a^2 + 12 \cdot c} \right) \rightarrow -12 \cdot \frac{\ln(b) - a}{(a^2 + 12 \cdot c)^2}$$

$$\frac{d}{dc} \left(\frac{\ln(b) - a}{a^2 + 12 \cdot c} \right) = -12 \cdot \frac{\ln(b) - a}{(a^2 + 12 \cdot c)^2} < 0$$

$$\begin{array}{lll} a := 0.7219 & b := 135.347 & c := 0.013 \\ \Delta a := 0.0001 & \Delta b := 0.001 & \Delta c := 0.001 \end{array}$$

$$\begin{array}{llll} a1 := a - \Delta a & a1 = 0.7218 & b1 := b - \Delta b & b1 = 135.346 \\ a2 := a + \Delta a & a2 = 0.722 & b2 := b + \Delta b & b2 = 135.348 \end{array}$$

$$\begin{array}{ll} c1 := c - \Delta c & c1 = 0.012 \\ c2 := c + \Delta c & c2 = 0.014 \end{array}$$

$$\begin{array}{ll} z := \frac{\ln(b) - a}{a^2 + 12 \cdot c} & z = 6.181800 \\ z1 := \frac{\ln(b1) - a2}{a2^2 + 12 \cdot c2} & z1 = 6.072728 \end{array}$$

$$z2 := \frac{\ln(b2) - a1}{a1^2 + 12 \cdot c1} \quad z2 = 6.294856 \quad 6.072728 < z < 6.294856$$

ВЫЧИСЛЕНИЕ ПО ФОРМУЛЕ МЕТОДОМ ГРАНИЦ С ПОШАГОВОЙ РЕГИСТРАЦИЕЙ ВЫЧИСЛЕНИЙ

$$\begin{array}{lll} a := 0.7218 & b := 135.346 & c := 0.013 \\ \Delta a := 0.0001 & \Delta b := 0.001 & \Delta c := 0.001 \end{array}$$

$$\begin{array}{lll} a1 := a - \Delta a & b1 := b - \Delta b & c1 := c - \Delta c \\ a2 := a + \Delta a & b2 := b + \Delta b & c2 := c + \Delta c \\ a1 = 0.7218 & b1 = 135.346 & c1 = 0.012 \\ a2 = 0.722 & b2 = 135.348 & c2 = 0.014 \end{array}$$

$d1 := \ln(b1)$	$d1 = 4.90783$	
$d2 := \ln(b2)$	$d2 = 4.90785$	
$e1 := d1 - a2$	$e1 = 4.18583$	
$e2 := d2 - a1$	$e2 = 4.18605$	
$f1 := (a1)^2 + 12 \cdot c1$	$f1 = 0.664995$	
$f2 := (a2)^2 + 12 \cdot c2$	$f2 = 0.68928$	
$g1 := \frac{e1}{f2}$	$g1 = 6.07273$	
$g2 := \frac{e2}{f1}$	$g2 = 6.29486$	
$z := \frac{\ln(b) - a}{a^2 + 12 \cdot c}$	$z = 6.18180$	
$6.07273 < z < 6.29486$		
$z1 = 6.07273$	$z2 = 6.29486$	$z = 6.18180$

ЛИТЕРАТУРА

1. Агабеков Л.Е., Борисов С.В., Ваулин А.С. и др. Программирование в среде Турбо Паскаль. Инструментальные средства персональных ЭВМ. Кн 4. / Под ред. Б.Г. Трусова. – М.: Высшая школа, 1993.
2. Азаров А. И., Басик В. А., Мелешко И. Н. и др. Сборник задач по методам вычислений. М.: Наука, ФМ, 1994.
3. Амосов А.А., Дубинская Ю.А., Копченков Н.В. Вычислительные методы для инженеров. М.: Высшая школа, 1994.
4. Бахвалов Н.С., Жидков Н. П., Кобельков Г. М. Численные методы. М.: Лаборатория базовых знаний, 2000
5. Волков Е. А. Численные методы. М.: Наука, ФМ, 1987.
6. Данилина Н.И., Дубровская Н.С., Кваша О.П., Смирнов Г.Л. Вычислительная математика. М.: Высшая школа, 1985.
7. Демидович Б. П., Марон И. А. Основы вычислительной математики.- М.: Наука, 1967.
8. Дьяконов В.П., Новиков Ю., Рычков В. Компьютер для студента: Самоучитель. СПб.: ПИТЕР, 2000.
9. Карпов М., Мирошниченко Н. Microsoft Visio 2000: Краткий курс. – СПб: Питер, 2001.
10. Кирьянов Д.В. Самоучитель Mathcad 11. - СПб.: БХВ - Петербург, 2003.
11. Лапчик М. П., Рагулина М. И., Хеннер Е. К. Численные методы.- М.: Академия, 2005.
12. Рашевская М.А. CorelDRAW. Практическое руководство. – М.: ДИАЛОГ-МИФИ, 2003.
13. Фаронов В.В. Турбо Паскаль 7.0: Учебное пособие. Начальный курс. – М.: Нолидж, 1997.

Гурьев Евгений Константинович, Зотов Владимир Александрович

ПРИБЛИЖЕННЫЕ ВЫЧИСЛЕНИЯ

Методические указания к лабораторной и курсовой работам
по курсам «Вычислительная математика» и «Информатика. Численные
методы»

Редактор *М.А. Соколова*

Технический редактор *А.Б. Седов*

Компьютерная верстка выполнена *А.И. Балаш*

Подписано в печать 13.04.05. Усл. печ. л. 10,23. Уч. изд. л. 8,02.

Формат 60 x 84 $\frac{1}{16}$. Печать офсетная.

Тираж 410 экз. Заказ № 123

Издательский центр МАТИ

109240, Москва, Берниковская наб., 14

Типография Издательского центра МАТИ

109240, Москва, Берниковская наб., 14