ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ В КОМПЬЮТЕРЕ

В настоящее время во всех вычислительных машинах информация представляется с помощью электрических сигналов. При этом возможны две формы ее представления – в виде непрерывного сигнала (с помощью сходной величины – аналога) и в виде нескольких сигналов (с помощью набора напряжений, каждое из которых соответствует одной из цифр представляемой величины).

Первая форма представления информации называется аналоговой, или непрерывной. Величины, представленные в такой форме, могут принимать принципиально любые значения в определенном диапазоне. Количество значений, которые может принимать такая величина, бесконечно велико. Отсюда названия – непрерывная величина и непрерывная информация. Слово непрерывность отчетливо выделяет основное свойство таких величин – отсутствие разрывов, промежутков между значениями, которые может принимать данная аналоговая величина. При использовании аналоговой формы для создания вычислительной машины потребуется меньшее число устройств (каждая величина представляется одним, а не несколькими сигналами), но эти устройства будут сложнее (они должны различать значительно большее число состояний сигнала). Непрерывная форма представления используется в аналоговых вычислительных машинах (АВМ). Эти машины предназначены в основном для решения задач, описываемых системами дифференциальных уравнений: исследования поведения подвижных объектов, моделирования процессов и систем, решения задач параметрической оптимизации и оптимального управления. Устройства для обработки непрерывных сигналов обладают более высоким быстродействием, они могут интегрировать сигнал, выполнять любое его функциональное преобразование и т. п. Однако из-за сложности технической реализации устройств выполнения логических операций с непрерывными сигналами, длительного хранения таких сигналов, их точного измерения АВМ не могут эффективно решать задачи, связанные с хранением и обработкой больших объемов информации.

Вторая форма представления информации называется дискретной (цифровой). Такие величины, принимающие не все возможные, а лишь вполне определенные значения, называются дискретными (прерывистыми). В отличие от непрерывной величины, количество значений дискретной величины всегда будет конечным. Дискретная форма представления используется в цифровых электронно-вычислительных машинах (ЭВМ), которые легко решают задачи, связанные с хранением, обработкой и передачей больших объемов информации.

Для автоматизации работы ЭВМ с информацией, относящейся к различным типам, очень важно унифицировать их форму представления – для этого обычно используется прием кодирования.

Кодирование – это представление сигнала в определенной форме, удобной или пригодной для последующего использования сигнала. Говоря строже, это правило, описывающее отображение одного набора знаков в другой набор знаков. Тогда отображаемый набор знаков называется исходным алфавитом, а набор знаков, который используется для отображения, – кодовым алфавитом, или алфавитом для кодирования. При этом кодированию подлежат как отдельные символы исходного алфавита, так и их комбинации. Аналогично для построения кода используются как отдельные символы кодового алфавита, так и их комбинации.

Совокупность символов кодового алфавита, применяемых для кодирования одного символа (или одной комбинации символов) исходного алфавита, называется кодовой комбинацией, или, короче, кодом символа. При этом кодовая комбинация может содержать один символ кодового алфавита.

Символ (или комбинация символов) исходного алфавита, которому соответствует кодовая комбинация, называется исходным символом.

Совокупность кодовых комбинаций называется кодом.

Взаимосвязь символов (или комбинаций символов, если кодируются не отдельные символы исходного алфавита) исходного алфавита с их кодовыми комбинациями составляет таблицу соответствия (или таблицу кодов).

В качестве примера можно привести систему записи математических выражений, азбуку Морзе, морскую флажковую азбуку, систему Брайля для слепых и др.

В вычислительной технике также существует своя система кодирования – она называется двоичным кодированием и основана на представлении данных последовательностью всего двух знаков: 0 и 1 (используется двоичная система счисления). Эти знаки называются двоичными цифрами, или битами (binary digital).

Если увеличивать на единицу количество разрядов в системе двоичного кодирования, то увеличивается в два раза количество значений, которое может быть выражено в данной системе. Для расчета количества значений используется следующая формула:

$N=2m$,

где N – количество независимо кодируемых значений,

а m – разрядность двоичного кодирования, принятая в данной системе.

Например, какое количество значений (N) можно закодировать 10-ю разрядами (m)?

Для этого возводим 2 в 10 степень (m) и получаем N=1024, т. е. в двоичной системе кодирования 10-ю разрядами можно закодировать 1024 независимо кодируемых значения.

Кодирование текстовой информации

Для кодирования текстовых данных используются специально разработанные таблицы кодировки, основанные на сопоставлении каждого символа алфавита с определенным целым числом. Восьми двоичных разрядов достаточно для кодирования 256 различных символов. Этого хватит, чтобы выразить различными комбинациями восьми битов все символы английского и русского языков, как строчные, так и прописные, а также знаки препинания, символы основных арифметических действий и некоторые общепринятые специальные символы. Но не все так просто, и существуют определенные сложности. В первые годы развития вычислительной техники они были связаны с отсутствием необходимых стандартов, а в настоящее время, наоборот, вызваны изобилием одновременно действующих и противоречивых стандартов. Практически для всех распространенных на земном шаре языков созданы свои кодовые таблицы. Для того чтобы весь мир одинаково кодировал текстовые данные, нужны единые таблицы кодирования, что до сих пор пока еще не стало возможным.

Кодирование графической информации

Кодирование графической информации основано на том, что изображение состоит из мельчайших точек, образующих характерный узор, называемый растром. Каждая точка имеет свои линейные координаты и свойства (яркость), следовательно, их можно выразить с помощью целых чисел – растровое кодирование позволяет использовать двоичный код для представления графической информации. Черно-белые иллюстрации представляются в компьютере в виде комбинаций точек с 256 градациями серого цвета – для кодирования яркости любой точки достаточно восьмиразрядного двоичного числа.

Для кодирования цветных графических изображений применяется принцип декомпозиции (разложения) произвольного цвета на основные составляющие. При этом могут использоваться различные методы кодирования цветной графической информации. Например, на практике считается, что любой цвет, видимый человеческим глазом, можно получить путем механического смешивания основных цветов. В качестве таких составляющих используют три основных цвета: красный (Red, R), зеленый (Green, G) и синий (Blue, B). Такая система кодирования называется системой RGB.

На кодирование цвета одной точки цветного изображения надо затратить 24 разряда. При этом система кодирования обеспечивает однозначное определение 16,5 млн различных цветов, что на самом деле близко к чувствительности человеческого глаза.

Режим представления цветной графики с использованием 24 двоичных разрядов называется полноцветным (True Color).

Каждому из основных цветов можно поставить в соответствие дополнительный цвет, то есть цвет, дополняющий основной цвет до белого. Соответственно дополнительными цветами являются: голубой (Cyan, C), пурпурный (Magenta, M) и желтый (Yellow, Y). Такой метод кодирования принят в полиграфии, но в полиграфии используется еще и четвертая краска – черная (Black, K). Данная система кодирования обозначается CMYK, и для представления цветной графики в этой системе надо иметь 32 двоичных разряда. Такой режим называется полноцветным (True Color).

Если уменьшать количество двоичных разрядов, используемых для кодирования цвета каждой точки, то можно сократить объем данных, но при этом диапазон кодируемых цветов заметно сокращается. Кодирование цветной графики 16-разрядными двоичными числами называется режимом High Color.

Кодирование звуковой информации

Приемы и методы кодирования звуковой информации пришли в вычислительную технику наиболее поздно и до сих пор далеки от стандартизации. Множество отдельных компаний разработали свои корпоративные стандарты, хотя можно выделить два основных направления. Метод FM (Frequency Modulation) основан на том, что теоретически любой сложный звук можно разложить на последовательность простейших гармоничных сигналов разной частоты, каждый из которых представляет правильную синусоиду, а следовательно, может быть описан числовыми параметрами, то есть кодом. В природе звуковые сигналы имеют непрерывный спектр, то есть являются аналоговыми. Их разложение в гармонические ряды и представление в виде дискретных цифровых сигналов выполняют специальные устройства – аналогово-цифровые преобразователи (АЦП). Обратное преобразование для воспроизведения звука, закодированного числовым кодом, выполняют цифро-аналоговые преобразователи (ЦАП). При таких преобразованиях часть информации теряется, поэтому качество звукозаписи обычно получается не вполне удовлетворительным и соответствует качеству звучания простейших электромузыкальных инструментов с «окрасом», характерным для электронной музыки.

Метод таблично-волнового синтеза (Wave-Table) лучше соответствует современному уровню развития техники. Имеются заранее подготовленные таблицы, в которых хранятся образцы звуков для множества различных музыкальных инструментов. В технике такие образцы называются сэмплами. Числовые коды выражают тип инструмента, номер его модели, высоту тона, продолжительность и интенсивность звука, динамику его изменения. Поскольку в качестве образцов используются «реальные» звуки, то качество звука, полученного в результате синтеза, получается очень высоким и приближается к качеству звучания реальных музыкальных инструментов.

ЕДИНИЦЫ ИЗМЕРЕНИЯ ОБЪЕМА ИНФОРМАЦИИ.

Для измерения длины есть такие единицы, как миллиметр, сантиметр, метр, километр. Известно, что масса измеряется в граммах, килограммах, центнерах и тоннах. Бег времени выражается в секундах, минутах, часах, днях, месяцах, годах, веках. Компьютер работает с информацией и для измерения ее объема также имеются соответствующие единицы измерения.

Мы уже знаем, что компьютер воспринимает всю информацию через нули и единички. Бит – это минимальная единица измерения информации, соответствующая одной двоичной цифре («0» или «1»).

Байт состоит из восьми бит. Используя один байт, можно закодировать один символ из 256 возможных (256 = 28). Таким образом, один байт равен одному символу, то есть 8 битам:

1 символ = 8 битам = 1 байту.

Изучение компьютерной грамотности предполагает рассмотрение и других, более крупных единиц измерения информации.

Таблица байтов:

1 байт = 8 бит

1 Кб (1 Килобайт) = 210 байт = 2*2*2*2*2*2*2*2*2*2 байт =

= 1024 байт (примерно 1 тысяча байт – 103 байт)

1 Мб (1 Мегабайт) = 220 байт = 1024 килобайт (примерно 1 миллион байт – 106 байт)

1 Гб (1 Гигабайт) = 230 байт = 1024 мегабайт (примерно 1 миллиард байт – 109 байт)

1 Тб (1 Терабайт) = 240 байт = 1024 гигабайт (примерно 1012 байт). Терабайт иногда называют тонна.

1 Пб (1 Петабайт) = 250 байт = 1024 терабайт (примерно 1015 байт).

1 Эксабайт = 260 байт = 1024 петабайт (примерно 1018 байт).

1 Зеттабайт = 270 байт = 1024 эксабайт (примерно 1021 байт).

1 Йоттабайт = 280 байт = 1024 зеттабайт (примерно 1024 байт).

В приведенной выше таблице степени двойки (210, 220, 230 и т.д.) являются точными значениями килобайт, мегабайт, гигабайт. А вот степени числа 10 (точнее, 103, 106, 109 и т.п.) будут уже приблизительными значениями, округленными в сторону уменьшения. Таким образом, 210 = 1024 байта представляет точное значение килобайта, а 103 = 1000 байт является приблизительным значением килобайта. Такое приближение (или округление) вполне допустимо и является общепринятым.

Ниже приводится таблица байтов с английскими сокращениями (в левой колонке):

1 Kb ~ 103 b = 10*10*10 b= 1000 b – килобайт

1 Mb ~ 106 b = 10*10*10*10*10*10 b = 1 000 000 b – мегабайт

1 Gb ~ 109 b – гигабайт

1 Tb ~ 1012 b – терабайт

1 Pb ~ 1015 b – петабайт

1 Eb ~ 1018 b – эксабайт

1 Zb ~ 1021 b – зеттабайт

1 Yb ~ 1024 b – йоттабайт

Выше в правой колонке приведены так называемые «десятичные приставки», которые используются не только с байтами, но и в других областях человеческой деятельности. Например, приставка «кило» в слове «килобайт» означает тысячу байт, также как в случае с километром она соответствует тысяче метров, а в примере с килограммом она равна тысяче грамм.

Возникает вопрос: есть ли продолжение у таблицы байтов? В математике есть понятие бесконечности, которое обозначается как перевернутая восьмерка: ∞. Понятно, что в таблице байтов можно и дальше добавлять нули, а точнее, степени к числу 10 таким

образом: 1027, 1030, 1033 и так до бесконечности. Но зачем это надо? В принципе, пока хватает терабайт и петабайт. В будущем, возможно, уже мало будет и йоттабайта.

Напоследок парочка примеров по устройствам, на которые можно записать терабайты и гигабайты информации. Есть удобный «терабайтник» – внешний жесткий диск, который подключается через порт USB к компьютеру. На него можно записать терабайт информации. Особенно удобно для ноутбуков (где смена жесткого диска бывает проблематична) и для резервного копир.

ДВОИЧНАЯ СИСТЕМА ИСЧИСЛЕНИЯ

В двоичной системе счисления используются всего две цифры 0 и 1. Другими словами, двойка является основанием двоичной системы счисления. (Аналогично у десятичной системы основание 10.)

Чтобы научиться понимать числа в двоичной системе счисления, сначала рассмотрим, как формируются числа в привычной для нас десятичной системе счисления.

В десятичной системе счисления мы располагаем десятью знаками-цифрами (от 0 до 9). Когда счет достигает 9, то вводится новый разряд (десятки), а единицы обнуляются и счет начинается снова. После 19 разряд десятков увеличивается на 1, а единицы снова обнуляются. И так далее. Когда десятки доходят до 9, то потом появляется третий разряд – сотни.

Двоичная система счисления аналогична десятичной за исключением того, что в формировании числа участвуют всего лишь две знака-цифры: 0 и 1. Как только разряд достигает своего предела (т.е. единицы), появляется новый разряд, а старый обнуляется.

Попробуем считать в двоичной системе:0 – это ноль1 – это один (и это предел разряда)10 – это два11 – это три (и это снова предел)100 – это четыре101 – пять110 – шесть111 – семь и т.д.

Перевод чисел из двоичной системы счисления в десятичную

Не трудно заметить, что в двоичной системе счисления длины чисел с увеличением значения растут быстрыми темпами. Как определить, что значит вот это: 10001001? Непривычный к такой форме записи чисел человеческий мозг обычно не может понять сколько это. Неплохо бы уметь переводить двоичные числа в десятичные.

В десятичной системе счисления любое число можно представить в форме суммы единиц, десяток, сотен и т.д. Например:

$1476 = 1000 + 400 + 70 + 6$

Можно пойти еще дальше и разложить так:

$1476 = 1 * 10^3 + 4 * 10^2 + 7 * 10^1 + 6 * 10^0$

Посмотрите на эту запись внимательно. Здесь цифры 1, 4, 7 и 6 - это набор цифр из которых состоит число 1476. Все эти цифры поочередно умножаются на десять возведенную в ту или иную степень. Десять – это основание десятичной системы счисления. Степень, в которую возводится десятка – это разряд цифры за минусом единицы.

Аналогично можно разложить и любое двоичное число. Только основание здесь будет 2:

$10001001 = 1*2^7 + 0*2^6 + 0*2^5 + 0*2^4 + 1*2^3 + 0*2^2 + 0*2^1 + 1*2^0$

Если посчитать сумму составляющих, то в итоге мы получим десятичное число, соответствующее 10001001:

$1*2^7 + 0*2^6 + 0*2^5 + 0*2^4 + 1*2^3 + 0*2^2 + 0*2^1 + 1*2^0 = 128 + 0 + 0 + 0 + 8 + 0 + 0 + 1 = 137$

Т.е. число 10001001 по основанию 2 равно числу 137 по основанию 10. Записать это можно так:

$10001001_2 = 137_{10}$

Почему двоичная система счисления так распространена?

Дело в том, что двоичная система счисления – это язык вычислительной техники. Каждая цифра должна быть как-то представлена на физическом носителе. Если это десятичная система, то придется создать такое устройство, которое может быть в десяти состояниях. Это сложно. Проще изготовить физический элемент, который может быть лишь в двух состояниях (например, есть ток или нет тока). Это одна из основных причин, почему двоичной системе счисления уделяется столько внимания.

Перевод десятичного числа в двоичное

Может потребоваться перевести десятичное число в двоичное. Один из способов – это деление на два и формирование двоичного числа из остатков. Например, нужно получить из числа 77 его двоичную запись:

77 / 2 = 38 (1 остаток)

38 / 2 = 19 (0 остаток)

19 / 2 = 9 (1 остаток)

9 / 2 = 4 (1 остаток)

4 / 2 = 2 (0 остаток)

2 / 2 = 1 (0 остаток)

1 / 2 = 0 (1 остаток)

Собираем остатки вместе, начиная с конца: 1001101. Это и есть число 77 в двоичном представлении. Проверим:

$1001101 = 1*2^6 + 0*2^5 + 0*2^4 + 1*2^3 + 1*2^2 + 0*2^1 + 1*2^0 = 64 + 0 + 0 + 8 + 4 + 0 + 1 = 77$

ПЕРЕВОД ИЗ ЛЮБОЙ СИСТЕМЫ СЧИСЛЕНИЯ В ДЕСЯТИЧНУЮ

Для того, что бы перевести десятичную дробь из любой системы счисления в десятичную, надо воспользоваться развернутой формой записи числа

Примеры:

$23{,}12_8 = 2*8^1 + 3*8^0 + 1*8^{-1} + 2*8^{-2} = 9{,}25_{10}$

$23{,}12_{16} = 2*16^1 + 3*16^0 + 1*16^{-1} + 2*16^{-2} = 35{,}125_{10}$

Перевод дробной части числа из двоичной системы счисления в восьмеричную

При переводе из 2-ой в 8-ую систему счисления надо дробную часть разбить на триады (по три разряда) и записать каждую триаду эквивалентным двоичным кодом, недостающее число разрядов надо дополнить справа нулями.

Сравните с алгоритмом перевода целой части числа и объясните разницу.

Пример: $111100101{,}0111_2 = 111\ 100\ 101{,}011\ 100_2 = 745{,}34_8$.

Перевод дробной части числа из двоичной системы счисления в шестнадцатеричную

При переводе из 2-ой в 16-ую систему счисления надо дробную часть разбить на тетрады (по 4 разряда) и записать каждую тетраду эквивалентным двоичным кодом, недостающее число разрядов надо дополнить справа нулями.

Пример: $11101001000{,}1101001_2 = 0111\ 0100\ 1000{,}1101\ 0010_2 = 748{,}D2_{16}$.

Для того, что бы перевести дробь из десятичной системы счисления в любую другую, надо:

1) последовательно умножать данную дробь на основание системы счисления, в которую переводим, до тех пор, пока дробная часть не будет равна нулю, или не будет достигнута требуемая точность вычислений. При этом необходимо выделять целые части получаемых произведений;

2) полученные целые части произведений, являются цифрами числа в новой системе счисления (при необходимости их надо привести в соответствие с алфавитом этой системы счисления);

3) составить дробную часть в новой системе счисления начиная с целой части первого произведения.

Пример 1. Перевести число $0{,}65625_{10}$ в восьмеричную систему счисления.	*Пример 2.* Перевести число $0{,}65625_{10}$ в шестнадцатеричную систему счисления.
0, 65625 × 8 5 25000 × 8 2 00000 $0{,}65625_{10} = 0{,}52_8$	0, 65625 × 16 10 50000 × 16 (A) 8 00000 $0{,}65625_{10} = 0{,}A8_{16}$

При переводе смешанных чисел целые и дробные части переводятся отдельно .

Задания

1. Переведите числа из десятичной системы счисления в двоичную:

1) 40,5 2) 31,75 3) 124,25

Ответы:1) 101000,1 2) 11111,11 3) 1111100,01

2. Переведите десятичные дроби в двоичную систему счисления (ответ записать с шестью двоичными знаками):

0,4622)	0,5198)	0,5803)	0,6) 124
0,7351)	0,7982)	0,8544)	0,9) 321

Ответы:

0,011101) 0,1000) 3) 0,100101) 01

0,100111) 0,1011) 6) 0,110011) 11

0,110110) 0,1110) 11

ГЕОМЕТРИЧЕСКИЙ РЯД

Так называется ряд (бесконечная сумма), члены которого образуют геометрическую прогрессию с первым членом a_0 и знаменателем прогрессии, равным q.

Если $|q| < 1$, то существует предел суммы n первых членов этой прогрессии при неограниченном увеличении количества этих членов n:

$$\sum_{k=0}^{\infty} a_0 q^k = a_0 + a_0 q + a_0 q^2 + \ldots + a_0 q^k + \ldots = \lim_{n \to \infty} \left(a_0 + a_0 q + \ldots + a_0 q^{n-1} \right) = \frac{a_0}{1-q}.$$

В этом случае говорят о бесконечно убывающей геометрической прогрессии.

ОБРАЩЕНИЕ ПЕРИОДИЧЕСКОЙ ДЕСЯТИЧНОЙ ДРОБИ В ОБЫКНОВЕННУЮ.

Предположим, мы хотим обратить периодическую десятичную дробь 0.(3) в обыкновенную. Рассмотрим эту десятичную дробь в следующем виде:

$$0.(3) = 0.3333\ldots = \frac{3}{10} + \frac{3}{100} + \frac{3}{1000} + \frac{3}{10000} + \ldots$$

Это бесконечно убывающая геометрическая прогрессия, первый член которой равен 3/10, а разность $q = 1/10$. В соответствии с выше приведенной формулой эта сумма равна:

$$\frac{3/10}{1 - 1/10} = \frac{3/10}{9/10} = \frac{3}{9} = \frac{1}{3}.$$

Таким образом, 0.(3) = 1/3.

ИСТОЧНИКИ И КЛАССИФИКАЦИЯ ПОГРЕШНОСТЕЙ РЕЗУЛЬТАТА ЧИСЛЕННОГО РЕШЕНИЯ ЗАДАЧИ

Приближенным числом или *приближением* называется число, незначительно отличающееся от точного значения величины и заменяющее его в вычислениях. Под погрешностью же принято понимать разность между абсолютным значением и его приближением.

Для правильного понимания подходов и критериев, используемых при решении прикладной задачи с применением ЭВМ, важно понимать, что получить точное значение решения практически невозможно. Получаемое на ЭВМ решение почти всегда (за исключением некоторых весьма специальных случаев) содержит погрешность, т.е. является приближенным. Невозможность получения точного решения следует уже из ограниченной разрядности вычислительной машины.

Наличие погрешности обусловлено рядом весьма глубоких причин.

1. Математическая модель является лишь 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.

Полная погрешность $(\delta y = y - y*)$ результата решения задачи на ЭВМ складывается из трех составляющих: неустранимой погрешности, погрешности метода и вычислительной погрешности: $(\delta y = \delta_{Н} y + \delta_{М} y + \delta_{В} y)$.

Появление неустранимой погрешности обусловлено тем, что принятие математической модели и задание исходных данных вносит в решение ошибку, которая не может быть устранена далее. Единственный способ уменьшить эту погрешность — перейти к более точной математической модели и задать более точные исходные данные.

Достоверная информация о порядке величины погрешности метода позволяет осознанно выбрать метод решения задачи и разумно задать его точность. Желательно, чтобы величина погрешности метода была в 2—10 раз меньше неустранимой погрешности. Большее значение ощутимо снижает точность результата, меньшее — обычно требует увеличения затрат, практически уже не влияя на значение полной погрешности.

Величина вычислительной погрешности (при фиксированных модели, входных данных и методе решения) в основном определяется характеристиками используемой ЭВМ. Желательно, чтобы эта величина была хотя бы на порядок меньше величины погрешности метода и совсем не желательна ситуация, когда она существенно ее превышает.

АБСОЛЮТНАЯ И ОТНОСИТЕЛЬНАЯ ПОГРЕШНОСТИ

Пусть имеется некоторая числовая величина, и числовое значение, которое ей присвоено (a), считается точным, тогда подпогрешностью приближенного значения числовой величины (ошибкой) $(\Delta\ a)$ понимают разность между точным и приближенным значением числовой величины: $a^* - a = \Delta\ a$

Погрешность может принимать как положительное так и отрицательное значение. Величина (a^*) называется известным приближением к точному значению числовой величины - любое число, которое используется вместо точного значения. Простейшей количественной мерой ошибки является абсолютная погрешность.

Абсолютной погрешностью приближенного значения (a^*) называют величину $\Delta\ (a^*)$, про которую известно, что: $|\ a^* - a\ | \leq \Delta\ (a^*)$.

Качество приближения существенным образом зависит от принятых единиц измерения и масштабов величин, поэтому целесообразно соотнести погрешность величины и ее значение, для чего вводится понятие относительной погрешности.

Относительной погрешностью приближенного значения называют величину $\delta(a^*)$, про которую известно, что: $\left|\frac{a^* - a}{a^*}\right| = \frac{\Delta\ (a^*)}{|a|} = \delta(a^*)$.

Относительную погрешность часто выражают в процентах. Использование относительных погрешностей удобно, в частности, тем, что они не зависят от масштабов величин и единиц измерения.

Так как точное значение обычно неизвестно, то непосредственное вычисление величин абсолютной и относительной погрешностей по предложенным формулам невозможно. Более реальная и часто поддающаяся решению задача состоит в получении оценок погрешности вида: $|a - a^*| \leq \bar{\Delta}(a^*)$; $\left|\frac{a - a^*}{a}\right| \leq \bar{\delta}(a^*)$ (*),

где $\bar{\Delta}(a^*)$ и $\bar{\delta}(a^*)$ — известные величины, которые называют верхними границами (или просто границами) абсолютной и относительной погрешностей.

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

Точно так же если величина $\bar{\delta}(a^*)$ известна, то следует положить: $\bar{\Delta}(a^*) = |a|\ \bar{\delta}(a^*)$

Но поскольку точное значение a неизвестно, на практике используют приближенные равенства вида: $\bar{\delta}(a^*) \approx \frac{\bar{\Delta}(a^*)}{|a^*|}$, $\bar{\Delta}(a^*) \approx |a^*|\ \bar{\delta}(a^*)$

В литературе по методам вычислений широко используется термин "точность". Точное значение величины — это значение, не содержащее погрешности. Повышение точности воспринимается как уменьшение 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. Вообще говоря, следовало бы говорить об абсолютной точности и относительной точности, но часто этого не делают, считая, что из контекста ясно, как измеряется величина погрешности.

ПРИБЛИЖЕННЫЕ ВЫЧИСЛЕНИЯ. ЗНАЧАЩИЕ ЦИФРЫ

Выполняя вычисления, всегда необходимо помнить о той точности, которую нужно или которую можно получить. Недопустимо вести вычисления с большой точностью, если данные задачи не допускают или не требуют этого (например, семизначная таблица логарифмов при вычислениях с числами, имеющими 5 верных значащих цифр - избыточна). Твёрдое знакомство с правилами приближенных вычислений необходимо каждому, кому приходится вычислять.

Если абсолютная погрешность величины a не превышает одной единицы разряда последней цифры числа a, то говорят, что у числа все знаки верные.

Приближенные числа следует записывать, сохраняя только верные знаки. Если, например, абсолютная погрешность числа 52400 равна 100, то это число должно быть записано, например, в виде $524 \cdot 10^2$ или $0{,}524 \cdot 10^5$. Оценить погрешность приближенного числа можно, указав, сколько верных значащих цифр оно содержит. При подсчете значащих цифр не считаются нули с левой стороны числа.

Примеры:

1 куб.фут = 0.0283 $м^3$ - три верных значащих цифры

1 дюйм = 2,5400 v пять верных значащих цифр

Если число a имеет n верных значащих цифр, то его относительная погрешность d_a Т $1/(z*d^n-1)$, где z - первая значащая цифра числа a; d - основание системы счисления.

У числа a с относительной погрешностью d_a верны n значащих цифр, где n - наибольшее целое число, удовлетворяющее неравенству $(1+Z)d_a$ Т d^{l-n}.

Пример:

Если число a = 47,542 получено в результате действий над приближенными числами и известно, что d_a = 0,1%, то а имеет 3 верных знака, так как $(4+1)0{,}001$ Т 10^{v2}.

Округление

Если приближенное число содержит лишние (или неверные) знаки, то его следует округлить. При округлении сохраняются только верные знаки; лишние знаки отбрасываются, причем если первая отбрасываемая цифра больше или равна $d/2$, то последняя сохраняемая цифра увеличивается на единицу. При округлении возникает дополнительная погрешность, не превышающая половины единицы разряда последней значащей цифры округленного числа. Поэтому, чтобы после округления все знаки были верны, погрешность до округления должна быть не больше половины единицы того разряда, до которого предполагают делать округление.

ДЕЙСТВИЯ НАД ПРИБЛИЖЕННЫМИ ЧИСЛАМИ

Результат действий над приближёнными числами представляет собой также приближённое число. Погрешность результата может быть выражена через погрешности первоначальных данных при помощи следующих теорем: 1) Предельная абсолютная погрешность алгебраической суммы равна сумме предельных абсолютных погрешностей слагаемых; 2) Относительная погрешность суммы заключена между наибольшей и наименьшей из относительных погрешностей слагаемых; 3) Относительная погрешность произведения или частного равна сумме относительных погрешностей сомножителей или, соответственно, делимого и делителя; 4) Относительная погрешность n-ой степени приближенного числа в n раз больше относительной погрешности основания (как у целых, так и для дробных n). Для приближения малых величин существуют правила. Пусть E – малая величина, тогда

1. $(1+E)^2 \approx 1+2E$;
2. $\frac{1}{1+E} \approx 1-E$;
3. $\frac{1}{1-E} \approx 1+E$;
4. $(1+E)^n \approx 1+nE, n \in N$;
5. $\sqrt[n]{1+E} \approx 1+\frac{1}{n}E$;
6. $10^E \approx 1+2{,}303E$;
7. $\lg(1+E) \approx 0{,}4343E$ /

Пользуясь этими теоремами, можно определить погрешность результата любой комбинации арифметических действий над приближенными числами. Пример:

$$Z = \sqrt{\frac{x}{1+y}};\ \delta_z = \frac{1}{2}(\delta_x + \delta_{1+y}) = \frac{1}{2}\left(\frac{\Delta_x}{x} + \frac{\Delta_y}{1+y}\right)$$

Предельная абсолютная погрешность заведомо превосходит абсолютную величину истинной погрешности, поскольку предельное значение вычисляется в предположения, что различные погрешности усиливают друг друга; практически это бывает редко. При массовых вычислениях, когда не учитывают погрешность каждого отдельного результата, пользуются следующими правилами подсчета цифр. При соблюдении этих правил можно считать, что в среднем полученные результаты будут иметь все знаки верными, хотя в отдельных случаях возможна ошибка в несколько единиц последнего знака.

1. При сложении и вычитании приближённых чисел в результате следует сохранять столько десятичных знаков, сколько их в приближённом данном с наименьшим числом десятичных знаков.
2. При умножении и делении в результате следует сохранять столько значащих цифр, сколько их имеет приближённое данное с наименьшим числом значащих цифр.
3. При возведении в квадрат или куб в результате следует сохранять столько значащих цифр, сколько их имеет возводимое в степень приближённое число (последняя цифра квадрата и особенно куба при этом менее надежна, чем последняя цифра основания).
4. При увеличении квадратного и кубического корней в результате следует брать столько значащих цифр, сколько их имеет приближённое значение подкоренного числа (последняя цифра квадратного и особенно кубического корня при этом более надёжна, чем последняя цифра подкоренного числа).
5. Во всех промежуточных результатах следует сохранять одной цифрой более, чем рекомендуют предыдущие правила. В окончательном результате эта запасная цифра отбрасывается.
6. Если некоторые данные имеют больше десятичных знаков (при сложении и вычитании) или больше значащих цифр (при умножении, делении, возведении в степень, извлечении корня), чем другие, то их предварительно следует округлить, сохраняя лишь одну лишнюю цифру.

Если данные можно брать с произвольной точностью, то для получения результата с K цифрами данные следует брать с таким числом цифр, какое даёт согласно правилам 1-4(K+1) цифру в результате.

ПОГРЕШНОСТИ ВЫЧИСЛЕНИЙ

В реальных расчетах на ЭВМ много хлопот пользователю доставляют погрешности, возникающие по разным причинам, включая следующие.

1). Погрешности данных. При вычислении по формулам значения констант, параметров и переменных не могут быть известны или представлены точно, так что возможны вычисления лишь с их приблизительными значениями. Ошибки, происходящие из-за неточности данных, называются погрешностями данных.

2). Погрешности округления. Машинная арифметика выполняется неточно даже для точно известных аргументов, так что большинство результатов арифметических операций и библиотечных функций будут неточными, т.к. все результаты представляются числами предписанного формата, имеющего фиксированное число "значащих" цифр. Разность между полученным и истинными результатами называется погрешностью округления вычислений.

3). Погрешность усечения. Многие математические объекты, такие как интегралы, производные, алгебраические и трансцендентные функции, определяются в действительности как пределы бесконечных последовательностей операций. В случае дифференцирования простых функций, имеющиеся правила дают значения этих пределов точно, в виде формул. Но так бывает далеко не всегда: вместо бесконечной последовательности вычислений приходится ограничиваться конечным числом шагов. Получающаяся ошибка приближенного результата называется ошибкой усечения.

Обычно результат, полученный исполнением машинной программы, будет искажен погрешностями указанных типов. Кроме того, невозможно провести черту, отделяющую одну ошибку от другой, или исключить их полностью. Например, числа $\frac{1}{3}$ или π, встречающиеся в формулах, невозможно представить точно в системе счисления конкретной ЭВМ. Получающаяся погрешность может рассматриваться как следствие погрешности данных, округления или усечения. Аналогично погрешность вычисления $\sin x$ с помощью библиотечной процедуры может быть следствием ошибки округления или усечения. Каков бы ни был источник, погрешность распространяется в численных расчетах, и точность результата сотен или тысяч неточных вычислений на неточных данных всегда требует определенного изучения. Проблема анализа погрешности численных результатов – одна из фундаментальных в проблеме надежности вычислений (кроме тех людей, которые слепо верят ЭВМ).

ПОРЯДОК ПРИБЛИЖЕНИЯ

«О» большое и «о» малое (O и o) — математические обозначения для сравнения асимптотического поведения функций. Используются в различных разделах математики, но активнее всего — в математическом анализе, теории чисел и комбинаторике, а также в информатике и теории алгоритмов. $o(f)$, «о малое от f» обозначает «бесконечно малое относительно f»[1], пренебрежимо малую величину при рассмотрении f. Смысл термина «О большое» зависит от его области применения, но всегда f растёт не быстрее, чем $O(f)$, «О большое от f» (точные определения приведены ниже).

В частности: фраза «сложность алгоритма есть $O(n!)$» означает, что с увеличением параметра n, характеризующего количество входной информации алгоритма, время работы алгоритма не может быть ограничено величиной, которая растет медленнее, чем n!; фраза «функция $f(x)$ является „о“ малым от функции $g(x)$ в окрестности точки p» означает, что с приближением x к p $f(x)$ уменьшается быстрее, чем $g(x)$ (отношение $|f(x)| / |g(x)|$ стремится к нулю).

Пусть $f(x)$ и $g(x)$ — две функции, определенные в некоторой проколотой окрестности точки x_0, причем в этой окрестности g не обращается в ноль. Говорят, что: f является «О» большим от g при $x \to x_0$, если существует такая константа $C > 0$, что для всех x из некоторой окрестности точки x_0 имеет место неравенство $|f(x)| \leqslant C|g(x)|$; f является «о» малым от g при $x \to x_0$, если для любого $\varepsilon > 0$ найдется такая проколотая окрестность U'_{x_0} точки x_0, что для всех $x \in U'_{x_0}$ имеет место неравенство $|f(x)| < \varepsilon|g(x)|$.

Иначе говоря, в первом случае отношение $|f|/|g|$ в окрестности точки x_0 ограничено сверху, а во втором оно стремится к нулю при $x \to x_0$.

Обычно выражение «f является „О“ большим („о“ малым) от g» записывается с помощью равенства f(x) = O(g(x)) (соответственно, f(x) = o(g(x))).

Это обозначение очень удобно, но требует некоторой осторожности при использовании (а потому в наиболее элементарных учебниках его могут избегать). Дело в том, что это не равенство в обычном смысле, а несимметричное отношение.

В частности, можно писать $f(x) = O(g(x))$ (или $f(x) = o(g(x))$), но выражения $O(g(x)) = f(x)$ (или $o(g(x)) = f(x)$) бессмысленны. Другой пример: при $x \to 0$ верно, что $O(x^2) = o(x)$, но неверно, что $o(x) = O(x^2)$. При любом x верно $o(x) = O(x)$, т.е. бесконечно малая величина является ограниченной, но неверно, что ограниченная величина является бесконечно малой: $O(x) = o(x)$. Вместо знака равенства методологически правильнее было бы употреблять знаки принадлежности и включения, понимая O( ) и o( ) как обозначения для множеств функций, то есть, используя запись в форме $x^3 + x^2 \in O(x^2)$ или $O(x^2) \subset o(x)$ вместо, соответственно, $x^3 + x^2 = O(x^2)$ и $O(x^2) = o(x)$ Однако на практике такая запись встречается крайне редко, в основном, в простейших случаях. При использовании данных обозначений должно быть явно оговорено (или очевидно из контекста), о каких окрестностях (одно- или двусторонних; содержащих целые, вещественные или комплексные числа и т. п.) и о каких допустимых множествах функций идет речь (поскольку такие же обозначения употребляются и применительно к функциям многих переменных, к функциям комплексной переменной, к матрицам и др.).