Заруцкая Татьяна Сергеевна
бакалавр, студентка
Научный руководитель
Макарова Елена Леонидовна
канд. пед. наук, преподаватель

ФГБОУ ВО «Самарский государственный
социально-педагогический университет»
г. Самара, Самарская область

ЧИСЛЕННЫЕ МЕТОДЫ И ПРОБЛЕМЫ ТОЧНОСТИ АРИФМЕТИКИ С ПЛАВАЮЩЕЙ ТОЧКОЙ В СОВРЕМЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

Аннотация*: в статье рассматриваются вопросы принципов представления чисел с плавающей точкой в современных вычислительных системах, а также анализируются основные проблемы, возникающие при выполнении арифметических операций над ними. Особое внимание уделяется причинам накопления вычислительных погрешностей, эффектам округления, катастрофическому сокращению и вопросам численной устойчивости алгоритмов. Теоретические положения подкреплены практическими примерами программной реализации на языке Python, что позволяет продемонстрировать влияние особенностей машинной арифметики на результаты вычислений.*

Ключевые слова*: числа с плавающей точкой, численные методы, вычислительные погрешности, ошибки округления, численная устойчивость, аппроксимация вещественных чисел, арифметические операции, вычислительные системы, представление чисел, алгоритм Кэхена, точность вычислений.*

Современные вычислительные системы являются основой для решения широкого круга прикладных задач. При этом подавляющее большинство вычислений опирается на операции над вещественными числами. На практике это

нередко приводит к ситуациям, когда формально корректные алгоритмы дают неожиданные или трудно интерпретируемые результаты.

Источником подобных эффектов является способ представления вещественных чисел в памяти компьютера. Несмотря на стандартизацию формата IEEE 754, понимание ограничений машинной арифметики остаётся важной профессиональной компетенцией специалистов в области информационных технологий. Целью данной работы является анализ указанных ограничений с опорой на практические программные примеры.

Вычислительные системы используют двоичную систему счисления, что существенно ограничивает точность представления вещественных чисел. Большинство десятичных дробей не имеют конечного двоичного представления и, следовательно, хранятся в памяти в виде приближённых значений.

Рассмотрим пример, демонстрирующий приближённый характер хранения вещественного числа, результат выполнения программы показывает, что значение 0,1 представлено в памяти не точно, а в виде ближайшего двоичного приближения.

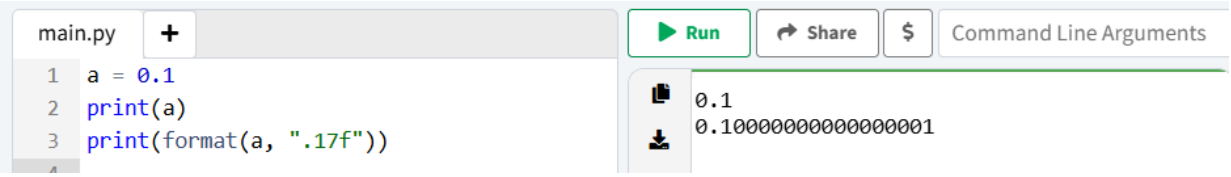

Рис. 1. Хранение вещественного числа

Данный эффект обусловлен бесконечным двоичным разложением десятичной дроби 0,1. Таким образом, любое вещественное число в формате с плавающей точкой является аппроксимацией математического значения.

Одним из наиболее наглядных проявлений ошибок округления является сложение десятичных дробей.

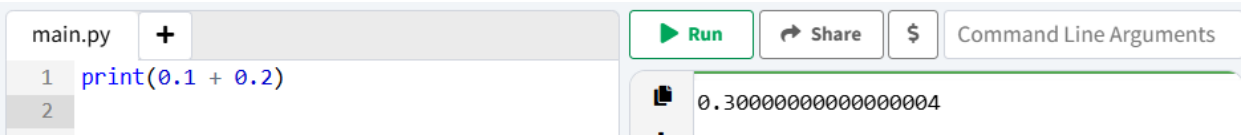

Рис. 2. Ошибка округления при сложении десятичных дробей

Ожидаемое значение 0,3 не достигается, поскольку оба операнда уже содержат погрешности округления, которые суммируются при выполнении операции. С точки зрения стандарта IEEE 754 такое поведение является корректным и предсказуемым.

При выполнении арифметических операций над числами, существенно отличающимися по величине, возможно явление потери значимости. Рассмотрим следующий пример.

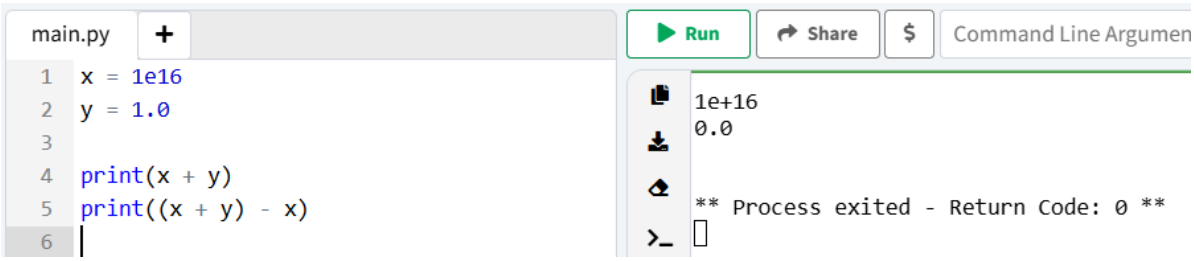

Рис. 3. Явление потери значимости

Добавление единицы к числу порядка не оказывает влияния на результат, поскольку меньшая величина оказывается «поглощённой» из-за ограниченной длины мантиссы. Подобные эффекты особенно критичны при суммировании больших массивов данных.

Катастрофическое сокращение возникает при вычитании близких по значению чисел и сопровождается резким снижением относительной точности результата.

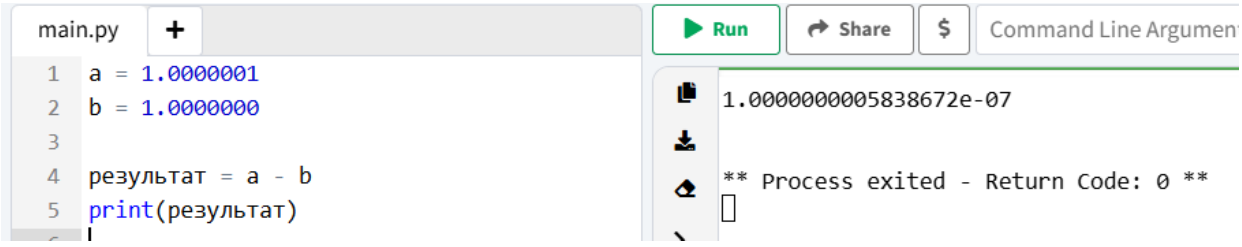

Рис. 4. Пример резкого снижения относительной точности результата

В более сложных вычислениях подобные операции могут приводить к существенным искажениям результатов, поскольку значимая часть числа формируется из разности округлённых значений.

Для снижения влияния ошибок округления применяются специальные методы организации вычислений. Одним из наиболее известных является алгоритм компенсированного суммирования Кэхена.

Пример стандартного суммирования.

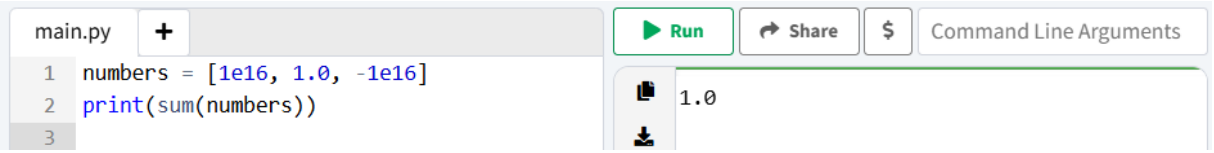

Рис. 5. Алгоритм компенсированного суммирования Кэхена

Математически ожидаемое значение равно 1,0, однако из-за потери точности результат оказывается некорректным.

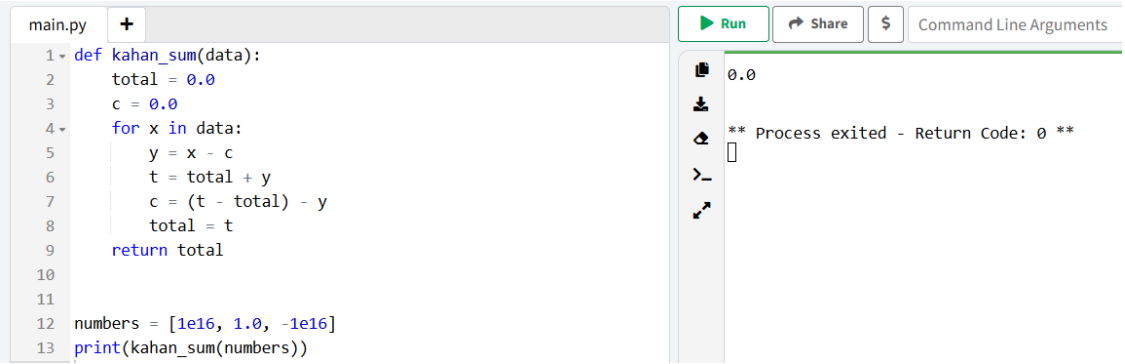

Рис. 6. Реализация алгоритма Кэхена

Получение результата 0,0 при использовании алгоритма компенсированного суммирования обусловлено потерей значимости на раннем этапе вычислений. В случае, когда малое по модулю число суммируется с существенно большей величиной, его вклад утрачивается до применения компенсационного механизма. Таким образом, эффективность алгоритма Кэхена зависит от порядка выполнения операций и не позволяет восстановить информацию, потерянную вследствие ограниченной разрядности представления чисел с плавающей точкой.

Рассмотренные примеры показывают, что ошибки машинной арифметики являются системным свойством вычислительных систем. Это имеет принципиальное значение при разработке: финансовых и экономических моделей;

инженерных и научных расчётов; алгоритмов обработки больших данных; систем машинного обучения и анализа данных.

Непонимание ограничений чисел с плавающей точкой может приводить к трудно выявляемым ошибкам, существенно влияющим на корректность результатов вычислений.

Числа с плавающей точкой являются фундаментальным инструментом современных вычислительных систем, однако их использование неизбежно связано с погрешностями округления и потерей точности. Ошибки округления, потеря значимости и катастрофическое сокращение представляют собой следствия приближённого характера машинного представления вещественных чисел.

Примеры подтверждают, что корректность вычислений определяется не только формальной правильностью алгоритма, но и способом организации вычислительного процесса. Осознанное использование устойчивых алгоритмов и понимание особенностей машинной арифметики являются необходимыми условиями разработки надёжных программных решений.

Список литературы

1. Хайэм Н. Численные методы: анализ и алгоритмы / Н. Хайэм; пер. с англ. – М.: Бином. Лаборатория знаний, 2011. – 640 с.

2. Кнут Д.Э. Искусство программирования. Т. 2: Получисленные алгоритмы / Д.Э. Кнут; пер. с англ. – 3-е изд. – М.: Вильямс, 2013. – 832 с.

3. IEEE Standard for Floating-Point Arithmetic (IEEE 754-2019) / IEEE Computer Society. – New York: IEEE, 2019. – 84 p.