УДК 004.42:519.6

ПОХИБКИ ОБЧИСЛЕНЬ У ЧИСЕЛЬНИХ МЕТОДАХ: ДЖЕРЕЛА ВИНИКНЕННЯ ТА ОЦІНКА ВПЛИВУ НА РЕЗУЛЬТАТИ

М. Ю. Кудирка, О. А. Павлюк

Анотація. У статті досліджено основні джерела виникнення похибок під час використання чисельних методів у комп'ютерних алгоритмах. Проведено глибокий аналіз похибок методу (усічення) та похибок округлення, розглянуто поняття абсолютної та відносної похибок, а також вплив стандарту машинного подання дійсних чисел IEEE 754 на накопичення похибок. На основі розрахункових даних продемонстровано вплив цих похибок на точність результатів ітераційних обчислень на прикладі розкладу функцій у ряд Маклорена та задачі чисельного диференціювання. Розглянуто проблему катастрофічного скасування та алгоритмічні підходи до підвищення обчислювальної стійкості програмного забезпечення. Отримані результати дають змогу оцінити межі застосування наближених методів обчислень, визначити оптимальні параметри кроку та оптимізувати алгоритми для мінімізації втрати точності в умовах скінченної розрядності машинної арифметики.

Ключові слова: чисельні методи, абсолютна похибка, відносна похибка, похибка округлення, числова стійкість.

Вступ. У сучасних інформаційних системах чисельні методи широко застосовуються для розв'язування задач моделювання, оптимізації, аналізу даних та інженерних розрахунків. Однак практичне використання цих методів супроводжується неминучим виникненням обчислювальних похибок [2]. Сучасні обчислювальні системи здатні виконувати мільярди операцій з рухомою комою за секунду. Водночас використання методів обчислень як основного інструменту моделювання неминуче стикається з фундаментальною проблемою точності [1].

Будь-який чисельний метод є наближеним, а використання комп'ютерної техніки вносить додаткові апаратні обмеження через скінченну розрядність машинного подання чисел [3; 5]. Розуміння природи цих похибок є критичним, оскільки ігнорування машинного округлення або неправильний вибір кроку ітераційного методу може призвести до суттєвого спотворення результатів обчислень або некоректної роботи алгоритму [4]. Відповідно аналіз та суворий контроль похибок обчислень є обов'язковим етапом розробки надійних алгоритмів.

Метою статті є комплексне дослідження природи виникнення обчислювальних похибок у чисельних методах, аналіз впливу машинного подання чисел на точність, а також математична та емпірична демонстрація впливу цих похибок на результати розрахунків. Окремим завданням є розгляд алгоритмічних підходів до оптимізації програмного коду для запобігання накопиченню машинних похибок.

Основна частина. Теорія похибок є фундаментальним складником курсу чисельних методів та проєктування алгоритмів [3]. У процесі розв'язання математичних задач комп'ютерними засобами виникають різні види похибок, які в сукупності визначають загальну точність кінцевого результату. Основними мірами точності в обчислювальній математиці традиційно виступають абсолютна та відносна похибки. Нехай x – точне (істинне) значення деякої величини, а x^* – її наближене значення, отримане внаслідок роботи комп'ютерного алгоритму або вимірювання. Тоді абсолютна похибка Δx визначається як модуль різниці між точним та наближеним значеннями [4]:

$$\Delta x = |x - x^*|. \tag{1}$$

Оскільки абсолютна похибка показує лише розмірність помилки, але не дає уявлення про її вагомість у контексті самої величини, для оцінки якості наближення використовують відносну похибку δx [3]. Вона визначається як відношення абсолютної похибки до модуля точного (або, на практиці, наближеного) значення і часто виражається у відсотках:

$$\delta x = \frac{\Delta x}{|x^*|} \cdot 100\ \%. \tag{2}$$

За джерелом виникнення всі похибки, з якими стикається фахівець під час реалізації чисельних алгоритмів, класифікують на три фундаментальні групи [1; 2].

Перша група – похибки вхідних даних, які пов'язані з неточністю фізичних вимірювань або використанням наближених фізичних констант. Алгоритм не може їх усунути, він лише трансформує їх у процесі обчислень.

Друга група – похибки методу. Вони виникають внаслідок заміни нескінченного математичного процесу скінченним алгоритмічним [3].

Третя, критично важлива для розробки програмного забезпечення група, – це похибки машинного округлення, пов'язані з апаратними обмеженнями комп'ютера [5].

Фундаментальний вплив на виникнення похибок округлення має машинна арифметика та стандарт IEEE 754. У більшості сучасних мов програмування та на апаратному рівні процесорів для роботи з дійсними числами використовується стандарт IEEE 754 [5]. Відповідно до нього, число з рухомою комою подається у вигляді $x = (-1)^s \cdot M \cdot 2^E$, де s – біт знака, M – мантиса, E – експонента. Найбільш поширеними типами у програмуванні є одинарна точність (тип float, виділяється 32 біти, забезпечує приблизно 7 значущих десяткових цифр) та подвійна точність (тип double, виділяється 64 біти, забезпечує 15–17 значущих десяткових цифр).

Через скінченну кількість бітів, виділених під мантису, більшість дійсних чисел не можуть бути подані в пам'яті комп'ютера абсолютно точно [4]. Наприклад, десятковий дріб 0.1 у двійковій системі числення є нескінченним періодичним дробом. Його усічення під час запису у змінну породжує первинну похибку округлення. Головною характеристикою точності обчислювальної системи є машинне епсилон (ε) – найменше додатне число, для якого виконується умова $1.0 + \varepsilon > 1.0$ у машинній арифметиці [5]. Для типу double це значення становить приблизно 2.22×10^{-16}.

Варто зазначити, що в розробці програмного забезпечення іноді застосовують базово-десяткові типи даних високої точності (наприклад, тип decimal у середовищі .NET). Він займає 128 бітів у пам'яті і забезпечує 28–29 значущих цифр, зменшуючи похибки, пов'язані з двійковим поданням деяких десяткових дробів. Проте обчислення з цим типом виконуються програмно, а не на рівні інструкцій співпроцесора (FPU), що уповільнює роботу алгоритму в десятки разів. Тому для інтенсивних математичних обчислень та чисельних методів стандартом де-факто залишається тип double, що вимагає ретельного контролю похибок [2].

Для наочної демонстрації впливу похибки методу (усічення) розглянемо класичний процес обчислення експоненційної функції $f(x) = e^x$ у точці $x = 1$ за допомогою розкладу в ряд Маклорена [3]:

$$e^x \approx \sum_{i=0}^{n} \frac{x^i}{i!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!}. \tag{3}$$

Залишковий член ряду, який визначає теоретичну похибку методу, описується формулою Лагранжа [4]: $R_n(x) = \frac{e^c}{(n+1)!} x^{n+1}$. Точне (аналітичне) значення з точністю до дев'ятого знака становить $e^1 \approx 2.718281828$. Проведені розрахунки для різної кількості ітерацій n (кількості членів ряду) дають змогу оцінити динаміку зміни похибок (табл. 1).

Таблиця 1

Оцінка похибок усічення при ітераційному обчисленні e^1

n (кількість ітерацій)	Наближене значення x^*	Абсолютна похибка Δx	Відносна похибка δx (%)
1	2.000000000	0.718281828	26.4241
2	2.500000000	0.218281828	8.0301
3	2.666666667	0.051615161	1.8988
4	2.708333333	0.009948495	0.3660
5	2.716666667	0.001615161	0.0594
7	2.718253968	0.000027860	0.0010
10	2.718281801	0.000000027	< 0.0001

Джерело: розраховано авторами

Аналіз отриманої аналітичної моделі показує, що зі збільшенням n похибка методу стрімко зменшується (швидкість збіжності є факторіальною). Під час проєктування блок-схем таких алгоритмів ключовим є правильний вибір критерію зупинки циклу while або do-while [1]. Замість жорсткого задання кількості ітерацій n надійний алгоритм повинен зупинятися, коли поточний доданок стає меншим за задану точність: $|term| < \varepsilon$. Однак на практиці за дуже великих значень n факторіал у знаменнику стає настільки великим, що черговий доданок стає меншим за машинне епсилон. На цьому етапі комп'ютер перестає фіксувати додавання, і подальше виконання циклу є безглуздим витрачанням процесорного часу [5].

Щоб глибше дослідити конфлікт між похибкою методу та похибкою округлення, доцільно розглянути задачу чисельного знаходження першої похідної функції $f(x)$ за допомогою центральної різницевої схеми [3; 4].

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}. \tag{4}$$

Згідно з розкладом у ряд Тейлора, похибка усічення для цієї формули пропорційна квадрату кроку: $E_{method} \approx \frac{h^2}{6}|f'''(x)|$. Теоретично для отримання ідеальної точності потрібно брати нескінченно малий крок $h \to 0$. Проте в реальній машинній арифметиці обчислення функції $f(x)$ відбувається з похибкою округлення ε [4]. Тому чисельник насправді обчислюється як $(f(x+h) \pm \varepsilon) - (f(x-h) \pm \varepsilon)$. В найгіршому випадку сумарна абсолютна похибка обчислення чисельника становить 2ε, а машинна похибка округлення для похідної дорівнює $\frac{\varepsilon}{h}$.

Повна обчислювальна похибка є сумою цих двох складників:

$$E_{total}(h) = \frac{h^2}{6}|f'''(x)| + \frac{\varepsilon}{h}. \tag{5}$$

Ця функція має яскраво виражений мінімум [4]. Якщо ми почнемо надмірно зменшувати крок h, похибка методу спадатиме, але похибка округлення почне катастрофічно зростати через ділення на надзвичайно мале число. Цей феномен демонструє обчислювальний парадокс: безмежне подрібнення кроку призводить до деградації результату, а не до його покращення. Прирівнявши похідну похибки до нуля, можна знайти оптимальний крок $h_{opt} \approx \sqrt[3]{\frac{3\varepsilon}{|f'''(x)|}}$, який для типу double становить приблизно 10^{-5} [3]. Якщо розробник задасть крок $h = 10^{-15}$, тип вичерпає свій ліміт значущих цифр, і результатом диференціювання буде обчислювальний «шум», що може призвести до різкого зростання відносної похибки та втрати достовірності результату.

Одним із найнебезпечніших наслідків машинної арифметики є так зване «катастрофічне скасування» – втрата значущих цифр під час віднімання двох дуже близьких за значенням чисел [2]. Коли ми віднімаємо 1.23456789 та 1.23456788, комп'ютер отримує 0.00000001. Вісім старших точних значущих цифр знищуються, залишаючи результат, що складається переважно з «машинного сміття», яке накопичилося в молодших розрядах мантиси. Щоб уникнути цього, алгоритми необхідно перепроєктовувати на рівні математичних формул [4]. Наприклад, вираз $\sqrt{x+1} - \sqrt{x}$ для великих x страждає від катастрофічного скасування. Розумним алгоритмічним кроком є множення чисельника та знаменника на спряжений вираз: $\frac{1}{\sqrt{x+1}+\sqrt{x}}$. Ця алгебраїчно тотожна формула в програмному коді виконується без втрати точності, оскільки замінює операцію віднімання близьких чисел на додавання.

До того ж накопичення похибки округлення є особливо критичним у багатокрокових ітераційних процесах, як-от знаходження суми великих масивів даних [5]. Під час звичайного циклічного додавання мільйонів чисел float дрібні доданки перестають впливати на загальну суму, оскільки вона стає занадто великою, порівняно з ними. Для вирішення цієї проблеми в комп'ютерних науках застосовується алгоритм підсумовування Кахана, який вводить окрему змінну для накопичення та компенсації дрібних похибок відкидання на кожній ітерації циклу, значно підвищуючи загальну точність без переходу на повільніші типи даних [4].

Рівень чутливості математичної задачі до похибок у вхідних даних або проміжних обчисленнях характеризується числом обумовленості [2]. Для погано обумовлених задач навіть мінімальні похибки округлення призводять до абсолютно хибних фінальних результатів, що вимагає використання специфічних методів регуляризації або перегляду всієї структури алгоритму та його блок-схеми.

Висновки. Проведене дослідження показало, що похибки обчислень є невід'ємним і часто визначальним фактором будь-якого комп'ютерного моделювання. Точність результату залежить як від обраного математичного апарату (що генерує похибку усічення) [3], так і від апаратних та програмних обмежень обчислювальної системи (похибки округлення стандарту IEEE 754) [5]. Виявлено фундаментальний обчислювальний парадокс: надмірне зменшення кроку або збільшення кількості ітерацій для мінімізації похибки методу неминуче призводить до катастрофічного накопичення похибок машинного округлення.

Існує чітка математична межа (оптимальний крок), перетин якої погіршує роботу алгоритму. Протидія цим явищам вимагає не просто сліпого переведення математичних формул у програмний код, а свідомого проєктування стійких алгоритмів: використання компенсаційних методів підсумовування, рефакторингу алгебраїчних виразів для уникнення катастрофічного скасування та встановлення динамічних умов виходу з ітераційних циклів на основі машинного епсилон [1; 2]. Отже, розробка ефективного та надійного програмного забезпечення вимагає від фахівців глибокого розуміння природи машинної арифметики та обов'язкової попередньої оцінки стійкості застосовуваних чисельних методів.

Abstract. This article investigates the main sources of errors when using numerical methods in computer algorithms. A thorough analysis of method errors (truncation) and rounding errors is conducted; the concepts of absolute and relative errors are examined, as well as the impact of the IEEE 754 standard for the representation of real numbers on the accumulation of errors. Based on computational data, the impact of these errors on the accuracy of iterative computation results is demonstrated using the example of the Maclaurin series expansion of functions and the numerical differentiation problem. The problem of catastrophic cancellation and algorithmic approaches to improving the computational stability of software are considered. The results obtained allow us to assess the limits of application of approximate computational methods, determine optimal step parameters, and optimize algorithms to minimize loss of accuracy under the conditions of finite precision in machine arithmetic.

Keywords: numerical methods, absolute error, relative error, rounding error, truncation error, computer calculations, machine epsilon, numerical stability, Kahan summation algorithm.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Комп'ютерне моделювання процесів та систем. Чисельні методи: підручник / С. П. Вислоух, О. В. Волошко, Г. С. Тимчик, М. В. Філіппова. Київ: КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2021. 228 с.
2. Третиник В. В., Любашенко Н. Д. Методи обчислень. Частина 1. Чисельні методи алгебри: навч. посіб. Київ: КПІ ім. Ігоря Сікорського, 2023. 182 с.
3. Голубєва К. М., Кашпур О. Ф., Клюшин Д. А. Чисельні методи: навч. посіб. Київ: ВПЦ «Київський університет», 2022. 145 с.
4. Sauer T. Numerical Analysis. 3rd ed. (Updated). Pearson, 2021. 664 p.
5. Методи та алгоритми комп'ютерних обчислень. Теорія і практика: підручник / Р. Н. Квєтний, Я. В. Іванчук, І. В. Богач, О. Ю. Софина, М. В. Барабан. Вінниця: ВНТУ, 2023. 280 с.

УДК 004.738.5:004.896.2:004.421.2

ЗБІР ТА СТАНДАРТИЗАЦІЯ ГЕТЕРОГЕННИХ ДАНИХ У ХМАРНИХ АНАЛІТИЧНИХ СЕРЕДОВИЩАХ

Д. Ю. Кохан, Т. В. Січко

Анотація. У статті розглянуто проблему інтеграції різнорідних за форматом і походженням даних у єдине хмарне сховище. Для її розв'язання запропоновано ETL-пайплайн на основі Apache Airflow, Google BigQuery та dbt. Описано трирівневу модель трансформації (Raw – Staging – Mart) і ключові операції стандартизації: уніфікацію типів, видалення дублікатів, заповнення пропусків і нормалізацію схем. Показано, що якість підготовлених даних визначає точність аналітичних метрик.

Ключові слова: гетерогенні дані, ETL-пайплайн, BigQuery, dbt, стандартизація даних.

Вступ. Більшість сучасних підприємств зберігають операційні дані одразу в кількох несумісних системах: реляційних базах даних, хмарних сервісах, файлових звітах і зовнішніх API.