

<https://doi.org/10.31891/2307-5732-2026-361-71>

УДК 004.4:004.056

ПОРОХНАВЕЦЬ БОГДАН

Харківський національний університет радіоелектроніки

e-mail: bohdan.porokhnavecs@nure.ua

ШУБІН ІГОР

Харківський національний університет радіоелектроніки

<https://orcid.org/0000-0002-1073-023X>

e-mail: igor.shubin@nure.ua

МЕТОДИ ОЦІНЮВАННЯ СКЛАДНОСТІ ПРОГРАМ З ВИСОКОТОЧНИМИ ОБЧИСЛЕННЯМИ У РОЗПОДІЛЕНИХ СИСТЕМАХ

В роботі наведено результати аналізу існуючих метрик оцінювання складності програмного забезпечення з точки зору можливостей їх застосування до програм з високоточними обчисленнями, що функціонують у розподілених обчислювальних системах. Розглянуто метрики розміру програм, метрики складності потоку керування та метрики складності потоку даних, а також визначено їх основні обмеження при використанні високоточної арифметики. Показано, що класичні метрики не враховують залежність обчислювальної складності від довжини мантиси числових даних та вартості арифметичних операцій. Обґрунтовано доцільність урахування параметрів точності обчислень для більш адекватного прогнозування часових витрат виконання програм у розподілених обчислювальних середовищах.

Ключові слова: метрики програмного забезпечення, складність програм, високоточні обчислення, мантиса, арифметичні операції.

POROKHNAVETS BOHDAN, SHUBIN IHOR

Kharkiv National University of Radio Electronics

SOFTWARE COMPLEXITY METRICS FOR HIGH-PRECISION COMPUTATIONS IN DISTRIBUTED SYSTEMS

Modern software systems increasingly rely on numerical algorithms that require high or arbitrary precision, especially in scientific computing, numerical modeling, optimization, and data analysis. The use of high-precision arithmetic significantly affects the computational behavior of programs, since the cost of arithmetic and transcendental operations depends on the precision level of numerical data representation. In such conditions, predicting program execution time and estimating computational complexity become non-trivial tasks, particularly in distributed computing environments.

This paper presents an analysis of existing software complexity metrics with respect to their applicability to programs performing high-precision computations in distributed systems. Classical software metrics, including size metrics, control flow complexity metrics, and data flow complexity metrics, are considered and systematized. Their main limitations are identified in the context of high-precision arithmetic, where the assumption of constant-time arithmetic operations is no longer valid.

It is shown that classical complexity metrics do not account for the dependence of computational cost on the mantissa length of numerical variables and the actual cost of arithmetic operations. As a result, programs with relatively simple control structures but intensive high-precision computations may require significantly more computational resources than programs with complex control logic implemented using standard floating-point arithmetic.

The study substantiates the necessity of extending existing complexity evaluation approaches by incorporating parameters related to numerical precision and operation cost. The computational complexity of a program is proposed to be considered as a function of the types of arithmetic operations, their execution frequency, and the mantissa length of numerical data. Such an approach allows for a more accurate estimation of execution time and resource consumption, which is especially important for programs executed in distributed computing systems where communication overhead and data transfer costs play a significant role.

The obtained results can be used for preliminary performance estimation, rational resource allocation, and the selection of appropriate parallelization strategies for high-precision numerical software in distributed environments.

Keywords: software metrics, program complexity, high-precision arithmetic, mantissa length, distributed computing systems.

Стаття надійшла до редакції / Received 13.12.2025

Прийнята до друку / Accepted 11.01.2026

Опубліковано / Published 29.01.2026



This is an Open Access article distributed under the terms of the [Creative Commons CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/)

© Порохнавець Богдан, Шубін Ігор

Постановка проблеми у загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями

Розвиток сучасних програмних систем супроводжується ускладненням алгоритмічної структури та зростанням обсягів обчислень, що реалізуються у програмному забезпеченні. У автоматизованих та інтелектуальних системах все частіше застосовуються математичні моделі й алгоритми, орієнтовані на оброблення великих масивів даних та виконання обчислень з підвищеними вимогами до точності [1]. За таких умов задача коректного оцінювання складності програмного забезпечення набуває принципового значення вже на етапі його проектування.

Особливу складність становлять програми з високоточними обчисленнями, у яких використовується арифметика з довільною або змінною довжиною мантиси. У таких програмах часові витрати виконання істотно залежать не лише від логіки керування, але й від характеристик числових обчислень, зокрема вартості арифметичних операцій та точності представлення даних. Нехтування цими факторами призводить до суттєвих помилок при оцінюванні складності та може спричинити неефективні проектні рішення.

Проблема особливо загострюється у розподілених обчислювальних системах, де на ефективність виконання програм додатково впливають комунікаційні витрати та обмеженість доступних ресурсів. Як

показують дослідження у сфері планування оброблення даних, точність попередньої оцінки обчислювального навантаження безпосередньо визначає ефективність використання розподілених ресурсів [3]. У випадку високоточних обчислень похибки такого оцінювання можуть мати критичні наслідки для продуктивності системи.

Крім того, тенденція до ускладнення програмних систем ускладнює їх супровід і повторне використання програмних компонентів, що посилює потребу у формалізованих і універсальних методах аналізу програмного забезпечення [2]. У цьому контексті класичні метрики складності, орієнтовані переважно на структурні характеристики програм, виявляються недостатніми та потребують адаптації до умов використання високоточної арифметики.

Отже, постає науково й практично значуща проблема аналізу можливостей застосування існуючих метрик оцінювання складності програмного забезпечення до програм з високоточними обчисленнями у розподілених системах, розв'язання якої сприятиме підвищенню ефективності проектування та експлуатації складних програмних систем.

Аналіз досліджень та публікацій

Питання оцінювання складності програмного забезпечення розглядається у значній кількості наукових робіт, у яких запропоновано різні підходи до кількісного аналізу програмного коду. Найбільш поширеними є метрики, що базуються на оцінюванні розміру програм, складності потоку керування та особливостей використання даних. Ці метрики широко застосовуються для аналізу якості програмного забезпечення та прогнозування трудомісткості розроблення.

У роботах, присвячених формалізованим методам проектування програмних систем, підкреслюється важливість використання математичних апаратів для аналізу складних обчислювальних процесів та програмних структур [1]. Такі підходи створюють теоретичне підґрунтя для побудови формальних моделей програм, однак, як правило, орієнтовані на стандартні обчислювальні схеми.

Окрему групу становлять дослідження, присвячені розподіленим обчислювальним системам та методам планування оброблення даних в умовах обмежених ресурсів [3]. У цих роботах розглядаються питання балансування навантаження та ефективного використання обчислювальних вузлів, проте характеристики високоточних числових обчислень зазвичай не включаються до моделей оцінювання складності.

Також у сучасних публікаціях, що стосуються проектування та супроводу програмного забезпечення, відзначається зростання складності програмних систем і потреба у вдосконаленні методів їх аналізу та повторного використання [2]. Водночас проблема адаптації класичних метрик складності до програм з високоточними обчисленнями у розподілених середовищах залишається недостатньо опрацьованою.

Таким чином, аналіз наукових публікацій свідчить про наявність значного теоретичного доробку у сфері оцінювання складності програмного забезпечення, але водночас вказує на відсутність цілісних підходів, що враховують вплив точності числових обчислень на складність програм у розподілених обчислювальних системах.

Формулювання цілей статті

Метою роботи є аналіз існуючих метрик оцінювання складності програмного забезпечення та дослідження можливостей їх застосування для програм з високоточними обчисленнями, що функціонують у розподілених обчислювальних системах.

Для досягнення поставленої мети в роботі передбачається розв'язання таких завдань:

- проаналізувати основні групи метрик оцінювання складності програмного забезпечення, зокрема метрики розміру програм, метрики складності потоку керування та метрики складності потоку даних;
- визначити обмеження застосування класичних метрик складності до програм з високоточними обчисленнями;
- дослідити вплив довжини мантиси числових даних на обчислювальну складність програм;
- обґрунтувати доцільність урахування вартості арифметичних операцій при оцінюванні складності програм з високоточною арифметикою;
- проаналізувати особливості оцінювання складності програм у розподілених обчислювальних системах.

Класифікація метрик оцінювання складності програмного забезпечення

Метрики оцінювання складності програмного забезпечення є інструментом кількісного аналізу програм і застосовуються для оцінювання трудомісткості розроблення, супроводу та прогнозування часових витрат виконання програм. Залежно від аспекту програмного забезпечення, який аналізується, метрики поділяють на кілька основних груп, зокрема метрики розміру програм, метрики складності потоку керування та метрики складності потоку даних.

Метрики розміру програм базуються на кількісному аналізі елементів програмного коду, таких як оператори, операнди або кількість рядків коду. Вони дозволяють оцінити масштаб програмного продукту та логічну насиченість коду, однак істотно залежать від стилю програмування та не враховують складність виконуваних обчислень. У випадку програм з високоточними обчисленнями метрики розміру не відображають реальні часові витрати, оскільки одна арифметична операція з довільною точністю може мати значно вищу вартість, ніж аналогічна операція у стандартній машинній арифметиці.

Метрики складності потоку керування, зокрема цикломатична складність, орієнтовані на аналіз логічної структури програм і дозволяють оцінити кількість незалежних шляхів виконання. Такі метрики є

корисними для аналізу тестованості програм і складності керування, однак не враховують характер і вартість обчислень, що виконуються всередині умовних операторів і циклів.

Метрики складності потоку даних аналізують використання змінних у програмі та інформаційні залежності між ними. Вони дозволяють оцінити насиченість програм інформаційними потоками, проте також не враховують точність числових обчислень і залежність вартості операцій від довжини мантиси.

Обмеження класичних метрик для програм з високоточними обчисленнями

Класичні метрики оцінювання складності програмного забезпечення були розроблені з припущенням використання стандартної машинної арифметики, у якій довжина мантиси числових даних є фіксованою. У таких умовах вартість виконання базових арифметичних операцій вважається приблизно сталою і не враховується при оцінюванні складності програм.

У програмах з високоточними обчисленнями це припущення не виконується. Довжина мантиси числових змінних може змінюватися залежно від вимог до точності, а вартість виконання арифметичних і трансцендентних операцій істотно зростає зі збільшенням точності. Унаслідок цього програми з простою структурою керування, але з інтенсивним використанням високоточних обчислень, можуть вимагати значно більших обчислювальних ресурсів, ніж програми зі складною логікою, але зі стандартною арифметикою.

Таким чином, застосування класичних метрик без урахування точності обчислень може призводити до суттєвого заниження або завищення оцінок обчислювальної складності програм. Це особливо критично для задач прогнозування часу виконання програм і планування обчислювальних ресурсів.

Інтерпретація метрик складності з урахуванням параметрів високоточної арифметики

Аналіз складності програмного забезпечення з використанням класичних метрик передбачає подальшу інтерпретацію отриманих числових значень з метою оцінювання трудомісткості виконання програм та ефективності їх реалізації. У випадку програм з високоточними обчисленнями така інтерпретація потребує додаткового аналізу, оскільки значення метрик не відображають реальної вартості обчислювальних операцій.

Зокрема, однакові значення метрик складності потоку керування або розміру програм можуть відповідати програмам з принципово різними характеристиками виконання, якщо в одній із них використовуються операції стандартної арифметики, а в іншій — високоточні числові обчислення. У такому випадку формальне порівняння значень метрик без урахування параметрів точності може призводити до хибних висновків щодо ефективності програмних рішень.

Важливим аспектом інтерпретації метрик складності для програм з високоточною арифметикою є врахування типів арифметичних операцій та їх відносної вартості. Наприклад, наявність великої кількості операцій множення або ділення багаторозрядних чисел може мати значно більший вплив на часову складність виконання програми, ніж збільшення кількості умовних переходів або змінних.

Крім того, при аналізі програм з високоточними обчисленнями доцільно розглядати залежність між значеннями структурних метрик та параметрами точності числових даних. Такий підхід дозволяє інтерпретувати результати оцінювання складності не як абсолютні показники, а як відносні характеристики, що залежать від заданого рівня точності обчислень.

У контексті розподілених обчислювальних систем інтерпретація метрик складності обтяжується додатковими факторами, пов'язаними з передаванням даних та синхронізацією між обчислювальними вузлами. Високоточні числові дані мають більший обсяг, що впливає на час комунікацій і може змінювати співвідношення між обчислювальними та комунікаційними витратами.

Таким чином, результати оцінювання складності програм з використанням класичних метрик доцільно розглядати як основу для подальшого аналізу, який повинен включати параметри точності обчислень та характеристики виконання арифметичних операцій. Це дозволяє отримати більш адекватну картину реальних обчислювальних витрат і підвищити практичну цінність метрик складності при аналізі програм з високоточними обчисленнями.

Вплив довжини мантиси на обчислювальну складність програм

У високоточній арифметиці числові дані подаються у вигляді чисел з довільною довжиною мантиси. Нехай l — довжина мантиси, виражена кількістю розрядів. Тоді часова складність виконання арифметичних операцій стає функцією параметра l .

Додавання та віднімання високоточних чисел мають лінійну складність і пропорційні довжині мантиси, тобто $O(l)$. Множення і ділення чисел реалізуються за алгоритмами зі складністю не нижче $O(l^2)$, а у випадку використання ітераційних або спектральних алгоритмів можуть мати ще вищу складність. Обчислення трансцендентних функцій, таких як експонента або тригонометричні функції, зазвичай потребує застосування рядів або ітераційних методів і характеризується складністю не нижче $O(l^2 \log l)$.

Таким чином, довжина мантиси є одним з ключових параметрів, що визначають обчислювальну складність програм з високоточними обчисленнями. Збільшення точності призводить до нелінійного зростання часових витрат, що повинно враховуватися при оцінюванні складності програм.

Оцінювання вартості арифметичних операцій

Для більш адекватного оцінювання складності програм з високоточними обчисленнями доцільно враховувати вартість окремих арифметичних операцій з урахуванням точності числових даних. У цьому випадку обчислювальну складність програми можна подати як суму часових витрат виконання окремих операцій:

$$C = \sum_{i=1}^n w_i \cdot T_i(l),$$

де $T_i(l)$ — часова складність i -ої операції при довжині мантиси l , а w_i — кількість виконань відповідної операції у програмі.

Такий підхід дозволяє перейти від суто структурного аналізу програм до оцінювання реальних обчислювальних витрат і створює передумови для більш точного прогнозування часу виконання програм.

Результати порівняльного оцінювання складності програм

Для ілюстрації обмежень класичних метрик складності та можливостей їх застосування до програм з високоточними обчисленнями розглянемо умовний приклад програмного фрагмента, що використовується в автоматизованій або інтелектуальній системі. Подібні програмні компоненти широко застосовуються в системах оброблення знань, багатоагентних архітектурах та інтелектуальних інформаційних системах, де формалізовані моделі та алгебраїчні методи відіграють ключову роль [1, 4, 5].

Нехай програма має фіксовану структуру керування та сталу цикломатичну складність. З точки зору класичних метрик розміру програм і потоку керування така програма оцінюється як відносно проста. Однак у процесі виконання вона реалізує числові обчислення з використанням високоточної арифметики, зокрема операції множення, ділення та обчислення функцій над числами з довільною довжиною мантиси.

У цьому випадку реальні часові витрати виконання програми зростають разом із підвищенням точності обчислень, що не відображається у значеннях класичних метрик. Якщо ж оцінювання складності здійснювати з урахуванням вартості арифметичних операцій та параметрів точності числових даних, отримується інша картина обчислювальної складності, яка більш адекватно відображає реальні витрати ресурсів.

Подібні особливості є характерними для програм, що функціонують у розподілених обчислювальних системах, де важливу роль відіграють процеси планування оброблення даних і балансування навантаження між вузлами [3]. У таких системах некоректна оцінка складності програм з високоточними обчисленнями може призводити до неефективного використання обчислювальних ресурсів.

Крім того, у системах оброблення знань та інтелектуальних інформаційних системах, описаних у [6, 7], складність програмного забезпечення визначається не лише структурними характеристиками коду, але й особливостями оброблення числових і логічних даних. Це додатково підтверджує обмеженість класичних метрик складності при аналізі програм з високоточними обчисленнями.

Таким чином, проведене порівняння показує, що застосування класичних метрик без урахування параметрів високоточної арифметики може призводити до заниження оцінок обчислювальної складності програмного забезпечення. Використання підходів, що враховують точність обчислень і вартість арифметичних операцій, дозволяє отримати більш адекватні результати оцінювання складності програм.

Особливості застосування метрик у розподілених обчислювальних системах

У розподілених обчислювальних системах обчислювальна складність програм доповнюється накладними витратами, пов'язаними з передаванням даних між обчислювальними вузлами. Для високоточних чисел обсяг переданих даних зростає зі збільшенням довжини мантиси, що може суттєво впливати на ефективність паралельного виконання програм.

У деяких випадках витрати на передавання результатів обчислень можуть перевищувати вигоду від паралельного виконання. Тому при оцінюванні складності програм з високоточними обчисленнями у розподілених системах необхідно враховувати як вартість арифметичних операцій, так і характеристики обчислювального середовища.

Висновки з даного дослідження

і перспективи подальших розвідок у даному напрямі

У статті проведено аналіз існуючих підходів до оцінювання складності програмного забезпечення з урахуванням специфіки програм з високоточними обчисленнями, що функціонують у розподілених обчислювальних системах. Показано, що класичні метрики складності, орієнтовані переважно на структурні характеристики програм, не дозволяють адекватно оцінювати обчислювальні витрати у випадку використання арифметики з довільною точністю.

Встановлено, що довжина мантиси числових даних є одним із визначальних чинників, які впливають на часову складність виконання програм з високоточними обчисленнями. Зі збільшенням точності представлення чисел вартість виконання арифметичних і трансцендентних операцій зростає нелінійно, що повинно враховуватися при прогнозуванні часу виконання програм і плануванні обчислювальних ресурсів.

Обґрунтовано доцільність доповнення класичних метрик оцінювання складності програмного забезпечення параметрами, які враховують вартість арифметичних операцій та залежність обчислювальної складності від довжини мантиси. Запропонований підхід дозволяє перейти від суто структурного аналізу програм до більш змістовного оцінювання реальних обчислювальних витрат, що є особливо важливим для програм, які виконуються у розподілених обчислювальних середовищах.

Показано, що у розподілених обчислювальних системах додатковим фактором, який впливає на ефективність виконання програм з високоточними обчисленнями, є накладні витрати, пов'язані з передаванням даних між обчислювальними вузлами. Для високоточних чисел обсяг переданих даних зростає зі збільшенням довжини мантиси, що може знижувати ефективність паралельного виконання і потребує врахування на етапі проектування програмних систем.

Перспективи подальших досліджень у даному напрямі пов'язані з розробленням формалізованих моделей виконання програм з високоточними обчисленнями, які дозволять поєднати оцінювання структурної складності програм із моделями вартості обчислень і комунікацій у розподілених середовищах. Зокрема, доцільним є використання формальних моделей паралельних процесів для аналізу критичних обчислювальних ділянок та більш точного прогнозування часу виконання програм.

Отримані результати порівняльного аналізу свідчать про те, що використання класичних метрик складності без урахування параметрів високоточної арифметики призводить до систематичних похибок в оцінюванні обчислювальної складності програмного забезпечення. Проведене зіставлення класичних метрик із підходами, що враховують вартість арифметичних операцій та параметри точності числових даних, показало обмеженість їх застосування для програм з інтенсивними високоточними обчисленнями, зокрема у розподілених обчислювальних системах.

Запропонований підхід до аналізу обмежень існуючих метрик створює підґрунтя для подальшого розвитку методів оцінювання складності програм, орієнтованих на високоточні обчислення та розподілені обчислювальні середовища.

Література

1. Shubin, I. Snisar, S., Litvin, S. (2021). Formalization and Application of Algebraic Methods in Automated Intelligent Systems. *IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T)*. Kharkiv, Ukraine, P. 67–70. DOI: <http://doi.org/10.1109/PICST54195.2021.9772174>
2. Шубін І., Каратаєв О. (2023). Проблеми повторного використання знань у процесі проектування програмних систем. *Сучасний стан наукових досліджень та технологій в промисловості*. No 2 (24). С. 62–71. DOI: <http://doi.org/10.30837/itssi.2023.24.062>
3. Козирев А., Шубін І. (2023). Метод планування завдань оброблення даних у розподілених системах з обмеженою інформацією про доступні ресурси. *Сучасний стан наукових досліджень та технологій в промисловості*. No. 3 (25). С. 27–39. DOI: <https://doi.org/10.30837/ITSSI.2023.25.027>
4. Sharonova, N., Doroshenko, A., Cherednichenko, O. (2021). Issues of Fact-based Information Analysis. *5th International Conference on Computational Linguistics and Intelligent Systems (COLINS-2021)*. CEUR Workshop Proceedings, Vol. 2136. URL: <https://ceur-ws.org/Vol-2136/10000011.pdf>
5. Karataiev, O. Shubin, I. (2023). Formal Model of Multi-Agent Architecture of a Software System Based on Knowledge Interpretation. *Radioelectronic and Computer Systems*. No. 4 (108), P. 53–64. DOI: <http://doi.org/10.32620/reks.2023.4.05>
6. Sarker, U., Deraman, A. B., Hasan, R. (2018). Descriptive Logic for Software Engineering Ontology: Aspect Software Quality Control. *4th International Conference on Computer and Information Sciences (ICCOINS)*. Kuala Lumpur, Malaysia, P. 1–5. DOI: <http://doi.org/10.1109/ICCOINS.2018.8510585>
7. Karataiev, O., Sitnikov, D., Sharonova, N. (2023). A Method for Investigating Links between Discrete Data Features in Knowledge Bases in the Form of Predicate Equations. *7th International Conference on Computational Linguistics and Intelligent Systems (COLINS-2023)*. CEUR Workshop Proceedings, Vol. 3387. URL: <https://ceur-ws.org/Vol-3387/paper17.pdf>

References

1. Shubin, I. Snisar, S., Litvin, S. (2021). Formalization and Application of Algebraic Methods in Automated Intelligent Systems. *IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T)*. Kharkiv, Ukraine, P. 67–70. DOI: <http://doi.org/10.1109/PICST54195.2021.9772174>
2. Shubin, I., Karataiev, O. (2023). Reuse of Information Based on The Interpretation of Knowledge. *Innovative technologies and scientific solutions for industries*. No. 2 (24), P. 62–71. DOI: <http://doi.org/10.30837/itssi.2023.24.062>
3. Kozyriyev, A., Shubin, I. (2023). Method of Planning Data Processing Tasks in Distributed Systems with Limited Information About Available Resources. *Innovative technologies and scientific solutions for industries*. No. 3 (25), P. 27–39. DOI: <http://doi.org/10.30837/ITSSI.2023.25.027>
4. Sharonova, N., Doroshenko, A., Cherednichenko, O. (2021). Issues of Fact-based Information Analysis. *5th International Conference on Computational Linguistics and Intelligent Systems (COLINS-2021)*. CEUR Workshop Proceedings, Vol. 2136. URL: <https://ceur-ws.org/Vol-2136/10000011.pdf>
5. Karataiev, O. Shubin, I. (2023). Formal Model of Multi-Agent Architecture of a Software System Based on Knowledge Interpretation. *Radioelectronic and Computer Systems*. No. 4 (108), P. 53–64. DOI: <http://doi.org/10.32620/reks.2023.4.05>
6. Sarker, U., Deraman, A. B., Hasan, R. (2018). Descriptive Logic for Software Engineering Ontology: Aspect Software Quality Control. *4th International Conference on Computer and Information Sciences (ICCOINS)*. Kuala Lumpur, Malaysia, P. 1–5. DOI: <http://doi.org/10.1109/ICCOINS.2018.8510585>
7. Karataiev, O., Sitnikov, D., Sharonova, N. (2023). A Method for Investigating Links between Discrete Data Features in Knowledge Bases in the Form of Predicate Equations. *7th International Conference on Computational Linguistics and Intelligent Systems (COLINS-2023)*. CEUR Workshop Proceedings, Vol. 3387. URL: <https://ceur-ws.org/Vol-3387/paper17.pdf>