МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Харківський національний університет імені В. Н. Каразіна

Факультет математики і інформатики

Кафедра теоретичної та прикладної інформатики

Кваліфікаційна робота

бакалавр

на тему «Явище розбіжності гармонічного ряду: наближені та символьні обчислення».

Виконав: студент 4 курсу, групи МФ-41
спеціальність 122 «Комп'ютерні науки»
освітньо-професійна програма
«Інформатика»

Миннахметов Р. Т.

Керівник: доц. Гефтер С. Л.

Рецензент: проф. кафедри
фундаментальної математики
Ямпольський О. Л.

Харків – 2024 року

Науково-дослідницька робота

ВСТУП

У наші часи розповсюджуються думки про тотальну перевагу комп'ютерів та штучного інтелекту над людиною. Дуже часто можна чути про непотрібність вміння рахувати вручну через те, що калькулятор може порахувати набагато швидше людини, чи, наприклад, про те, що штучний інтелект може легко замінити людей різних професій, навіть творчих. Хоча такі думки не є безпричинними, їх автори забувають про те, що у людей також є переваги над комп'ютерами. Не слід забувати, що є речі, які людина може робити, а комп'ютери чи штучний інтелект не можуть. Наприклад, при прямому комп'ютерному обчисленні часткових сум елементів гармонічного ряду неможливо побачити, що ряд є розбіжним (див. розділ 2.3).

Гармонічний ряд – це дуже цікавий ряд, який при постійному зменшенні своїх елементів здається збіжним, але насправді є розбіжним. Його розбіжність була доведена ще у XIV столітті, коли ніяких комп'ютерів ще не існувало. Більш того, навіть за допомогою сучасного комп'ютера неможливо довести його розбіжність. Доведення розбіжності гармонічного ряду є суто теоретичним, тобто не полягає на експериментальні іспити. Але розбіжність не єдина особливість цього ряду.

Часткові суми гармонічного ряду, тобто скінченні суми перших n елементів ряду, із зростанням числа елементів зростають дуже повільно. Настільки повільно, що щоб часткова сума гармонічного ряду була хоча би наближена до 100, потрібно знайти суму більше 10^{40} елементів гармонічного ряду. Цей факт значно ускладнює використання комп'ютерів для безпосереднього обчислення його часткових сум. Крім того, на обчислення суми перших 10^{40} елементів ряду на звичайному комп’ютері знадобиться більше 10^{29} років, що більше ніж прогнозований термін життя Сонця (див. розділ 2.3).

Суперкомп’ютер, тобто комп’ютер з великим числом обчислювальних ядр, що дозволяють робити швидкі паралельні обчислення, не допоможе, бо щоб хоча би встигнути до вимирання Сонця, йому потрібно буде прискорити обчислення більше ніж у 10^{20} разів, для чого знадобиться стільки же обчислювальних ядр, що технічно не є реалістичним, бо на сьогоднішній день суперкомп’ютери мають лише кілька мільйонів ядр. Квантовий комп’ютер, тобто комп’ютер побудований на кубітах, тобто бітах, що знаходяться у стані суперпозиції між 0 та 1, теж не допоможе. По-перше, сьогоднішні квантові комп’ютери дуже ненадійні та вразливі до різних помилок через те, як вони побудовані. Наприклад, вони дуже чуткі до впливів навколишнього середовища. По-друге, операція додавання лише двох чисел на квантовому комп’ютері значно менш ефективна ніж на класичному комп’ютері. В цілому квантові комп’ютери мають перевагу над класичними комп’ютерами лише у тих задачах, де можна скоротити кількість операцій. Тобто швидкість квантового комп’ютера зростає лише за рахунок скорочення операцій. В нашому випадку є конкретна ціль отримати суму 10^{40} дробів шляхом прямого додавання, тому ніяких скорочень бути не може, що робить квантовий комп’ютер неактуальним для обчислення скінченних сум гармонічного ряду [1]. Крім цього існують системи символьних обчислень, такі як Maple, що проводять математичні операції над виразами, а не над числами. У роботі буде показано, що символьні обчислення не дозволять ефективно спрощувати часткові суми гармонічного ряду.

У цій роботі для комп'ютерного обчислення буде використано комп'ютер зі звичайними характеристиками (див. розділ 2.2). Обчислення буде проводитись на мові Python, тому що, по-перше, це найпопулярніша мова програмування для чисельних досліджень, а по-друге, після експерименту буде доведено, що незалежно від обраної мови, прямі комп'ютерні обчислення сум гармонічного ряду не є ефективними.

На щастя, є декілька теоретичних методів знаходження наближених часткових сум гармонічного ряду. Було виведено багато формул, що приблизно показують, чому дорівнюють суми, але не всі формули дають можливість просто оцінити похибку наближення. У цій роботі буде продемонстровано метод, що дозволяє досить легко оцінити похибку (див. розділ 2.4).

Хоча є багато публікацій на тему гармонічного ряду та теоретичного знаходження його часткових сум, немає комплексних публікацій, що досліджують динаміку теоретичних та комп'ютерних обчислень, а також надають кількісну оцінку таким видам обчислень.

Метою роботи є дослідження гармонічного ряду та різних методів обчислення його часткових сум.

Завданням роботи є:

1) Дослідити явище розбіжності гармонічного ряду.

2) Провести експеримент з прямим комп'ютерним обчисленням часткових сум гармонічного ряду з різною кількістю елементів та різною точністю.

3) Розглянути можливості символьних обчислень для спрощення часткових сум гармонічного ряду.

4) Дослідити теоретичні методи обчислення часткових сум гармонічного ряду.

5) Порівняти теоретичні та комп’ютерні методи обчислення.

ОСНОВНА ЧАСТИНА

РОЗДІЛ 2.1

Гармонічній ряд

В математиці можна зустріти суми нескінченної кількості доданків. Вони мають наступний вигляд: $a_1 + a_2 + a_3 + \cdots$.
Наприклад, розглянемо суму нескінченної геометричної прогресії.
Маємо рекурентне відношення $b_{n+1} = b_n q$. Якщо $|q| < 1$, то
$b_1 + b_2 + b_3 + \cdots = \frac{b_1}{1-q}$.
Тобто якщо $b_1 = \frac{1}{2}$ і $q = \frac{1}{2}$, то маємо наступну нескінченну суму:
$\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \cdots = \frac{\frac{1}{2}}{1-\frac{1}{2}} = 1$.
Цей вираз означає, що сума нескінченної кількості чисел може бути скінченною і дорівнювати конкретному числу.

Тепер наведемо кілька означень [2, глава 7].

Означення 1. Нехай $\{a_n\}$ – це послідовність дійсних чисел. Рядом будемо називати вираз $a_1 + a_2 + a_3 + \cdots + a_n + \cdots$.
Цей вираз ще не має чіткого сенсу, але поки що просто беремо його до уваги.

Означення 2. Числовій ряд $a_1 + a_2 + a_3 + \cdots + a_n + \cdots$ збігається, якщо існує скінченна границя послідовності скінченних сум, тобто $\exists \lim\limits_{n\to\infty}(a_1 + a_2 + a_3 + \cdots + a_n) \in R$. У свою чергу, числовий ряд, що не збігається, називають розбіжним та кажуть, що він розбігається.

Означення 3. Якщо числовий ряд $a_1 + a_2 + a_3 + \cdots + a_n + \cdots$ збігається, то сумою ряду є границя послідовності його скінченних сум, тобто

$$a_1 + a_2 + a_3 + \cdots + a_n + \cdots = \lim_{n\to\infty}(a_1 + a_2 + a_3 + \cdots + a_n). \qquad (2.1.1)$$

Рівність (2.1.1) можна переписати: $\sum_{k=1}^{\infty} a_k = \lim_{n\to\infty} \sum_{k=1}^{n} a_k$.

Означення 4. Число a_n має назву n-го загального члену ряду $a_1 + a_2 + a_3 + \cdots + a_n + \cdots$.

Означення 5. Число $S_n = a_1 + \ a_2 + \ a_3 + \cdots + a_n$ має назву n-ї часткової суми ряду $a_1 + a_2 + a_3 + \cdots + a_n + \cdots$.

Прийнявши ці означення до уваги, розглянемо деякі приклади рядів.

Приклад 1. Повернемося до ряду $\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \cdots$. Таким чином, $a_n = \frac{1}{2^n}$, $S_n = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \cdots + \frac{1}{2^n} = \frac{\frac{1}{2}\left(1 - \frac{1}{2^n}\right)}{1 - \frac{1}{2}}$. Оскільки $\lim_{n\to\infty} \frac{1}{2^n} = 0$, $\lim_{n\to\infty} S_n = 1$. Тобто ряд $\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \cdots$ збігається і $\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \cdots = 1$.

Приклад 2. Розглянемо ряд $1 + 1 + 1 + \cdots$. В цьому випадку $S_n = n$. Тобто $\lim_{n\to\infty} S_n = +\infty$, що свідчить про те, що ряд $1 + 1 + 1 + \cdots$ розбігається та $1 + 1 + 1 + \cdots = +\infty$.

Приклад 3. Розглянемо ряд $1 - 1 + 1 - 1 + \cdots$. Маємо $S_1 = 1$, $S_2 = 0$, $S_3 = 1$, $S_4 = 0$, … Очевидно, що послідовність $\{S_n\}$ не має границі, тобто ряд розбігається і не має конкретного значення.

Але можна знайти узагальнене значення цього ряду.

$S = 1 - 1 + 1 - 1 + \cdots = 1 - (1 - 1 + 1 - 1 + \cdots) = 1 - S$, тобто $S = \frac{1}{2}$.

Зазначимо, що таке ж значення можна отримати за допомогою формули для суми нескінченної геометричної прогресії:

$b_1 = 1, q = -1, 1 - 1 + 1 - 1 + \cdots$ "=" $\frac{1}{1-(-1)} = \frac{1}{2}$. Знак "=" означає, що праворуч виразу знаходиться узагальнене значення, тому що звісно цей ряд не дорівнює $\frac{1}{2}$ буквально. Узагальнені значення розбіжних числових рядів іноді застосовуються при обчисленнях.

В розглянутих прикладах збіжність ряду визначалась за допомогою точних обчислень послідовності часткових сум, але такі обчислення не завжди можна зробити. Наступний приклад покаже випадок дослідження збіжності ряду без точного обчислення часткових сум.

Приклад 4. Розглянемо ряд $1 + \frac{1}{2} + \frac{1}{3} + \cdots = \sum_{k=1}^{\infty} \frac{1}{k}$.

Він має назву «гармонічний ряд», оскільки кожний його доданок, починаючи з другого, є середнім гармонічним двох сусідніх:
якщо $c = \frac{1}{k},\ a = \frac{1}{k-1},\ b = \frac{1}{k+1}$, то $\frac{1}{c} = \frac{1}{2}\left(\frac{1}{a} + \frac{1}{b}\right)$.
Явище розбіжності гармонічного ряду у 1350 році відкрив французький математик Ніколя Орем [3, ст. 22-23]. Цікаво відзначити, що це відкриття ним було зроблено чисто теоретично (на відміну від багатьох математичних відкриттів, що спочатку були зроблені експериментально, тобто за допомогою безпосередніх обчислень). Вже пізніше було знайдено суму 1 000 000 доданків гармонічного ряду і отримано приблизне значення: $S_{1\,000\,000} \approx 14.392$.

Наведемо тепер міркування Орема . Будемо розглядати часткові суми гармонічного ряду вигляду S_{2^n}:

$S_2 = 1 + \frac{1}{2}$,

$S_4 = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} > 1 + \frac{1}{2} + \left(\frac{1}{4} + \frac{1}{4}\right) = 1 + \frac{1}{2} + \frac{1}{2} = 1 + \frac{2}{2}$,

$$S_8 = \left(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4}\right) + \left(\frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8}\right) > 1 + \frac{2}{2} + \left(\frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8}\right) =$$

$$= 1 + \frac{2}{2} + \frac{1}{2} = 1 + \frac{3}{2},$$

…

Неважко тепер перевірити за індукцією, що $S_{2^n} > 1 + \frac{n}{2}$. Таким чином, $S_{2^n} \to +\infty$. Оскільки послідовність S_n зростає, то маємо, що $S_n \to +\infty$, тобто $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots = +\infty$.

Зауважимо, що в нашому прикладі ми не можемо знайти послідовність часткових сум $\{S_n\}$ у «явному» вигляді. У розділі 2.4 дипломної роботи буде доведений наступний більш слабкий варіант цього «твердження».

Теорема. Не існує такої раціональної функції $R(x)$, що $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots + \frac{1}{n} = R(n)$ для всіх натуральних n.

РОЗДІЛ 2.2

Експериментальна частина

Опис експерименту:

Експеримент передбачає використання програмного коду для підтвердження чи спростування гіпотез про час обчислення скінченних сум гармонічного ряду та можливість експериментально побачити явище його розбіжності. В ході експерименту будуть зібрані чисельні дані, за допомогою яких можна буде зробити прогнози стосовно гіпотези, що перевіряється експериментом.

Мета експерименту:

Практична перевірка гіпотези про неможливість обчислення великих скінченних сум гармонічного ряду та дослідження його розбіжності лише за допомогою обчислювальної машини через занадто великий час виконання.

Необхідні передумови для відтворення експерименту:

Для відтворення експерименту потрібна обчислювальна машина здатна на виконання програмного коду на мові Python. У випадку цієї роботи було використано:

Комп’ютер із наступними характеристиками:

Процесор: Intel Core i5 5300U, 2.30GHz, Broadwell-U 14nm Technology
Оперативна пам’ять: 8.00GB Dual-Channel DDR3, 798MHz

Інші характеристики не мають значення для відтворення цього експерименту.

Програмне забезпечення:

Текстовий редактор за типом Visual Studio, PyCharm чи стандартного IDLE з інтерпретатором мови Python 3.9.11.

Додатковий модуль mpmath для мови Python призначений для обчислень з високою точністю [4].

Процедура:

Береться Код 1 (див. Додатки) та запускається на обчислювальній машині. Код 1 виводить наступні результати: результат обчислення за допомогою чисел з плаваючою точкою із стандартною точністю мови Python (10^{-16}) та час обчислення. Після отримання результатів у консолі, їх потрібно занести у Таблицю 1.

Наступним кроком береться Код 2 та запускається на обчислювальній машині. Код 2 виводить наступні результати: результат обчислення за допомогою дробів переведений у тип чисел з плаваючою точкою із стандартною точністю мови Python та час обчислення. Після отримання результатів у консолі, їх потрібно занести у Таблицю 2.

Далі береться Код 3 та запускається на обчислювальній машині. Код 3 виводить наступні результати: результат обчислення за допомогою чисел з плаваючою точкою з точністю 10^{-40} та час обчислення. Після отримання результатів у консолі, їх потрібно занести у Таблицю 3.

Після цього випадково вибирається 14 елементів гармонічного ряду. Для кожного обраного елемента береться вибірка із десяти обчислювальних повторень за допомогою Коду 4. Для кожного повторення час, який пішов на обчислення елемента, заноситься до таблиці. Після цього обчислюється середнє значення (медіана) за часом кожної таблиці.

Отримані дані:

Таблиця 1 (обчислення за допомогою чисел з плаваючою точкою)

Кількість елементів ряду	Час обчислення (секунди)	Отримана сума	Очікуване значення за наближеною формулою (див. розділ 2.4)
10^6	0.18	14.392726722864989	14.392726722865724
10^9	147.60	21.300481502348550	21.300481502347942
10^{10}	1867.14	23.603066594997500	23.603066594891990
10^{11}	19502.38	25.905651686536430	25.905651687841036
10^{12}	215027.26	28.208236780759341	28.208236780830582

Таблиця 2 (обчислення за допомогою дробів)

Кількість елементів ряду	Час обчислення (секунди)	Отримана сума	Очікуване значення за наближеною формулою (див. розділ 2.4)
10^6	1887.84	14.392726722865724	14.392726722865724

Таблиця 3 (обчислення за допомогою чисел з плаваючою точкою з точністю 10^{-40})

Кількість елементів ряду	Час обчислення (секунди)	Отримана сума
10^6	19.29	14.3927267228657236313811274931885876765564
10^7	305.35	16.6953113658598518153991189395404518840811
10^8	2633.37	18.9978964138538983244171103942239828423220
10^9	22814.33	21.3004815023479440166851018489083469705026
10^{10}	217824.29	23.6030665948919897007855933035927111737199

Таблиці часу обчислення випадковий елементів ряду

№	Час обчислення елементу 1/5 з точністю 10^{-40} (секунди)
1	0.0006507
2	0.0005878
3	0.0004619
4	0.0004836
5	0.0018685
6	0.0011598
7	0.0005666
8	0.0004835
9	0.0004536
10	0.0006979

Результат обчислення: 0.2000000000000000000000000000000000000000

Середній час обчислення: 0.0005772 секунд.

№	Час обчислення елементу 1/5783 з точністю 10^{-40} (секунди)
1	0.0004987
2	0.0006131
3	0.0005808
4	0.0005881
5	0.0005532
6	0.0005446
7	0.0006666
8	0.0005782
9	0.0006660
10	0.0005011

Результат обчислення: 0.0001729206294310911291717101850250734913

Середній час обчислення: 0.00058445 секунд.

№	Час обчислення елементу 1/5783934 з точністю 10^{-40} (секунди)
1	0.0004827
2	0.0004681
3	0.0005013
4	0.0006268
5	0.0004894
6	0.0010108
7	0.0005324
8	0.0005031
9	0.0006667
10	0.0005289

Результат обчислення: 0.0000001728927058987879183960259574192928

Середній час обчислення: 0.0005160 секунд.

№	Час обчислення елементу 1/5783934965 з точністю 10^{-40} (секунди)
1	0.0006857
2	0.0005574
3	0.0012643
4	0.0004853
5	0.0010221
6	0.0004786
7	0.0007766
8	0.0005630
9	0.0006218
10	0.0005621

Результат обчислення: 0.0000000001728926770531210493996278881051

Середній час обчислення: 0.00056255 секунд.

№	Час обчислення елементу 1/5783934965892 з точністю 10^{-40} (секунди)
1	0.0004698
2	0.0007446
3	0.0004985
4	0.0005940
5	0.0007086
6	0.0008552
7	0.0003848
8	0.0005460
9	0.0005757
10	0.0004960

Результат обчислення: 0.000000000000172892677026457494425233378З

Середній час обчислення: 0.00056085 секунд.

№	Час обчислення елементу 1/5783934965892795 з точністю 10^{-40} (секунди)
1	0.0006246
2	0.0007831
3	0.0005136
4	0.0005340
5	0.0005692
6	0.0005110
7	0.0007175
8	0.0004678
9	0.0005493
10	0.0005589

Результат обчислення: 0.0000000000000001728926770264337303824067

Середній час обчислення: 0.0005541 секунд.

№	Час обчислення елементу 1/5783934965892795837 з точністю 10^{-40} (секунди)
1	0.0005355
2	0.0005914
3	0.0005276
4	0.0014092
5	0.0005141
6	0.0005612
7	0.0005490

8	0.0005112
9	0.0007543
10	0.0006079

Результат обчислення: 0.0000000000000000001728926770264337053629

Середній час обчислення: 0.0005551 секунд.

№	Час обчислення елементу 1/5783934965892795837639 з точністю 10^{-40} (секунди)
1	0.0006264
2	0.0005528
3	0.0005629
4	0.0007175
5	0.0006428
6	0.0005227
7	0.0006185
8	0.0004901
9	0.0005323
10	0.0005674

Результат обчислення: 0.0000000000000000000001728926770264337053

Середній час обчислення: 0.00056515 секунд.

№	Час обчислення елементу 1/5783934965892795837639418 з точністю 10^{-40} (секунди)
1	0.0008265
2	0.0007246
3	0.0005791
4	0.0008661

5	0.0005310
6	0.0005423
7	0.0005310
8	0.0005290
9	0.0005984
10	0.0005380

Результат обчислення: 0.00000000000000000000001728926770264337

Середній час обчислення: 0.0005607 секунд.

№	Час обчислення елементу 1/5783934965892795837639418716 з точністю 10^{-40} (секунди)
1	0.0005372
2	0.0008517
3	0.0005665
4	0.0009529
5	0.0005624
6	0.0005341
7	0.0005906
8	0.0009513
9	0.0005576
10	0.0015969

Результат обчислення: 0.00000000000000000000000001728926770264

Середній час обчислення: 0.0005786 секунд.

№	Час обчислення елементу 1/5783934965892795837639418716861 з точністю 10^{-40} (секунди)
1	0.0005529

2	0.0006709
3	0.0005271
4	0.0010785
5	0.0005660
6	0.0005463
7	0.0004615
8	0.0009245
9	0.0011305
10	0.0009060

Результат обчислення: 0.0000000000000000000000000000001728926770

Середній час обчислення: 0.0006185 секунд.

№	Час обчислення елементу 1/5783934965892795837639418716861243 з точністю 10^{-40} (секунди)
1	0.0005875
2	0.0006221
3	0.0006065
4	0.0005809
5	0.0007175
6	0.0005154
7	0.0007955
8	0.0006455
9	0.0006055
10	0.0006959

Результат обчислення: 0.0000000000000000000000000000000001728927

Середній час обчислення: 0.0006143 секунд.

№	Час обчислення елементу 1/5783934965892795837639418716861243758 з точністю 10^{-40} (секунди)
1	0.0005169
2	0.0006167
3	0.0005177
4	0.0005044
5	0.0006270
6	0.0008372
7	0.0005122
8	0.0005462
9	0.0006149
10	0.0005175

Результат обчислення: 0.0000000000000000000000000000000000001729

Середній час обчислення: 0.00053195 секунд.

№	Час обчислення елементу 1/5783934965892795837639418716861243758214 з точністю 10^{-40} (секунди)
1	0.0005506
2	0.0006160
3	0.0006138
4	0.0007127
5	0.0006862
6	0.0006483
7	0.0011636
8	0.0005815

9	0.0005688
10	0.0005351

Результат обчислення: 0.0000000000000000000000000000000000000001

Середній час обчислення: 0.0006149 секунд.

РОЗДІЛ 2.3

Теоретичне обґрунтування результатів експерименту

Обчислення суми гармонічного ряду за допомогою чисел з плаваючою точкою дають точні та відносно швидкі результати на кількості доданків до 10^{10}. Після 10^{10} доданків доводиться розраховувати на час обчислення більше години. З Таблиці 1 видно, що час обчислень зростає в геометричній прогресії для наступної кількості доданків. За такими прогнозами, виходить, що обчислення суми гармонічного ряду для 10^{13} доданків займе понад двадцять два дні.

Обчислення за допомогою дробів хоч і дають точний результат, але арифметичні операції займають значно більше часу за рахунок роботи із загальними дільниками. Значна різниця в ефективності порівняно з обчисленнями за допомогою чисел з плаваючою точкою робить цю стратегію обчислення неоптимальною.

Нижче за допомогою методів математичного аналізу буде показано, що для отримання «великого» значення часткової суми (приблизно 100) необхідно буде знайти суму не менше ніж 10^{40} доданків гармонічного ряду. Для того, щоб усі ці доданки вносили відповідний внесок у суму, їх необхідно буде знаходити з точністю не менше ніж 10^{-40}.

З Таблиці 3 видно, шо надточні обчислення швидко стають витратними. Вже на 10^6 доданків швидкість обчислення з точністю 10^{-40} в порівнянні зі стандартною точністю падає в понад сто разів, а часова відмітка в два дні досягається вже при 10^{10} доданках, на відміну від 10^{12} доданків зі звичайним обчисленням. Спостерігається таке саме відношення часів обчислень, що у Таблиці 1 – геометрична прогресія. З певного моменту часи обчислень стабільно відрізняються один від одного приблизно у 8.6 разів. За такими прогнозами виходить, що щоб розрахувати суму 10^{12} доданків з точністю 10^{-40}, знадобиться близько 14817016 секунд, що еквівалентно 5.7 місяцям – майже половині року.

На Графіку 1 (див. Додатки) горизонтальна вісь відповідає за десяткові діапазони знаменників у науковій нотації при обчисленні окремих значень елементів гармонічного ряду. Тобто $\frac{1}{5}$ на цьому графіку знаходиться в діапазоні $1 \cdot 10^0$, тому що п'ятірка в науковій нотації представляється як $5 \cdot 10^0$. $\frac{1}{5783}$ на графіку знаходиться в діапазоні $1 \cdot 10^3$, тому що 5783 в науковій нотації представляється як $5.783 \cdot 10^3$. Вертикальна вісь відповідає за значення середнього часу обчислення в секундах.

З Графіка 1 видно, що зростання часу обчислення елемента гармонічного ряду зі збільшенням знаменника перебуває у межах похибки і не є стабільним. Таким чином, можна зробити висновок, що всі доданки часткової суми гармонічного ряду обчислюються приблизно з однаковим часом.

Виходить, що кожен доданок часткової суми гармонічного ряду обчислюється приблизно за 0.0005 секунд. Припустимо, що на потужному комп'ютері доданки обчислювалися б приблизно за 0.0001 секунди. Таким чином, на обчислення одного доданку знадобилося би $1 \cdot 10^{-4}$ секунди, а на 10^{40} доданків $1 \cdot 10^{-4} \cdot 10^{40} = 1 \cdot 10^{36}$ секунди. Це значення еквівалентне

$(1 \cdot 10^{36})/3600/24 = 1.157 \cdot 10^{31}$ дням. А це значення еквівалентне $1.157 \cdot 10^{31}/365 = 3.171 \cdot 10^{29}$ рокам.

Тобто на одне лише обчислення всіх 10^{40} доданків часткової суми гармонічного ряду знадобиться близько $3 \cdot 10^{29}$ років. Якщо враховувати додавання цих чисел, то знадобиться ще більше часу. Однак отриманого часу вже вистачає, щоб зробити висновок, що це обчислення неможливе. Вчені прогнозують, що вже через 5 мільярдів $(5 \cdot 10^{9})$ років Сонце перетвориться у білого карлика, тобто вичерпає усю свою енергію, що означає фактичну смерть зірки. Але і до цього моменту Сонце встигне поглинути навколишні планети сонячної системи, в тому числі і Землю [5].

Покажемо тепер, чому у наших обчисленнях нам приходиться мати справу з такими катастрофічно великими числами. Ми показали, що послідовність часткових сум гармонічного ряду необмежено зростає (див. розділ 2.1). Для подальшого дослідження швидкості зростання послідовності $S_n = 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{n}$, і, зокрема, для пояснення результатів нашого комп'ютерного експерименту, необхідно встановити важливі зв'язки між деякими властивостями часткових сум числових рядів та властивостями відповідних інтегралів.

Припустимо, що функція $f(x)$ визначена, додатна, неперервна на проміжку $[1, +\infty)$, та спадає на цьому проміжку. Отримаємо подвійну нерівність для суми значень цієї функції у перших n натуральних точках (див. Рис. 2.3.1).

$$f(1) + f(2) + f(3) + \cdots + f(n) \qquad (2.3.1)$$

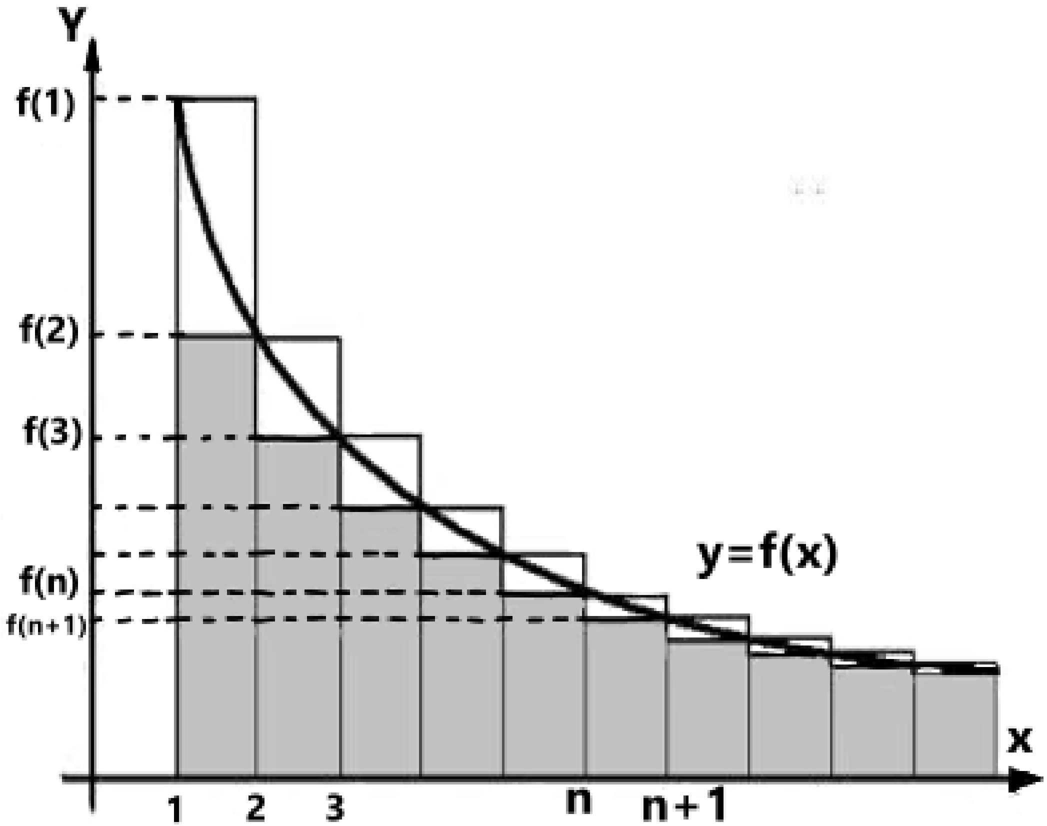


Рис. 2.3.1. Графік функції f(x).

Для цього відмітимо, що кожен доданок суми (2.3.1) є площею прямокутника основи якого дорівнюють 1, а висоти дорівнюють значенням функції у відповідних точках (див. Рис. 2.3.1). При цьому сума площ цих прямокутників, тобто сума (2.3.1) більша площі криволінійної трапеції $S_{[1,n+1]}(f)$, що відповідає графіку функції $f(x)$ на проміжку $[1, n+1]$. Тобто $f(1)+f(2)+f(3)+\cdots+f(n) \geq S_{[1,n+1]}(f)$. З іншого боку, площа цієї криволінійної трапеції більше за суму площ маленьких прямокутників, у яких основи також дорівнюють одиниці, а висоти дорівнюють $f(2), f(3), \dots, f(n+1)$. Тому $f(2)+f(3)+\cdots+f(n+1) \leq S_{[1,n+1]}(f)$. Оскільки площа криволінійної трапеції, що відповідає функції $f(x)$ на проміжку $[a, b]$, дорівнює $\int_a^b f(x)dx$, то

$$f(1)+f(2)+f(3)+\cdots+f(n) \geq \int_1^{n+1} f(x)dx \text{ і} \qquad (2.3.2)$$

$$f(2)+f(3)+\cdots+f(n+1) \leq \int_1^{n+1} f(x)dx. \qquad (2.3.3)$$

Якщо тепер $f(x)=\frac{1}{x}$, то за формулою Ньютона-Лейбніца маємо

$\int_1^{n+1}\frac{dx}{x} = \ln(\mathrm{n}+1)-\ln(1)=\ln(\mathrm{n}+1)$. Тому $1+\frac{1}{2}+\frac{1}{3}+\ldots+\frac{1}{n} \geq \ln(n+1)$, і $\frac{1}{2}+\frac{1}{3}+\ldots+\frac{1}{n}+\frac{1}{n+1} \leq \ln(\mathrm{n}+1)$, або

$$1+\frac{1}{2}+\frac{1}{3}+\ldots+\frac{1}{n}+\frac{1}{n+1} \leq 1+\ln(n+1). \qquad (2.3.4)$$

Нерівність (2.3.4) пояснює чому немає практичної можливості побачити явище нескінченності суми гармонічного ряду при безпосередньому знаходженні його часткових сум. Наприклад, якщо ми бажаємо щоб виконувалась нерівність

$1+\frac{1}{2}+\frac{1}{3}+\ldots+\frac{1}{n} \geq 101$, то з цього буде випливати нерівність $\ln(n+1) > 100$, тобто $n+1>e^{100}$, або $n \geq e^{100}$. Число e^{100} є дуже великим. Оскільки $\lg(e) \approx 0{,}43$, то $\lg(e^{100})=100\cdot\lg(e) \approx 43 > 40$. Тому $e^{100} > 10^{40}$. Знайти на звичайному комп'ютері суму 10^{40} доданків гармонічного ряду практично неможливо. Справа не тільки у тому, що катастрофічно збільшується кількість доданків. Для того, щоб усі ці доданки вносили відповідний вклад у суму, їх необхідно буде знаходити з великою точністю, на що також буде витрачатися дуже багато часу (див. міркування на початку розділу 2.3).

РОЗДІЛ 2.4

Наближене обчислення часткових сум гармонічного ряду за допомогою теорії

Покажемо, що існує число γ, для якого виконується така рівність:
$\lim_{n\to\infty}\left(1+\frac{1}{2}+\frac{1}{3}+\cdots+\frac{1}{\mathrm{n}}-\ln(\mathrm{n}+1)\right)= \gamma.$

Якщо ми подивимося на Рис. 2.3.1, то ми побачимо, що часткова сума гармонічного ряду, це сума площ вертикальних прямокутників. Ця сума складається з площі області під кривою $\frac{1}{x}$, де 1≤ x ≤ n+1, тобто $\int_1^{n+1}\frac{1}{x}dx=\ln(n+1)$, та суми площ шматочків білих прямокутників над кривою $\frac{1}{n}$. Легко бачити, що γ – це і є сума площ шматочків над кривою $\frac{1}{n}$. Існування числа γ стає очевидним, якщо ми звернемо увагу на суму площ маленьких білих прямокутників над сірими прямокутниками. Площа k-го білого прямокутника рахується різницею $f(k)-f(k+1)$, тому що від площі усього прямокутника віднімається площа сірого прямокутника. Беручи це до уваги, ми можемо порахувати суму площ n білих прямокутників: $Sn = f(1)-f(2)+f(2)-f(3)+f(3)+\cdots+f(n)-f(n+1)=f(1)-f(n+1).$ Тобто якщо сума площ білих прямокутників – це скінченне число, то і сума площ шматочків білих прямокутників над кривою $\frac{1}{n}$ – теж скінченне число. Таким чином, $\lim_{n\to\infty}\left(1+\frac{1}{2}+\frac{1}{3}+\cdots+\frac{1}{\mathrm{n}}-\ln(\mathrm{n}+1)\right)$ – це скінченна границя, тобто число γ існує, що і потрібно було довести. Це число називається константою Ейлера-Маскероні.

Тепер можна обчислити γ. Видно, що у формі, у якій число γ було визначено, його обчислити неможливо, бо вона містить у собі саму часткову суму гармонічного ряду, яку ми і глобально намагаємося обчислити. Тому треба привести число γ у інший вигляд.

Покажемо, що $\sum_{n=1}^{\infty}\left(\frac{1}{n}-\ln\left(1+\frac{1}{n}\right)\right)=\gamma$

$$\sum_{n=1}^{\infty}\left(\frac{1}{n}-\ln\left(1+\frac{1}{n}\right)\right)= \lim_{n\to\infty}\left(1-\ln(2)+\frac{1}{2}-\ln\left(1+\frac{1}{2}\right)+\frac{1}{3}-\ln\left(1+\frac{1}{3}\right)+\cdots+\frac{1}{n}-\ln\left(1+\frac{1}{n}\right)\right)=\lim_{n\to\infty}\left(1-\ln(2)+\frac{1}{2}-\ln\left(\frac{3}{2}\right)+\frac{1}{3}-\ln\left(\frac{4}{3}\right)+\cdots+\frac{1}{n}-\ln\left(\frac{n+1}{n}\right)\right)=\lim_{n\to\infty}\left(1-\ln(2)+\frac{1}{2}-(\ln(3)-\ln(2))+\frac{1}{3}-(\ln(4)-\ln(3))+\cdots+\frac{1}{n}-(\ln(n+1)-\ln(n))\right)=\lim_{n\to\infty}\left(1+\frac{1}{2}+\frac{1}{3}+\cdots+\frac{1}{n}-\ln(n+1)\right)$$

Отриманий вираз є границею різниці суми елементів виду $\frac{1}{n}$ та $\ln(n+1)$, що, за визначенням, дорівнює γ. Що і потрібно було показати.

Обчислення γ за допомогою комп'ютера з точністю 10^{-7}, тобто для $n=10^7$, дає результат 0.57721561490132. Для порівняння, більш точне значення γ виглядає як 0.57721566490153. Звернемо увагу на те, що отримане значення є наближеним, бо нескінченну суму було перетворено на скінченну. Тому залишилося знайти похибку цього наближення.

Для початку зробимо декілька допоміжних верхніх та нижніх оцінок. Візьмемо функцію $f(x)=\ln(1+x)$. Дотична до цієї функції у точці 0 є пряма $y=x$. Беручи це до уваги, при $x\in(-1;\ +\infty), x\ \neq 0$ маємо нерівність

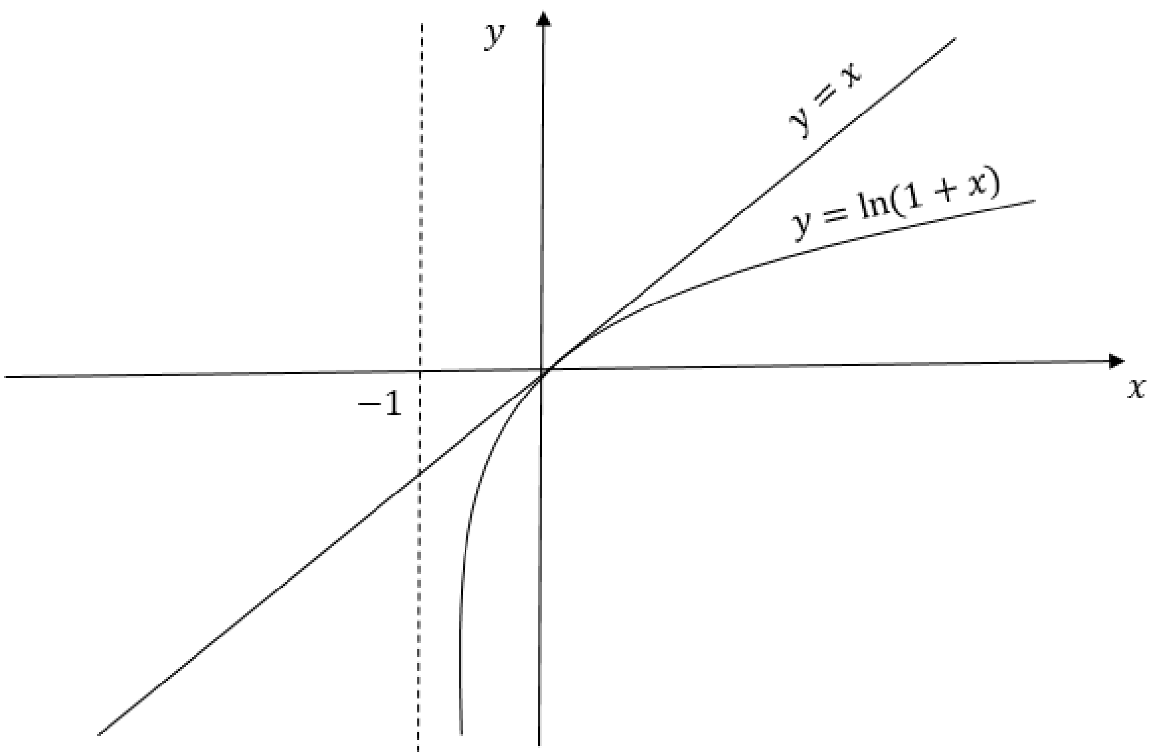
$\ln(1+x) < x$ (див. графік нижче). Якщо замість x підставимо $\frac{1}{n}$, отримаємо нерівність $\ln\left(1+\frac{1}{n}\right) < \frac{1}{n}$. Покажемо, що $\ln\left(1+\frac{1}{n}\right) > \frac{1}{n+1}$.

$\ln\left(1+\frac{1}{n}\right) = \ln\left(\frac{n+1}{n}\right) = -\ln\left(\frac{n}{n+1}\right) = -\ln\left(1-\frac{1}{n+1}\right) > \frac{1}{n+1}$, оскільки

$\ln\left(1-\frac{1}{n+1}\right) < -\frac{1}{n+1}$.

Отже, $\frac{1}{n+1} < \ln\left(1+\frac{1}{n}\right) < \frac{1}{n}$.

Тому $\frac{1}{n} - \ln\left(1+\frac{1}{n}\right) < \frac{1}{n} - \frac{1}{n+1}$, тобто $0 < \frac{1}{n} - \ln\left(1+\frac{1}{n}\right) < \frac{1}{n(n+1)}$.



Тепер перейдемо до оцінки похибки приблизного обчислення γ.

$\left|\gamma - \sum_{n=1}^{N}\left(\frac{1}{n} - \ln\left(1+\frac{1}{n}\right)\right)\right| = \left|\left(1-\ln(2)+\frac{1}{2}-\ln\left(1+\frac{1}{2}\right)+\frac{1}{3}-\ln\left(1+\frac{1}{3}\right)+\cdots\right)-\left(1-\ln(2)+\frac{1}{2}-\ln\left(1+\frac{1}{2}\right)+\frac{1}{3}-\ln\left(1+\frac{1}{3}\right)+\cdots+\frac{1}{N}-\ln\left(1+\frac{1}{n}\right)\right)\right| =$

$\frac{1}{N+1} - \ln\left(1 + \frac{1}{N+1}\right) + \frac{1}{N+2} - \ln\left(1 + \frac{1}{N+2}\right) + \cdots < \frac{1}{N+1} - \frac{1}{N+2} + \frac{1}{N+2} - \frac{1}{N+3} + \cdots = \frac{1}{N+1}$.

Таким чином, похибка приближення константи γ не перевищує $\frac{1}{n+1}$.

Маючи константу Ейлера-Маскероні, ми можемо приблизно обчислювати часткові суми гармонічного ряду n елементів за формулою $\sum_{k=1}^{n} \frac{1}{k} \approx \gamma + \ln(n+1)$. Зрозуміло, що навіть з невеликою точністю γ, дуже легко обчислювати часткові суми гармонічного ряду навіть для дуже великих n, таких як, наприклад, 10^{40}. Формула надає результат 92.680619334663150.

Порівняємо результати отримані за допомогою наближеної формули з результатами отриманими за допомогою комп’ютерних обчислень.

Кількість елементів ряду	Комп’ютерні обчислення за формулою	Комп’ютерні обчислення з розділу 2.2
10^6	14.392726172865594	14.392726722864989
10^9	21.300481451847730	21.300481502348550
10^{10}	23.603066544841777	23.603066594997500
10^{11}	25.905651637835824	25.905651686536430
10^{12}	28.208236730829867	28.208236780759341

Результати досить непогані, особливо зважаючи на те, що обчислення за допомогою наближеної формули проходять моментально. Фактично точність обчислення часткової суми залежить від точності обчислення константи Ейлера-Маскероні.

РОЗДІЛ 2.5

Символьні обчислення та неможливість спрощення часткових сум гармонічного ряду

У математиці є два головних види обчислень: чисельні та символьні. Чисельні обчислення працюють на приблизних та, якщо це можливо, точних чисельних значеннях, таких як 0.25, 2.236 та 3.14. Символьні обчислення, у свою чергу, працюють на математичних виразах, таких як $\frac{1}{4}$, $\sqrt{5}$ та π.

Перевага символьних обчислень перед чисельними у тому, що у проміжних обчисленнях не втрачається точність. Деякі раціональні та усі ірраціональні числа неможливо точно представити у чисельному виді. Наприклад, число $\frac{1}{3}$ у чисельному виді можна представити настільки точно, наскільки багато «трійок» ми поставимо у числі (0.33333), але таке число ніколи точно не буде дорівнювати $\frac{1}{3}$, за рахунок чого і втрачається точність при чисельних обчисленнях.

Для кожного з видів обчислень є різні комп’ютерні алгоритми розв’язування математичних виразів.

Звичайний калькулятор, наприклад, сканує математичні вирази на те, щоб визначитися із порядком операцій, а потім для кожного підвиразу використовує алгоритми перетворення виразів на числа, щоб у правильному порядку над цими числами провести відповідні операції, за рахунок чого і отримується чисельна відповідь.

Системи символьних обчислень роблять майже те ж саме, але не роблять конвертацію підвиразів у числа. Вони роблять математичні операції над самими підвиразами.

Тобто, наприклад, якщо звичайному калькулятору задати вираз $\sqrt{2} \cdot \sqrt{3}$, він визначить, що над двома елементами $\sqrt{2}$ та $\sqrt{3}$ треба провести операцію множення, потім переведе кожен з елементів у числа, і на виході покаже множення цих двох чисел. На виході ми отримаємо приблизно 2.449489743. Такий результат ми отримаємо навіть якщо ми самостійно переведемо кожен з елементів у числа та потім самі їх помножимо, що підтверджує те, що калькулятор просто конвертує кожний підвираз у число.

Якщо задати такий вираз системі символьного обчислення, то вона теж визначить підвирази, порядок операцій, та проведе усі операції над самими підвиразами, а не над їхніми чисельними значеннями. Наприклад, система Maple покаже $\sqrt{6}$ (див Рис. 2.5.1).

$$simplify(\sqrt{2} \cdot \sqrt{3}) = \sqrt{6}$$

Рис. 2.5.1 Обчислення у системі Maple.

На цьому можливості систем символьного обчислення не закінчуються. Алгоритми вбудовані у ці систему також дозволяють їм «згортати» суми числових рядів, тобто перетворювати їх з неявного вигляду на явний. Нижче надано кілька прикладів обчислень у системі Maple.

Maple:

$$\sum_{i=1}^{n}\left(\frac{1}{i\cdot(i+1)}\right) = -\frac{1}{n+1}+1$$

1)

$$\sum_{i=0}^{n-1}(x^i) = \frac{x^n}{x-1}-\frac{1}{x-1}$$

2)

$$\sum_{i=1}^{n}(i) = \frac{(n+1)^2}{2}-\frac{n}{2}-\frac{1}{2}$$

3)

$$\sum_{i=1}^{n}\left(\frac{(2\cdot i^2-1)}{(2\cdot i-1)\cdot(2\cdot i+1)}\right) = \frac{n}{2}-\frac{1}{4}+\frac{1}{8\left(n+\frac{1}{2}\right)}$$

4)

$$\sum_{i=1}^{n}\left(\frac{(i\cdot(i+3))}{(i+1)\cdot(i+2)}\right) = n-1+\frac{2}{n+2}$$

5)

Зверніть увагу, що «згортка» сум числових рядів має дуже нестандартний вигляд. Це лише підтверджує факт того, що системи виводять перетворення самостійно. Наприклад, сума чисел від 1 до n, що відомо «розгортається» у $\frac{n\cdot(n+1)}{2}$, Maple виразив як $\frac{(n+1)^2}{2}-\frac{n}{2}-\frac{1}{2}$, що, звичайно, дорівнює виразу $\frac{n\cdot(n+1)}{2}$, але цей вираз явно був виведений якимось алгоритмом, а не скопійований з бази знань системи, яка, до речі, теж існує, і це буде показано далі. Справа у тому, що деякі вирази не «згортаються» так, як вирази з прикладу. Сума гармонічного ряду – один з таких виразів. Якщо системі Maple задати вираз суми гармонічного ряду, то вона виведе вже знайому з розділу 2.4 наближену формулу у іншому вигляді (див Рис. 2.5.2 нижче).

$$\sum_{i=1}^{n}\left(\frac{1}{i}\right) = \Psi(n+1) + \gamma$$

Рис. 2.5.2 Згортка часткової суми гармонічного ряду у системі Maple.

Функція Ψ – це дігамма-функція, що розгортається як $\Psi(x) = \frac{\Gamma'(x)}{\Gamma(x)}$, де $\Gamma(x)$ – гамма функція. Гамма функція для усіх цілих невід’ємних k має наступний вигляд: $\Gamma(k+1) = k!$ Похідна гамма функції для усіх цілих невід’ємних k має наступний вигляд: $\Gamma'(k+1) = k!\left(-\gamma + \sum_{l=1}^{k}\frac{1}{l}\right)$ [6].

З цього виходить, що $\Psi(n+1) = \frac{\Gamma'(n+1)}{\Gamma(n+1)} = \frac{n!\left(-\gamma+\sum_{l=1}^{n}\frac{1}{l}\right)}{n!} = -\gamma + \sum_{l=1}^{n}\frac{1}{l}$. Якщо до цього виразу додати γ, то ми і отримаємо означення часткової суми гармонічного ряду: $\sum_{l=1}^{n}\frac{1}{l} = \gamma + \Psi(n+1)$ для всіх натуральних n. Тобто якийсь алгоритм розпізнає у заданому виразі суму гармонічного ряду, та з бази знань бере формулу для наближеного обчислення цього виразу через дігамма функцію.

Хоча система Maple і може «згортати» часткові суми, але ми не знаємо як вона це робить. Справа у тому, що Maple – це ліцензійний продукт, внутрішні алгоритми обчислення якого є комерційною таємницею. Також теоретично може статися так, що система символьного обчислення не зможе «згорнути» суму, а людина зможе. Тому важливо вміти перевіряти роботу людини за допомогою комп’ютера. Маючи рекурентне співвідношення часткової суми та «згорнутий» вигляд часткової суми, можна легко перевірити правильність «згортки».

Наприклад, візьмемо часткову суму $\frac{1}{1\cdot 2}+\frac{1}{2\cdot 3}+\frac{1}{3\cdot 4}+\cdots+\frac{1}{n(n+1)}$. Вона має рекурентне співвідношення $S_{n+1}=S_n+\frac{1}{(n+1)(n+2)}$, де S_n – часткова сума перших n елементів деякого числового ряду.
Припустимо, що задана часткова сума «згортається» у $\frac{n}{n+1}$. Цю «згортку» можна легко перевірити. Виходячи з неї, часткова сума $n+1$ елементів буде виглядати як $\frac{n+1}{n+2}$. Виходячи з рекурентного співвідношення, часткова сума $n+1$ елементів буде виглядати як $\frac{n}{n+1}+\frac{1}{(n+1)(n+2)}$. Склавши дробові доданки, ми як раз і отримаємо $\frac{n+1}{n+2}$, що свідчить, що «згортка» була підібрана правильно. Такий же результат нам надає система Maple (див. Рис. 2.5.3 нижче).

$$\mathit{verify}\left(\frac{(n+1)}{n+2}, \frac{n}{n+1}+\frac{1}{(n+1)\cdot(n+2)}, \mathit{equal}\right) = \mathit{true}$$

Рис. 2.5.3 Перевірка «згортки» часткової суми у системі Maple.

Таким же чином можна перевіряти «згортки» інших часткових сум. Та, що більш важливо, ми можемо перевіряти «згортки» надані самою системою Maple.

За допомогою символьних обчислень неможливо показати, що для послідовності $S_n = 1+\frac{1}{2}+\frac{1}{3}+\cdots+\frac{1}{n}$ немає простої явної формули, тобто її неможливо «згорнути». Відмітимо також, що гамма-функція не є елементарною функцією, але з цього не можна зробити висновок про неможливість точного представлення часткових сум гармонічного ряду через елементарні функції. При цьому нам не вдалося знайти у літературі підтвердження цього твердження. Використовуючи константу Ейлера-Маскероні ми доведемо більш простий факт.

Покажемо, що для послідовності $S_n = 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{n}$ часткових сум гармонічного ряду не існує явної формули вигляду $1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{n} = R(n)$, де $R(x)$ – раціональна функція. Для доведення цього факту наведемо декілька відомих допоміжних тверджень [2, глава 2].

Лема 1.

$\ln(n) = o(n)$, тобто послідовність $\ln(n)$ є нескінченно малою у порівнянні з послідовністю натуральних чисел.

Лема 2.

Якщо $R(x)$ – раціональна функція і $R(n) \to +\infty$, то $\ln(n) = o\big(R(n)\big)$.

Доведемо основну теорему цього розділу.

Теорема.

Не існує такої раціональної функції $R(x)$, що $1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{n} = R(n)$ для всіх $n = 1, 2, 3, \ldots$

Доведення:

Припустимо, що $1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} = R(n)$ для деякої раціональної функції та для всіх $n \in \mathbb{N}$.

Тоді $R(n) \neq 0,\ R(n) \to +\infty$ і $\lim\limits_{n\to\infty}\left(1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} - \ln(n)\right) =$

$\lim\limits_{n\to\infty}(R(n) - \ln(n)) = \lim\limits_{n\to\infty}\left(R(n)\left(1 - \frac{\ln(n)}{R(n)}\right)\right) = +\infty$, згідно з лемою 2.

Але, $\lim\limits_{n\to\infty}\left(1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{n} - \ln(n)\right) = \gamma$, де γ – стала Ейлера-Маскероні.

Отримали суперечність, яка й доводить, що твердження теореми є правильним.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. J. Chen, Review on Quantum Communication and Quantum Computation, 2021. [Електронний ресурс]. Режим доступу: https://iopscience.iop.org/article/10.1088/1742-6596/1865/2/022008/pdf

2. А. Я. Дороговцев, Математичний аналіз, частина 1, 1993.

3. J. Havil, Gamma: Exploring Euler's constant, 2003.

4. F. Johansson, mpmath. Офіційна документація модуля mpmath. [Електронний ресурс]. Режим доступу: https://mpmath.org

5. NASA, Our Sun: Facts. [Електронний ресурс]. Режим доступу: https://science.nasa.gov/sun/facts

6. M. Haiman, Notes on Gamma and Zeta, 2014. [Електронний ресурс]. Режим доступу: https://math.berkeley.edu/~mhaiman/math185-summer14/gamma-notes.pdf

ДОДАТКИ

Код 1

```
import time
import datetime

def harmonic_sum(num: int):
    for i in range(1, num + 1):
        yield 1/i

start = time.perf_counter()

n = #оберіть своє значення кількості елементів ряду

print(f"The program started at: {datetime.datetime.now()}")

print(sum(harmonic_sum(n)))

end = time.perf_counter()
print(f"The program finished at: {datetime.datetime.now()}")
print(f"The program took {end - start} seconds.")
```

Код 2

```
from fractions import Fraction
import time
import datetime

def harmonic_sum(num: int):
    for i in range(1, num + 1):
        yield Fraction(1, i)

start = time.perf_counter()

n = #оберіть своє значення кількості елементів ряду

print(f"The program started at: {datetime.datetime.now()}")

print(float(sum(harmonic_sum(n))))

end = time.perf_counter()
print(f"The program finished at: {datetime.datetime.now()}")
print(f"The program took {end - start} seconds.")
```

Код 3

```
import time
import datetime
from mpmath import mpf, workdps

def harmonic_sum(num: int):
    for i in range(1, num + 1):
        yield mpf(1)/i

start = time.perf_counter()

n = #оберіть своє значення кількості елементів ряду

print(f”The program started at: {datetime.datetime.now()}”)

with workdps(40):
    print(repr(sum(harmonic_sum(n))))

end = time.perf_counter()
print(f”The program finished at: {datetime.datetime.now()}”)
print(f”The program took {end - start} seconds.”)
```

Код 4

```
import time
import datetime
from mpmath import mpf, workdps, nprint

start = time.perf_counter()

precision = 40

n = #оберіть своє значення знаменника дробу

print(f"The program started at: {datetime.datetime.now()}")

with workdps(precision):
    res = 1/mpf(n)
    nprint(res, 120)

end = time.perf_counter()
print(f"The program finished at: {datetime.datetime.now()}")
print(f"The program took {end - start} seconds.")
```

Графік 1 (середній час обчислення конкретних елементів гармонічного ряду з точністю 10^{-40})