

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра прикладної математики

Звіт

із лабораторної роботи

з дисципліни «АЛГОРИТМИ І СИСТЕМИ КОМП'ЮТЕРНОЇ
МАТЕМАТИКИ 1.МАТЕМАТИЧНІ АЛГОРИТМИ»

на тему

“Чисельне диференціювання та інтегрування”

Виконала:

студентка групи КМ-01

Резниченко Є. С.

Перевірила:

Асистент кафедри ПМА

Ковальчук-Химюк Л. О.

Зміст

Вступ.....	3
Основна частина	4
Варіант 1	4
Опис програми.....	5
Висновки	6
Відповіді на контрольні питання	13
Перелік посилань.....	14
Додаток А – Скріншоти роботи програми.....	15
Додаток В – Код програми	19

Вступ

Метою роботи є ознайомитися з програмними засобами чисельного диференціювання й інтегрування функцій; практичне розв'язання задач з використанням СКМ, порівняльний аналіз методів чисельного диференціювання й інтегрування.

Основна частина

Варіант 1

Підінтегральна функція	Проміжок інтегрування	Кількість частин розбиття	Крок h	Первісна функція
$\frac{x}{(x+3)^2}$	[0; 2]	40	0.05	$\frac{x}{x+3} + \ln(x+3)$

Теоретичні відомості

Метод Сімпсона використовує інтерполяційний многочлен другого ступеня $p_2(x)$ для наближення значень підінтегральної функції на відрізку $[a, b]$, що фактично відповідає наближенню її графіка параболою на цьому відрізку.

$$\int_a^b f(x)dx \approx \int_a^b p_2(x)dx = \frac{b-a}{6} \left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right)$$

Рис. 1 – Формула Сімпсона

Для більш точного обчислення інтеграла, інтервал $[a, b]$ розбивають на N елементарних відрізків однакової довжини і застосовують формулу Сімпсона на складових відрізках. Кожен відрізок складається з сусідньої пари елементарних відрізків. Значення вихідного інтеграла є сумою результатів інтегрування на складових відрізках:

$$\int_a^b f(x)dx \approx \frac{h}{3} \cdot \left(\frac{1}{2}f(x_0) + \sum_{k=1}^{N-1} f(x_k) + 2 \sum_{k=1}^N f\left(\frac{x_{k-1} + x_k}{2}\right) + \frac{1}{2}f(x_N) \right)$$

Рис. 2 – Ітераційна формула Сімпсона

Похибка рахується шляхом віднімання по модулю результату за методом Сімпсона та результату за формулою Ньютона-Лейбніца.

Опис програми

Програма складається з двох частин – диференціювання та інтегрування.

Python:

Для інтегрування використовувалась функція `simps` [2] із бібліотеки `SymPy` [1]. Математичні функції задаються у текстовому файлі, а потім зчитуються програмою.

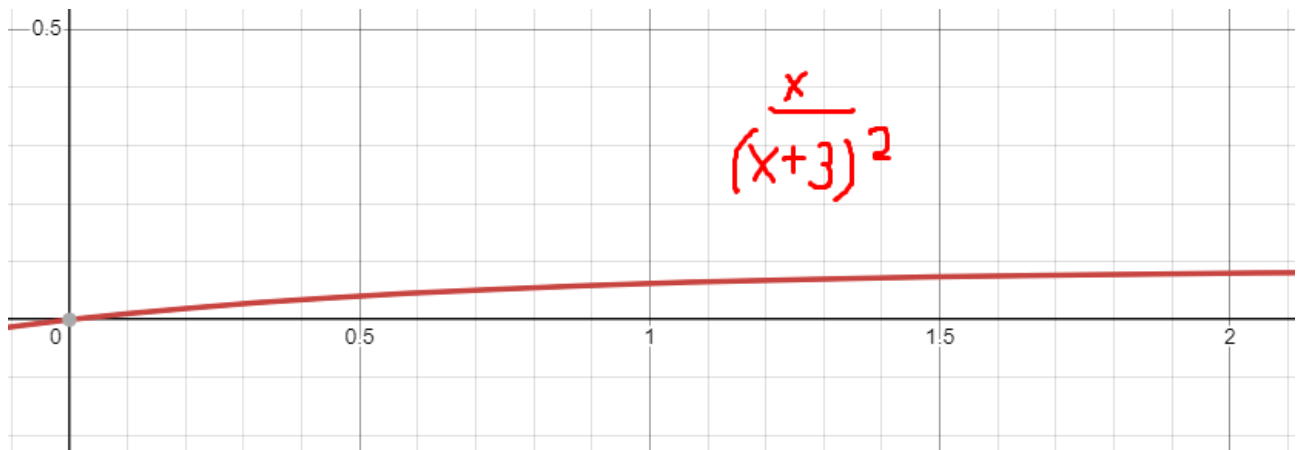
Диференціювання здійснюється функцією `diff` [3] із бібліотеки `SymPy` [1]

Octave:

Диференціювання здійснюється за допомогою символного диференціювання, однак якщо функція складна, то це може привести до помилок округлення при обчисленні значень похідних.

Для інтегрування використовується метод Сімпсона [2]. Нижня межа є статичною, а верхня змінюється з кроком $h = 0.05$

Задана функція на проміжку $[0; 2]$ є зростаючою, тому значення інтегралів мають зростати.



<https://www.desmos.com/calculator/l84wp4wlmg>

Висновки

У ході диференціювання та інтегрування було досягнуто таких результатів:

Диференціювання на Python:

Час виконання 0.3297 сек

	$f(x)$	$(F(x))'$	Різниця
0.00	0	0	0
0.05	0.00537489922063961	0.00537489922063961	7.80625564189563e-18
0.10	0.0104058272632674	0.0104058272632674	1.21430643318376e-17
0.15	0.0151171579743008	0.0151171579743008	3.46944695195361e-17
0.20	0.0195312500000000	0.0195312500000001	5.55111512312578e-17
0.25	0.0236686390532544	0.0236686390532544	1.04083408558608e-17
0.30	0.0275482093663912	0.0275482093663912	6.93889390390723e-18
0.35	0.0311873468478503	0.0311873468478503	3.46944695195361e-18
0.40	0.0346020761245675	0.0346020761245674	2.77555756156289e-17
0.45	0.0378071833648393	0.0378071833648393	2.77555756156289e-17
0.50	0.0408163265306122	0.0408163265306122	6.93889390390723e-18
0.55	0.0436421344971236	0.0436421344971236	2.77555756156289e-17
0.60	0.0462962962962963	0.0462962962962963	3.46944695195361e-17
0.65	0.0487896415837868	0.0487896415837868	4.16333634234434e-17
0.70	0.0511322132943755	0.0511322132943755	6.93889390390723e-18
0.75	0.0533333333333333	0.0533333333333333	1.38777878078145e-17
0.80	0.0554016620498615	0.0554016620498615	3.46944695195361e-17
0.85	0.0573452521504470	0.0573452521504469	1.38777878078145e-17
0.90	0.0591715976331361	0.0591715976331361	2.77555756156289e-17
0.95	0.0608876782566896	0.0608876782566896	1.38777878078145e-17
1.00	0.0625000000000000	0.0625000000000000	0
1.05	0.0640146319158665	0.0640146319158665	1.38777878078145e-17

1.10	0.0654372397382511	0.0654372397382510	2.77555756156289e-17
1.15	0.0667731165626361	0.0667731165626361	1.38777878078145e-17
1.20	0.0680272108843537	0.0680272108843537	1.38777878078145e-17
1.25	0.0692041522491350	0.0692041522491350	0
1.30	0.0703082747431044	0.0703082747431044	2.77555756156289e-17
1.35	0.0713436385255648	0.0713436385255648	1.38777878078145e-17
1.40	0.0723140495867769	0.0723140495867769	1.38777878078145e-17
1.45	0.0732230778942053	0.0732230778942053	1.38777878078145e-17
1.50	0.0740740740740741	0.0740740740740741	0
1.55	0.0748701847602947	0.0748701847602946	1.38777878078145e-17
1.60	0.0756143667296787	0.0756143667296786	1.38777878078145e-17
1.65	0.0763093999306278	0.0763093999306278	1.38777878078145e-17
1.70	0.0769578995020371	0.0769578995020371	0
1.75	0.0775623268698061	0.0775623268698061	1.38777878078145e-17
1.80	0.0781250000000000	0.0781250000000000	1.38777878078145e-17
1.85	0.0786481028802211	0.0786481028802211	2.77555756156289e-17
1.90	0.0791336942940442	0.0791336942940441	2.77555756156289e-17
1.95	0.0795837159473523	0.0795837159473523	0
2.00	0.0800000000000000	0.0800000000000000	1.38777878078145e-17

Інтегрування на Python:

Час виконання 0.2361 сек

	м.Сімпсона	м.Ньютона-Лейбніца	Різниця
0.00	0.000000	0.000000	0.000000e+00
0.05	0.000134	0.000136	1.486848e-06
0.10	0.000532	0.000532	1.237764e-09
0.15	0.001171	0.001171	4.680520e-08
0.20	0.002039	0.002039	2.249133e-09
0.25	0.003120	0.003120	3.829336e-08

0.30	0.004401	0.004401	3.080472e-09
0.35	0.005870	0.005870	3.129502e-08
0.40	0.007516	0.007516	3.767675e-09
0.45	0.009327	0.009327	2.550873e-08
0.50	0.011294	0.011294	4.338739e-09
0.55	0.013406	0.013406	2.069931e-08
0.60	0.015655	0.015655	4.815654e-09
0.65	0.018033	0.018033	1.668197e-08
0.70	0.020531	0.020531	5.215816e-09
0.75	0.023144	0.023144	1.331051e-08
0.80	0.025862	0.025862	5.553066e-09
0.85	0.028682	0.028682	1.046857e-08
0.90	0.031595	0.031595	5.838490e-09
0.95	0.034597	0.034597	8.062932e-09
1.00	0.037682	0.037682	6.081013e-09
1.05	0.040845	0.040845	6.018534e-09
1.10	0.044082	0.044082	6.287860e-09
1.15	0.047388	0.047388	4.274586e-09
1.20	0.050758	0.050758	6.464911e-09
1.25	0.054189	0.054189	2.781615e-09
1.30	0.057677	0.057677	6.616976e-09
1.35	0.061219	0.061219	1.499154e-09
1.40	0.064810	0.064810	6.748003e-09
1.45	0.068449	0.068449	3.939582e-10
1.50	0.072132	0.072132	6.861251e-09
1.55	0.075856	0.075856	5.614133e-10
1.60	0.079618	0.079618	6.959423e-09
1.65	0.083416	0.083416	1.389698e-09
1.70	0.087248	0.087248	7.044763e-09
1.75	0.091111	0.091111	2.109818e-09

1.80	0.095004	0.095004	7.119149e-09
1.85	0.098923	0.098923	2.737577e-09
1.90	0.102868	0.102868	7.184153e-09
1.95	0.106836	0.106836	3.286223e-09
2.00	0.110826	0.110826	7.241098e-09

Диференціювання на Octave:

x	f(x)	(F(x)')	Різниця
0.00	0.0000000000000000	0.1111111111111111	0.1111111111111111
0.05	0.005374899220640	0.103973460333684	0.098598561113045
0.10	0.010405827263267	0.097344835688631	0.086939008425363
0.15	0.015117157974301	0.091182857622767	0.076065699648466
0.20	0.019531250000000	0.085449218750000	0.065917968750000
0.25	0.023668639053254	0.080109239872554	0.056440600819299
0.30	0.027548209366391	0.075131480090158	0.047583270723767
0.35	0.031187346847850	0.070487393728617	0.039300046880767
0.40	0.034602076124567	0.066151027885203	0.031548951760635
0.45	0.037807183364839	0.062098755285243	0.024291571920404
0.50	0.040816326530612	0.058309037900875	0.017492711370262
0.55	0.043642134497124	0.054762217422767	0.011120082925644
0.60	0.046296296296296	0.051440329218107	0.005144032921811
0.65	0.048789641583787	0.048326936869083	0.000462704714704
0.70	0.051132213294375	0.045406984778789	0.005725228515586
0.75	0.0533333333333333	0.0426666666666667	0.010666666666667
0.80	0.055401662049862	0.040093308062400	0.015308353987462
0.85	0.057345252150447	0.037675261153082	0.019669990997365
0.90	0.059171597633136	0.035401810549740	0.023769787083397
0.95	0.060887678256690	0.033263088721176	0.027624589535513
1.00	0.062500000000000	0.031250000000000	0.031250000000000
1.05	0.064014631915866	0.029354152201279	0.034660479714587

1.10	0.065437239738251	0.027567795011680	0.037869444726571
1.15	0.066773116562636	0.025883764408775	0.040889352153861
1.20	0.068027210884354	0.024295432458698	0.043731778425656
1.25	0.069204152249135	0.022796661917362	0.046407490331773
1.30	0.070308274743104	0.021381765127599	0.048926509615506
1.35	0.071343638525565	0.020045466763249	0.051298171762315
1.40	0.072314049586777	0.018782870022539	0.053531179564237
1.45	0.073223077894205	0.017589425918019	0.055633651976186
1.50	0.074074074074074	0.016460905349794	0.057613168724280
1.55	0.074870184760295	0.015393373683435	0.059476811076859
1.60	0.075614366729679	0.014383167584450	0.061231199145229
1.65	0.076309399930628	0.013426873888087	0.062882526042541
1.70	0.076957899502037	0.012521310306965	0.064436589195072
1.75	0.077562326869806	0.011663507799971	0.065898819069835
1.80	0.078125000000000	0.010850694444444	0.067274305555556
1.85	0.078648102880221	0.010080280670076	0.068567822210145
1.90	0.079133694294044	0.009349845727545	0.069783848566499
1.95	0.079583715947352	0.008657125277878	0.070926590669474
2.00	0.080000000000000	0.008000000000000	0.072000000000000

Інтегрування на Octave:

х	м.Сімпсона	м.Ньютона-Лейбніца	Різниця
0.00	0.000000000000000	0.000000000000000	0.000000000000000
0.05	0.000134372480516	0.000135859328259	0.000001486847743
0.10	0.000528890642614	0.000531758306861	0.000002867664248
0.15	0.001166965273553	0.001171116550384	0.000004151276831
0.20	0.002033175472910	0.002038521137571	0.000005345664661
0.25	0.003113172699242	0.003119630750459	0.000006458051218
0.30	0.004393593909733	0.004401088895234	0.000007494985501
0.35	0.005861982815089	0.005870445228567	0.000008462413478

0.40	0.007506718389399	0.007516084130476	0.000009365741077
0.45	0.009316949876635	0.009327159766463	0.000010209889828
0.50	0.011282537624021	0.011293536970115	0.000010999346094
0.55	0.013393999149714	0.013405737354426	0.000011738204711
0.60	0.015642459919550	0.015654890127287	0.000012430207738
0.65	0.018019608366552	0.018032687145468	0.000013078778917
0.70	0.020517654738506	0.020531341792879	0.000013687054374
0.75	0.023129293404199	0.023143551314210	0.000014257910011
0.80	0.025847668288778	0.025862462274756	0.000014793985978
0.85	0.028666341143786	0.028681638852362	0.000015297708576
0.90	0.031579262388376	0.031595033698260	0.000015771309884
0.95	0.034580744285621	0.034596961130997	0.000016216845375
1.00	0.037665436242039	0.037682072451781	0.000016636209742
1.05	0.040828302039935	0.040845333191079	0.000017031151143
1.10	0.044064598831288	0.044082002115323	0.000017403284035
1.15	0.047369857738810	0.047387611839557	0.000017754100747
1.20	0.050739865924985	0.050757950906927	0.000018084981942
1.25	0.054170650003322	0.054189047209392	0.000018397206070
1.30	0.057658460678128	0.057677152636058	0.000018691957930
1.35	0.061199758509845	0.061218728846276	0.000018970336431
1.40	0.064791200712654	0.064810434074287	0.000019233361634
1.45	0.068429628899678	0.068449110880825	0.000019481981147
1.50	0.072112057698885	0.072131774774831	0.000019717075946
1.55	0.075835664169744	0.075855603635409	0.000019939465664
1.60	0.079597777956994	0.079617927870418	0.000020149913424
1.65	0.083395872123501	0.083416221253735	0.000020349130234
1.70	0.087227554609318	0.087248092388328	0.000020537779010
1.75	0.091090560268614	0.091111276746861	0.000020716478247
1.80	0.094982743440359	0.095003629245735	0.000020885805376
1.85	0.098902071012365	0.098923117312210	0.000021046299845

1.90	0.102846615941721	0.102867814407654	0.000021198465933
1.95	0.106814551197756	0.106835893973095	0.000021342775339
2.00	0.110804144096440	0.110825623765990	0.000021479669550

Результати непогані, але диференціювання на Octave показало себе не дуже добре, у порівнянні з диференціюванням на Python. Це вийшло через те що, на Octave був використаний символічний метод диференціювання, а на Python чисельний.

Скріншоти з результатами наведені в Додатку А.

Код програм наведений у Додатку Б.

Відповіді на контрольні питання

1. **Як залежить похибка усікання і похибка округлення від розміру кроку при чисельному диференціюванні ?** – Похибка усікання і похибка округлення в чисельному диференціюванні залежать від розміру кроку. Похибка усікання зазвичай зменшується при зменшенні розміру кроку, оскільки більша кількість точок дозволяє краще апроксимувати функцію. Однак похибка округлення може збільшуватися при зменшенні розміру кроку через накопичення помилок округлення при виконанні більшої кількості обчислень.
2. **Оцініть похибку заданої формули чисельного диференціювання за допомогою дослідження на апроксимацію** – Оцінка похибки заданої формули чисельного диференціювання може бути виконана за допомогою дослідження на апроксимацію. Це включає в себе порівняння результатів, отриманих за допомогою формули, з точними значеннями, якщо вони доступні, або з результатами, отриманими за допомогою більш точних методів.
3. **Проведіть порівняння точності формул трапецій і Сімпсона на підставі аналізу залишкових членів формул** – Точність формул трапецій і Сімпсона можна порівняти на основі аналізу залишкових членів формул. Зазвичай формула Сімпсона дає більш точні результати, ніж формула трапецій, оскільки вона базується на квадратичному апроксимаційному поліномі, тоді як формула трапецій використовує лінійний апроксимаційний поліном.
4. **У яких випадках доцільно використовувати квадратні формули Гауса ?** – Квадратурні формули Гауса доцільно використовувати, коли функція є достатньо гладкою і не має особливостей в межах інтегрування. Вони також ефективні для інтегрування поліноміальних функцій, оскільки вони можуть надати точний результат для поліномів до певного степеня.
5. **Які формули чисельного інтегрування зручніше програмувати для ЕОМ ?** – Для програмування на ЕОМ зручно використовувати ті формули чисельного інтегрування, які легко реалізуються і не вимагають складних обчислень. Це можуть бути такі методи, як метод прямокутників, метод трапецій та метод Сімпсона. Однак оптимальний вибір методу залежить від конкретної задачі і вимог до точності.

Перелік посилань

1. SymPy - <https://www.sympy.org/en/index.html>
2. Метод Сімпсона - <https://www.blacksacademy.net/texts/SimpsonMethod.pdf>
3. SymPy.diff - <https://docs.sympy.org/latest/tutorials/intro-tutorial/calculus.html>
4. Символьне диференціювання - <https://studfile.net/preview/7728558/page:92/>

Додаток А – Скріншоти роботи програми

Диференціювання за допомогою Python

Час виконання 0.3297 сек

	$f(x)$	$(F(x)')$	Різниця
0.00	0	0	0
0.05	0.00537489922063961	0.00537489922063961	7.80625564189563e-18
0.10	0.0104058272632674	0.0104058272632674	1.21430643318376e-17
0.15	0.0151171579743008	0.0151171579743008	3.46944695195361e-17
0.20	0.0195312500000000	0.0195312500000001	5.55111512312578e-17
0.25	0.0236686390532544	0.0236686390532544	1.04083408558608e-17
0.30	0.0275482093663912	0.0275482093663912	6.93889390390723e-18
0.35	0.0311873468478503	0.0311873468478503	3.46944695195361e-18
0.40	0.0346020761245675	0.0346020761245674	2.77555756156289e-17
0.45	0.0378071833648393	0.0378071833648393	2.77555756156289e-17
0.50	0.0408163265306122	0.0408163265306122	6.93889390390723e-18
0.55	0.0436421344971236	0.0436421344971236	2.77555756156289e-17
0.60	0.0462962962962963	0.0462962962962963	3.46944695195361e-17
0.65	0.0487896415837868	0.0487896415837868	4.16333634234434e-17
0.70	0.0511322132943755	0.0511322132943755	6.93889390390723e-18
0.75	0.0533333333333333	0.0533333333333333	1.38777878078145e-17
0.80	0.0554016620498615	0.0554016620498615	3.46944695195361e-17
0.85	0.0573452521504470	0.0573452521504469	1.38777878078145e-17
0.90	0.0591715976331361	0.0591715976331361	2.77555756156289e-17
0.95	0.0608876782566896	0.0608876782566896	1.38777878078145e-17
1.00	0.0625000000000000	0.0625000000000000	0
1.05	0.0640146319158665	0.0640146319158665	1.38777878078145e-17
1.10	0.0654372397382511	0.0654372397382510	2.77555756156289e-17
1.15	0.0667731165626361	0.0667731165626361	1.38777878078145e-17
1.20	0.0680272108843537	0.0680272108843537	1.38777878078145e-17
1.25	0.0692041522491350	0.0692041522491350	0
1.30	0.0703082747431044	0.0703082747431044	2.77555756156289e-17
1.35	0.0713436385255648	0.0713436385255648	1.38777878078145e-17
1.40	0.0723140495867769	0.0723140495867769	1.38777878078145e-17
1.45	0.0732230778942053	0.0732230778942053	1.38777878078145e-17
1.50	0.0740740740740741	0.0740740740740741	0
1.55	0.0748701847602947	0.0748701847602946	1.38777878078145e-17
1.60	0.0756143667296787	0.0756143667296786	1.38777878078145e-17
1.65	0.0763093999306278	0.0763093999306278	1.38777878078145e-17
1.70	0.0769578995020371	0.0769578995020371	0
1.75	0.0775623268698061	0.0775623268698061	1.38777878078145e-17
1.80	0.0781250000000000	0.0781250000000000	1.38777878078145e-17
1.85	0.0786481028802211	0.0786481028802211	2.77555756156289e-17
1.90	0.0791336942940442	0.0791336942940441	2.77555756156289e-17
1.95	0.0795837159473523	0.0795837159473523	0
2.00	0.0800000000000000	0.0800000000000000	1.38777878078145e-17

Рис. 1 – Диференціювання Python

Інтегрування за допомогою Python

Час виконання 0.2361 сек

	м.Сімпсона	м.Ньютона-Лейбніца	Різниця
0.00	0.000000	0.000000	0.000000e+00
0.05	0.000134	0.000136	1.486848e-06
0.10	0.000532	0.000532	1.237764e-09
0.15	0.001171	0.001171	4.680520e-08
0.20	0.002039	0.002039	2.249133e-09
0.25	0.003120	0.003120	3.829336e-08
0.30	0.004401	0.004401	3.080472e-09
0.35	0.005870	0.005870	3.129502e-08
0.40	0.007516	0.007516	3.767675e-09
0.45	0.009327	0.009327	2.550873e-08
0.50	0.011294	0.011294	4.338739e-09
0.55	0.013406	0.013406	2.069931e-08
0.60	0.015655	0.015655	4.815654e-09
0.65	0.018033	0.018033	1.668197e-08
0.70	0.020531	0.020531	5.215816e-09
0.75	0.023144	0.023144	1.331051e-08
0.80	0.025862	0.025862	5.553066e-09
0.85	0.028682	0.028682	1.046857e-08
0.90	0.031595	0.031595	5.838490e-09
0.95	0.034597	0.034597	8.062932e-09
1.00	0.037682	0.037682	6.081013e-09
1.05	0.040845	0.040845	6.018534e-09
1.10	0.044082	0.044082	6.287860e-09
1.15	0.047388	0.047388	4.274586e-09
1.20	0.050758	0.050758	6.464911e-09
1.25	0.054189	0.054189	2.781615e-09
1.30	0.057677	0.057677	6.616976e-09
1.35	0.061219	0.061219	1.499154e-09
1.40	0.064810	0.064810	6.748003e-09
1.45	0.068449	0.068449	3.939582e-10
1.50	0.072132	0.072132	6.861251e-09
1.55	0.075856	0.075856	5.614133e-10
1.60	0.079618	0.079618	6.959423e-09
1.65	0.083416	0.083416	1.389698e-09
1.70	0.087248	0.087248	7.044763e-09
1.75	0.091111	0.091111	2.109818e-09
1.80	0.095004	0.095004	7.119149e-09
1.85	0.098923	0.098923	2.737577e-09
1.90	0.102868	0.102868	7.184153e-09
1.95	0.106836	0.106836	3.286223e-09
2.00	0.110826	0.110826	7.241098e-09

Рис. 2 – Інтегрування Python

Диференціювання за допомогою Octave			
x	f(x)	(F(x)')	Різниця
0.00	0.0000000000000000	0.1111111111111111	0.1111111111111111
0.05	0.005374899220640	0.103973460333684	0.098598561113045
0.10	0.010405827263267	0.097344835688631	0.086939008425363
0.15	0.015117157974301	0.091182857622767	0.076065699648466
0.20	0.0195312500000000	0.0854492187500000	0.0659179687500000
0.25	0.023668639053254	0.080109239872554	0.056440600819299
0.30	0.027548209366391	0.075131480090158	0.047583270723767
0.35	0.031187346847850	0.070487393728617	0.039300046880767
0.40	0.034602076124567	0.066151027885203	0.031548951760635
0.45	0.037807183364839	0.062098755285243	0.024291571920404
0.50	0.040816326530612	0.058309037900875	0.017492711370262
0.55	0.043642134497124	0.054762217422767	0.011120082925644
0.60	0.046296296296296	0.051440329218107	0.005144032921811
0.65	0.048789641583787	0.048326936869083	0.000462704714704
0.70	0.051132213294375	0.045406984778789	0.005725228515586
0.75	0.0533333333333333	0.0426666666666667	0.010666666666667
0.80	0.055401662049862	0.040093308062400	0.015308353987462
0.85	0.057345252150447	0.037675261153082	0.019669990997365
0.90	0.059171597633136	0.035401810549740	0.023769787083397
0.95	0.060887678256690	0.033263088721176	0.027624589535513
1.00	0.0625000000000000	0.0312500000000000	0.0312500000000000
1.05	0.064014631915866	0.029354152201279	0.034660479714587
1.10	0.065437239738251	0.027567795011680	0.037869444726571
1.15	0.066773116562636	0.025883764408775	0.040889352153861
1.20	0.068027210884354	0.024295432458698	0.043731778425656
1.25	0.069204152249135	0.022796661917362	0.046407490331773
1.30	0.070308274743104	0.021381765127599	0.048926509615506
1.35	0.071343638525565	0.020045466763249	0.051298171762315
1.40	0.072314049586777	0.018782870022539	0.053531179564237
1.45	0.073223077894205	0.017589425918019	0.055633651976186
1.50	0.074074074074074	0.016460905349794	0.057613168724280
1.55	0.074870184760295	0.015393373683435	0.059476811076859
1.60	0.075614366729679	0.014383167584450	0.061231199145229
1.65	0.076309399930628	0.013426873888087	0.062882526042541
1.70	0.076957899502037	0.012521310306965	0.064436589195072
1.75	0.077562326869806	0.011663507799971	0.065898819069835
1.80	0.0781250000000000	0.010850694444444	0.067274305555556
1.85	0.078648102880221	0.010080280670076	0.068567822210145
1.90	0.079133694294044	0.009349845727545	0.069783848566499
1.95	0.079583715947352	0.008657125277878	0.070926590669474
2.00	0.0800000000000000	0.0080000000000000	0.0720000000000000

Рис. 3 – Диференціювання Octave

```

Інтегрування за допомогою Octave

      х      м.Сімпсона      м.Ньютона-Лейбніца      Різниця
0.00      0.0000000000000000      0.0000000000000000      0.0000000000000000
0.05      0.000134372480516      0.000135859328259      0.000001486847743
0.10      0.000528890642614      0.000531758306861      0.000002867664248
0.15      0.001166965273553      0.001171116550384      0.000004151276831
0.20      0.002033175472910      0.002038521137571      0.000005345664661
0.25      0.003113172699242      0.003119630750459      0.000006458051218
0.30      0.004393593909733      0.004401088895234      0.000007494985501
0.35      0.005861982815089      0.005870445228567      0.000008462413478
0.40      0.007506718389399      0.007516084130476      0.000009365741077
0.45      0.009316949876635      0.009327159766463      0.000010209889828
0.50      0.011282537624021      0.011293536970115      0.000010999346094
0.55      0.013393999149714      0.013405737354426      0.000011738204711
0.60      0.015642459919550      0.015654890127287      0.000012430207738
0.65      0.018019608366552      0.018032687145468      0.000013078778917
0.70      0.020517654738506      0.020531341792879      0.000013687054374
0.75      0.023129293404199      0.023143551314210      0.000014257910011
0.80      0.025847668288778      0.025862462274756      0.000014793985978
0.85      0.028666341143786      0.028681638852362      0.000015297708576
0.90      0.031579262388376      0.031595033698260      0.000015771309884
0.95      0.034580744285621      0.034596961130997      0.000016216845375
1.00      0.037665436242039      0.037682072451781      0.000016636209742
1.05      0.040828302039935      0.040845333191079      0.000017031151143
1.10      0.044064598831288      0.044082002115323      0.000017403284035
1.15      0.047369857738810      0.047387611839557      0.000017754100747
1.20      0.050739865924985      0.050757950906927      0.000018084981942
1.25      0.054170650003322      0.054189047209392      0.000018397206070
1.30      0.057658460678128      0.057677152636058      0.000018691957930
1.35      0.061199758509845      0.061218728846276      0.000018970336431
1.40      0.064791200712654      0.064810434074287      0.000019233361634
1.45      0.068429628899678      0.068449110880825      0.000019481981147
1.50      0.072112057698885      0.072131774774831      0.000019717075946
1.55      0.075835664169744      0.075855603635409      0.000019939465664
1.60      0.079597777956994      0.079617927870418      0.000020149913424
1.65      0.083395872123501      0.083416221253735      0.000020349130234
1.70      0.087227554609318      0.087248092388328      0.000020537779010
1.75      0.091090560268614      0.0911111276746861      0.000020716478247
1.80      0.094982743440359      0.095003629245735      0.000020885805376
1.85      0.098902071012365      0.098923117312210      0.000021046299845
1.90      0.102846615941721      0.102867814407654      0.000021198465933
1.95      0.106814551197756      0.106835893973095      0.000021342775339
2.00      0.110804144096440      0.110825623765990      0.000021479669550
>> |

```

Рис. 4 – Інтегрування Octave

Додаток В – Код програми

Вміст файлу main.py :

```
import time
import numpy as np
import pandas as pd
import sympy as sp
from scipy.integrate import.simps

def get_func():
    '''Читання формули, первісної та меж інтегрування з файлу
    ...
    with open('D:\KPI\ASKM\Labs\Reznichenko\Lab_2\umova.txt', 'r') as f:
        lines = f.readlines()
        lower_limit = float(lines[0].strip())
        upper_limit = float(lines[1].strip())
        n = float(lines[2].strip())
        func = lines[3].strip()
        antiderivative = lines[4].strip()
    return [lower_limit, upper_limit, n, func, antiderivative]

def dif():
    start_timer = time.time()
    lower_limit, upper_limit, n, func_str, antiderivative_str = get_func()

    try:
        func_str = func_str.replace('.', '')
        func_str = func_str.replace('^', '**')
        antiderivative_str = antiderivative_str.replace('.', '')
        antiderivative_str = antiderivative_str.replace('^', '**')

        x = sp.symbols('x')
        func = sp.sympify(func_str) # Функція
        antiderivative = sp.sympify(antiderivative_str) # Первісна

        antiderivative = sp.diff(antiderivative, x) # Обчислення формули похідної

        h = (upper_limit - lower_limit) / n # Крок
        x_values = np.arange(lower_limit, upper_limit+h, h) # Створюємо масив
        # Обчислення значень функцій у цих точках
        func_values = [func.subs(x, val).evalf() for val in x_values]
        antiderivative_values = [antiderivative.subs(x, val).evalf() for val in
x_values]

        # Обчислення різниці між значеннями функцій
```

```

difference = np.array(func_values) - np.array(antiderivative_values)

diff_py = pd.DataFrame({
    'f(x)': func_values,
    '(F(x)\')': antiderivative_values,
    'Різниця': np.abs(difference)
}, index=x_values)

print('\n\nДиференціювання за допомогою Python\n')
print(f'Час виконання {round(time.time() - start_timer, 4)} сек\n')
print(diff_py)
except:
    print('SYNTAX ERROR')

def integr():
    start_timer = time.time()
    lower_limit, upper_limit, n, func_str, antiderivative_str = get_func()

    try:
        func_str = func_str.replace('.', '')
        func_str = func_str.replace('^', '**')
        antiderivative_str = antiderivative_str.replace('.', '')
        antiderivative_str = antiderivative_str.replace('^', '**')

        x = sp.symbols('x')
        func = sp.sympify(func_str) # Функція
        antiderivative = sp.sympify(antiderivative_str) # Первісна

        int_func = sp.integrate(func, x) # Інтеграл від функції

        h = (upper_limit - lower_limit) / n # Крок
        x_values = np.arange(lower_limit, upper_limit+h, h) # Створюємо масив
        # точок на інтервалі [a, b] з кроком h

        # Перетворюємо функцію та первісну у викликаємі функції
        func = sp.lambdify(x, func)
        antiderivative = sp.lambdify(x, antiderivative)

        y_values = [func(point) for point in x_values] # Створюємо масив значень
        # функції для кожної точки

        # Обчислюємо чисельне значення визначеного інтегралу для кожної точки
        simpson = [simps(y_values[:i+1], x_values[:i+1], dx=h) for i in
        range(len(x_values))]

        # Обчислюємо чисельне значення визначеного інтегралу за допомогою методу
        # Ньютона-Лейбніца
        newton_leibniz = [antiderivative(point) - antiderivative(lower_limit) for
        point in x_values]

        # Обчислюємо різницю між значеннями, отриманими двома методами

```

```

        difference = [abs(simpson[i] - newton_leibniz[i]) for i in
range(len(x_values))]

    int_py = pd.DataFrame({
        'м.Сімпсона': simpson,
        'м.Ньютона-Лейбніца': newton_leibniz,
        'Різниця': difference
    }, index=x_values)

    print('\n\nІнтегрування за допомогою Python\n')
    print(f'Час виконання {round(time.time() - start_timer, 4)} сек\n')
    print(int_py)
except:
    print('SYNTAX ERROR')

if __name__ == '__main__':
    lower_limit, upper_limit, _, func, _ = get_func()
    if lower_limit > upper_limit:
        print('lower_limit > upper_limit')

    dif()

    integr()

```

Вміст файлу main.m :

```
pkg load symbolic; % Завантаження пакету symbolic
```

```
% Читання формули, первісної та меж інтегрування з файлу
```

```
try
```

```
    fid = fopen('D:\\KPI\\ASKM\\Labs\\Reznichenko\\Lab_2\\umova.txt', 'r');
```

```
    lower_limit = str2double(fgetl(fid));
```

```
    upper_limit = str2double(fgetl(fid));
```

```
    n = str2double(fgetl(fid));
```

```
    func_str = fgetl(fid);
```

```
    antiderivative_str = fgetl(fid);
```

```
    fclose(fid);
```

```
catch
```

```
    disp('ERROR: Помилка при читанні файлу');
```

```
end_try_catch
```

```
% Заміна символів у рядках
```

```
func_str = strrep(func_str, '.', '');
```

```
func_str = strrep(func_str, '**', '^');
```

```
antiderivative_str = strrep(antiderivative_str, '.', '');
```

```
antiderivative_str = strrep(antiderivative_str, '**', '^');
```

```
% Обчислення формули похідної
```

```
func_derivative = diff(sym(func_str));
```

```
% Обчислення значень функцій у цих точках
```

```
h = (upper_limit - lower_limit) / n; % Крок
```

```
x_values = lower_limit:h:upper_limit; % Створюємо масив точок на інтервалі [a,  
b] з кроком h
```

```
func_values = arrayfun(@(x) eval(func_str), x_values);
```

```
func_derivative_values = arrayfun(@(x) eval(func_derivative), x_values);
```

```
% Обчислення різниці між значеннями функцій
```

```
difference = func_values - func_derivative_values;
```

```
% Виведення результатів
```

```
printf('\n\n')
```

```
disp('Диференціювання за допомогою Octave');
```

```
printf('%5s %20s %20s %20s\n', 'x', 'f(x)', 'F(x)', 'Різниця');
```

```
printf('%5.2f %20.15f %20.15f %20.15f\n', [x_values; func_values;  
func_derivative_values; abs(difference)]);
```

```
% Обчислюємо чисельне значення визначеного інтегралу для кожної точки
```

```
simpson = arrayfun(@(i) trapz(x_values(1:i), func_values(1:i)), 1:length(x_values));
```

% Обчислюємо чисельне значення визначеного інтегралу за допомогою методу Ньютона-Лейбніца

```
newton_leibniz = arrayfun(@(x) eval(strrep(antiderivative_str, 'x', num2str(x))),  
x_values) - eval(strrep(antiderivative_str, 'x', num2str(lower_limit)));
```

% Обчислюємо різницю між значеннями, отриманими двома методами

```
difference = abs(simpson - newton_leibniz);
```

% Виведення результатів

```
printf('\n\n')
```

```
disp('Інтегрування за допомогою Octave');
```

```
printf('%5s %25s %40s %25s\n', 'x', 'м.Сімпсона', 'м.Ньютона-Лейбніца', 'Різниця');
```

```
printf('%5.2f %20.15f %20.15f %20.15f\n', [x_values; simpson; newton_leibniz;  
difference]);
```