

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
Факультет прикладної математики
Кафедра прикладної математики

Звіт
із курсової роботи
з дисципліни «Методи оптимізацій»
на тему
«Безумовна та умовна оптимізації»

Виконав:
студент групи КМ-03
Орленко Антон Сергійович

Керівник:
Ладогубець Т. С.

Зміст

ВСТУП.....	3
1. ПОСТАНОВКА ЗАДАЧІ.....	4
2. ТЕОРЕТИЧНА ЧАСТИНА.....	5
3. ХІД РОБОТИ.....	6
3.1. Безумовна оптимізація.....	6
3.1.1. Визначення оптимального кроку обчислення похідних	7
3.1.2. Визначення оптимальної схеми обчислення похідних	9
3.1.3. Визначення оптимального значення параметру в алгоритмі Свенна ..	11
3.1.4. Визначення оптимальної точності методу одновимірного пошуку ..	14
3.1.5. Визначення оптимального методу одновимірного пошуку.....	16
3.1.6. Визначення оптимального критерію закінчення	18
3.1.7. Визначення необхідності наявності рестартів	20
3.2. Умовна оптимізація.....	21
3.2.1. Метод штрафних функцій в залежності від розташування локального мінімуму	21
3.2.2. Метод штрафних функцій в залежності від виду допустимої області (невипукла чи випукла)	24
ВИСНОВКИ.....	27
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	28
ДОДАТКИ.....	29

ВСТУП

Методи оптимізації відіграють важливу роль у багатьох галузях науки та технологій, де необхідно знайти найкращий розв'язок для задачі мінімізації або максимізації певної функції. Одним з найпоширеніших методів оптимізації є метод Пірсона, який базується на ітеративному покращенні поточного розв'язку шляхом використання градієнтних напрямків.

У даній курсовій роботі ми зосередимося на дослідженні збіжності метода Пірсона (алгоритм 2) при мінімізації функції Розенброка. Функція Розенброка є широко використовуваною для тестування ефективності різних методів оптимізації, оскільки вона представляє собою складну нелінійну функцію з багатьма локальними мінімумами та одним глобальним мінімумом.

Метою даної роботи є дослідження впливу різних факторів на збіжність метода Пірсона при мінімізації функції Розенброка. Ми розглянемо такі фактори, як початкова точка, величина кроку, критерій зупинки та похибка збіжності. Варіюючи ці фактори, ми будемо аналізувати, як вони впливають на швидкість збіжності метода Пірсона та точність отриманого розв'язку.

1. ПОСТАНОВКА ЗАДАЧІ

Дослідити збіжність метода Пірсона (алгоритм 2) при мінімізації функції Розенброка в залежності від:

1. Величини кроку h при обчисленні похідних.
2. Схеми обчислення похідних.
3. Виду методу одновимірного пошуку (ДСК-Пауелла або золотого перетину).
4. Точності методу одновимірного пошуку.
5. Значення параметру в алгоритмі Свена.
6. Вигляду критерію закінчення.

$$\left\{ \begin{array}{l} \frac{\|x^{k+1} - x^k\|}{\|x^k\|} \leq \varepsilon \\ |f(x^{k+1}) - f(x^k)| \leq \varepsilon \end{array} \right. \text{ або } \|\nabla f(x^k)\| \leq \varepsilon.$$

7. Наявності рестартів.

Використати метод штрафних функцій (метод зовнішньої точки) для умовної оптимізації в залежності від:

1. Розташування локального мінімуму (всередині/поза допустимою областю).
2. Виду допустимої області (випукла/невипукла).

2. ТЕОРЕТИЧНА ЧАСТИНА

Алгоритм Пірсона (2) – квазіньютонівський алгоритм, що базується на апроксимації матриці, оберненої до матриці Гессе в точці. Вперше опублікований у журналі «The Computer Journal» у 1969р. Формула, що використовується для обчислення $A^{(k+1)}$:

$$A^{(k+1)} = A^{(k)} + [\Delta x^{(k)} - A^{(k)} \Delta g^{(k)}] * \frac{[A^{(k)} \Delta g^{(k)}]^T}{\Delta g^{(k)T} A^{(k)} \Delta g^{(k)}}$$

Обчислення величини кроку (λ) здійснюється з використанням алгоритму Свенна для знаходження інтервалу невизначеності, та методу золотого перетину або ДСК-Пауела для знаходження конкретного значення на цьому відрізку [2].

Формула для знаходження інтервалу невизначеності в алгоритмі Свенна:

$$x_{k+1} = x_k \pm \Delta * 2^k; k = \overline{0, 1, \dots, N}$$

$$\Delta = q * \frac{||x||}{||s||}$$

3. ХІД РОБОТИ

3.1. Безумовна оптимізація

Перелік початкових налаштувань:

- Початковою точкою було обрано взяти $[-1.2, 0]$
- Початкове значення в алгоритмі Свенна: 10^{-3}
- Початковий крок похідних: 10^{-3}
- Початкова схема обчислення похідних: центр
- Алгоритм одновимірного пошуку: золотий перетин
- Точність одновимірного пошуку: 10^{-3}
- Очікувана точність: 10^{-7}
- Наявність рестартів: так, 100 максимум
- Максимальна дозволена кількість ітерацій алгоритму Пірсона: 100000 [1].

3.1.1. Визначення оптимального кроку обчислення похідних

Було обрано наступний набір можливих значень h :

[1, 0.1, 0.01, 0.001, 0.0001, 1e-05, 1e-06, 1e-07, 1e-08, 1e-09, 1e-10]

В результаті отримані наступні результати:

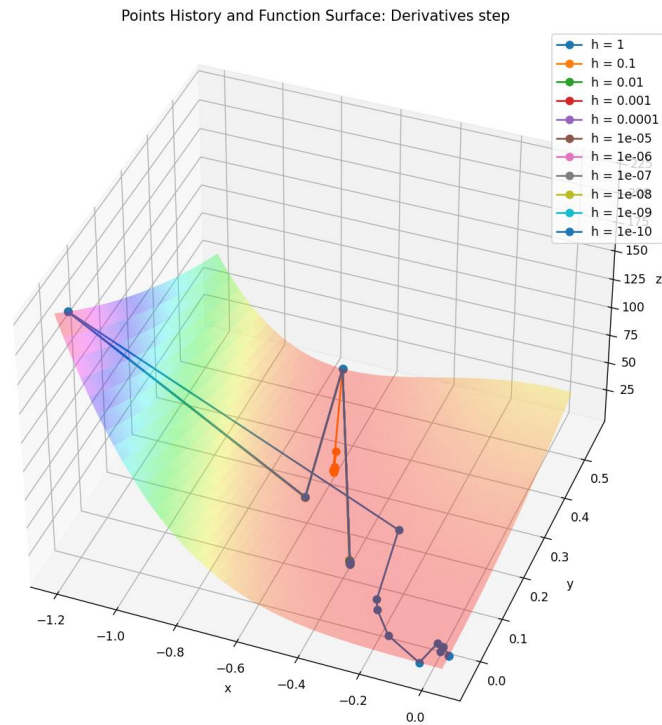


Рисунок 1 – шлях пошуку мінімуму в залежності від кроку похідних

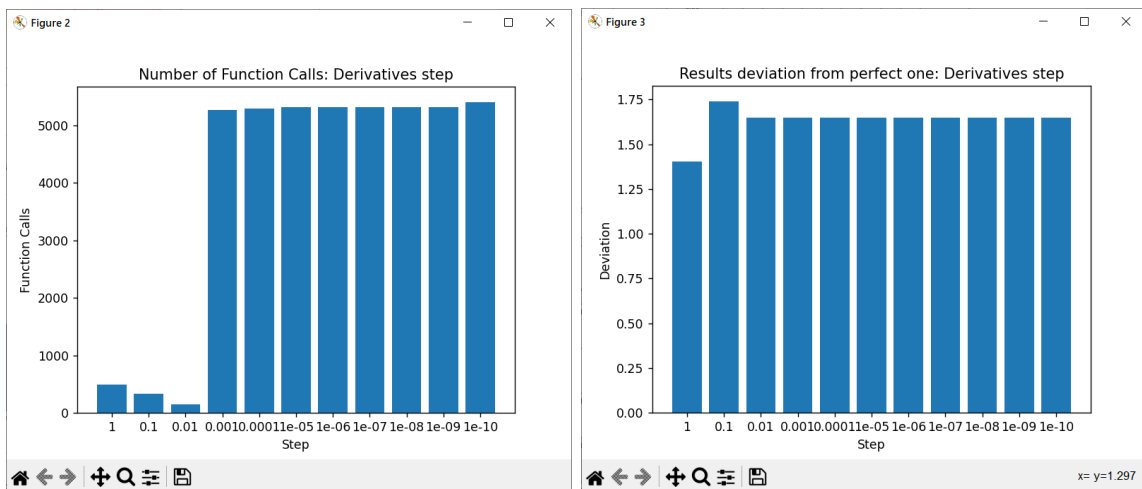


Рисунок 2 – кількість викликів функції в залежності від кроку похідних, та відхилення від ідеального розв'язку

Таблиця результатів:

Таблиця 1 – Результат тестування кроку обчислення похідних

h	Результат	Кількість викликів функції	Відхилення	F(x)
1	[0.03289443 -0.01522221]	485	1.402130280155859	0.9618760532909798
0.1	[-0.602141 0.32098591]	335	1.7400907798663194	2.7398109253401843
0.01	[-0.40295233 0.1298756]	153	1.6508760396482913	2.0738675753306
0.001	[-0.3971871 0.12577414]	5267	1.648151284625934	2.054425949546751
0.0001	[-0.39712603 0.12572979]	5289	1.6481230388092603	2.0542286909862715
1e-05	[-0.39712539 0.1257293]	5311	1.6481227586756988	2.054226807531317
1e-06	[-0.39712539 0.1257293]	5311	1.6481227568286938	2.054226807531307
1e-07	[-0.39712539 0.1257293]	5311	1.648122756823406	2.0542267885218335
1e-08	[-0.39712539 0.1257293]	5311	1.6481227573398742	2.054226793355458
1e-09	[-0.39712536 0.12572932]	5311	1.6481227165120353	2.0542263812577377
1e-10	[-0.39712903 0.12573198]	5399	1.648124418765861	2.05423828732074

Як бачимо, кількість викликів функцій різко зростає після кроку у 0.001, у той час як найкраще наближення дає значення у 1.

В результаті обираємо значення 1.

3.1.2. Визначення оптимальної схеми обчислення похідних

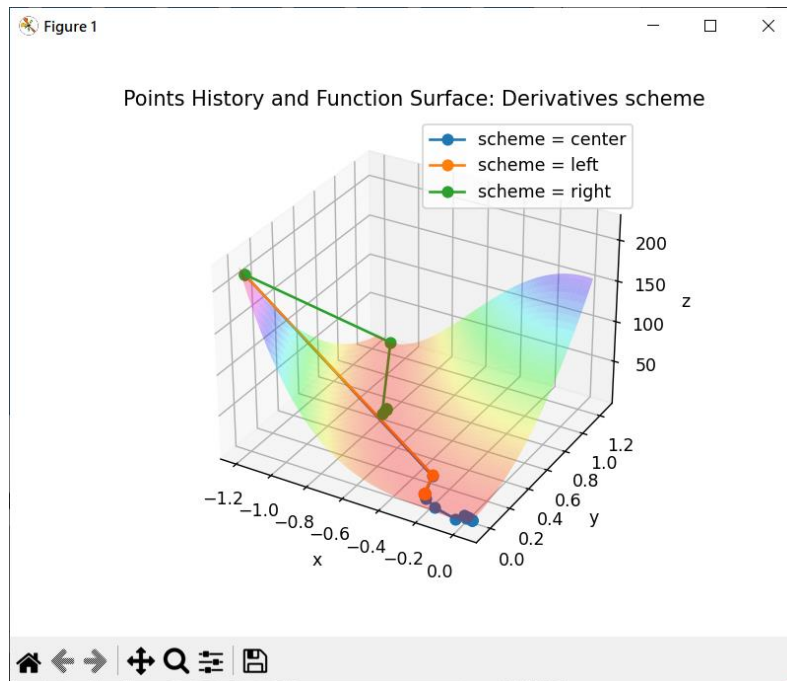


Рисунок 3 - шлях пошуку мінімуму в залежності від схеми обчислення похідних

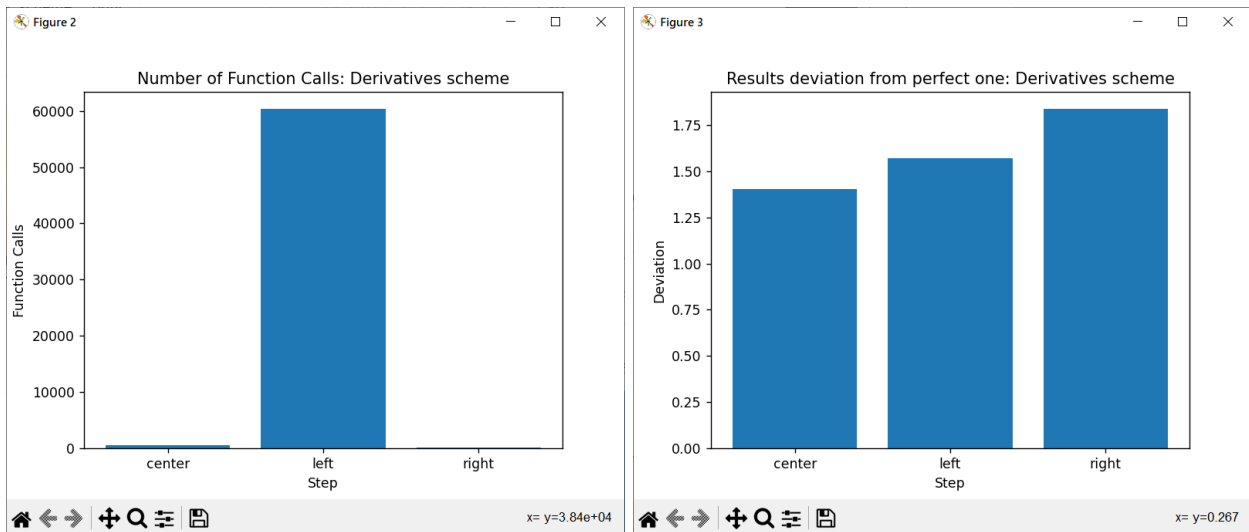


Рисунок 4 - кількість викликів функції в залежності від схеми обчислення похідних, та відхилення від ідеального розв'язку

Таблиця результатів:

Таблиця 2 - Результат тестування різних схем обчислення похідних

Схема	Результат	Кількість викликів функції	Відхилення	F(x)
Центральна	[0.03289443 - 0.01522221]	485	1.402130280155859	0.9618760532909798
Ліва	[-0.2672644 0.07473709]	60363	1.5690986278141381	1.607052573071719
Права	[-0.80399962 0.65057985]	151	1.8375279774121425	3.2561489001443737

Не зважаючи на те, що права схема обчислення похідних впоралась за меншу кількість викликів функції, центральна дає кращу точність. Отже, обираємо центральну схему обчислення.

3.1.3. Визначення оптимального значення параметру в алгоритмі Свенна

Наступним тестом я вирішив обрати тест коефіцієнту Свенна, адже, на мою думку, він має чи не найбільший вплив на точність обчислень. Для тестування було обрано наступні значення:

[1, 1e-1, 1e-2, 1e-3, 1e-4, 1e-5, 1e-6, 1e-7, 1e-8, 1e-9, 1e-10]

Після тестування отримано наступні результати:

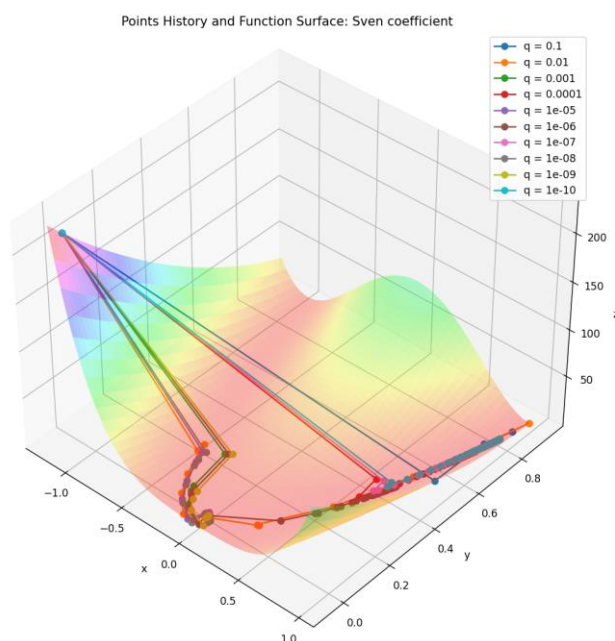


Рисунок 5 - шлях пошуку мінімуму в залежності від значення в алгоритмі Свенна



Рисунок 6 - кількість викликів функції в залежності від значення в алгоритмі Свенна, та відхилення від ідеального розв'язку

Таблиця результатів:

Таблиця 3 - Результат тестування різних коефіцієнтів в алгоритмі Свенна

q	Результат	Кількість викликів функції	Відхилення	F(x)
0.1	[0.93127721 0.86228005]	86	0.15391428136899454	0.007220004608210852
0.01	[0.96256815 0.92463989]	456	0.08414445947227292	0.0017612148610198345
0.001	[0.03289443 -0.01522221]	458	1.402130280155859	0.9618760532909798
0.0001	[0.86825761 0.72074126]	1187	0.3087741918333492	0.1271158273193447
1e-05	[0.0153318 -0.00981332]	254247	1.4104234839399876	0.979668465185977
1e-06	[0.70334814 0.45015873]	2142	0.624762153511848	0.28638233402158936
1e-07	[0.89947301 0.77949633]	2466	0.24233767116626123	0.09745761614778409
1e-08	[0.01088469 -0.01179098]	1398	1.4149452600674755	0.9925326190236207
1e-09	[-0.00200772 -0.00487846]	1042	1.4190842777331933	1.006403343936101
1e-10	[0.91572765 0.8112237]	2840	0.2067324818974631	0.08181342109802764

Найбільш відповідним значенням було обрано 0.01, адже за цього значення досягається найкращий результат, до того ж за адекватну кількість викликів функції.

3.1.4. Визначення оптимальної точності методу одновимірного пошуку

Обрали наступний набір можливих значень eps:

[1, 0.1, 0.01, 0.001, 0.0001, 1e-05, 1e-06, 1e-07, 1e-08, 1e-09, 1e-10]

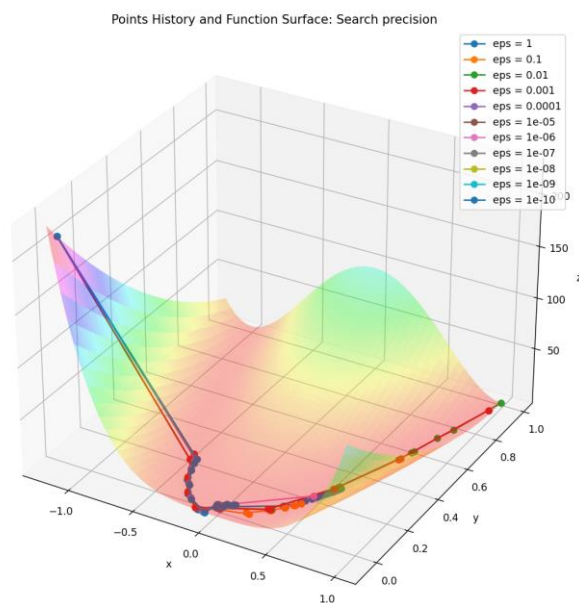


Рисунок 7 - шлях пошуку мінімуму в залежності від точності одновимірного пошуку

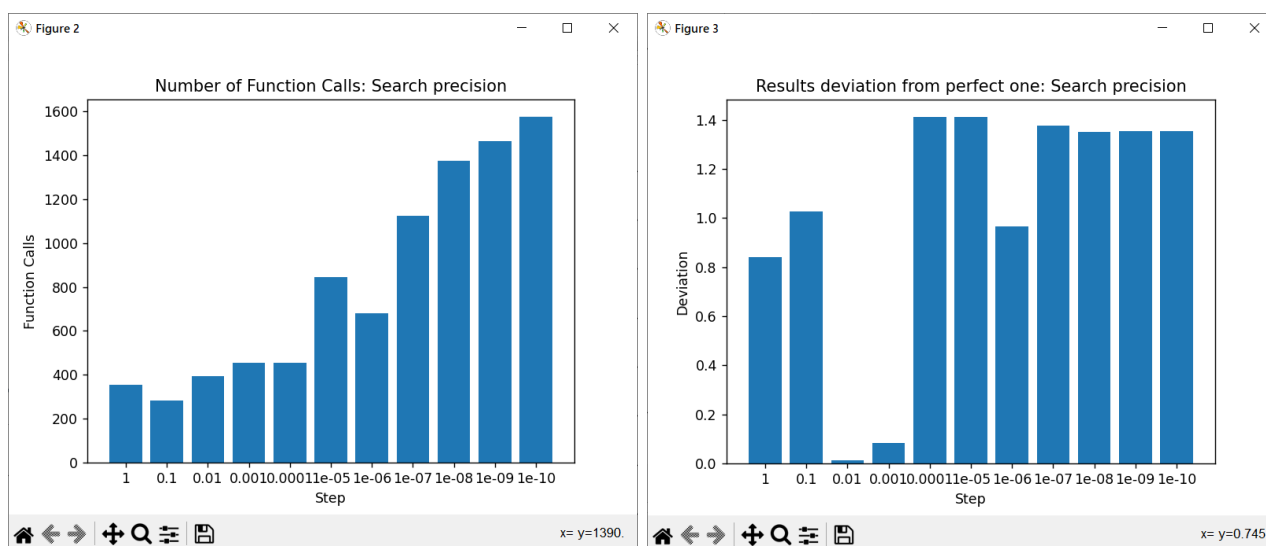


Рисунок 8 - кількість викликів функції в залежності від точності одновимірного пошуку, та відхилення від ідеального розв'язку

Таблиця результатів:

Таблиця 4 - Результат тестування різних точностей одновимірного пошуку

eps	Результат	Кількість викликів функції	Відхилення	F(x)
1	[0.56862551 0.2777755]	345	0.8412444223604624	0.3936505001606073
0.1	[0.43238782 0.14208526]	284	1.0286891097372695	0.5235509009810673
0.01	[0.99381371 0.98704685]	393	0.014354590156989953	7.656691055730527e-05
0.001	[0.96256815 0.92463989]	456	0.08414445947227292	0.0017612148610198345
0.0001	[0.00744052 -0.00487396]	455	1.4124255722180026	0.9876041458177693
1e-05	[0.00768719 -0.00514561]	846	1.41244553991344	0.987393600319339
1e-06	[0.45801695 0.19905555]	679	0.967087192316439	0.3052459892499895
1e-07	[0.0356166 0.01862755]	1125	1.3759096017155361	0.9601688542853817
1e-08	[0.06022443 0.02663069]	1375	1.3530062565101229	0.936095182224552
1e-09	[0.05960459 0.02643616]	1466	1.3535767736936943	0.936708787261382
1e-10	[0.06013798 0.02660358]	1576	1.3530858045356284	0.9361808510029231

Найкращою точністю є 0.01, адже за 393 виклики функції вдалося обчислити значення мінімуму з найбільшою точністю.

3.1.5. Визначення оптимального методу одновимірного пошуку

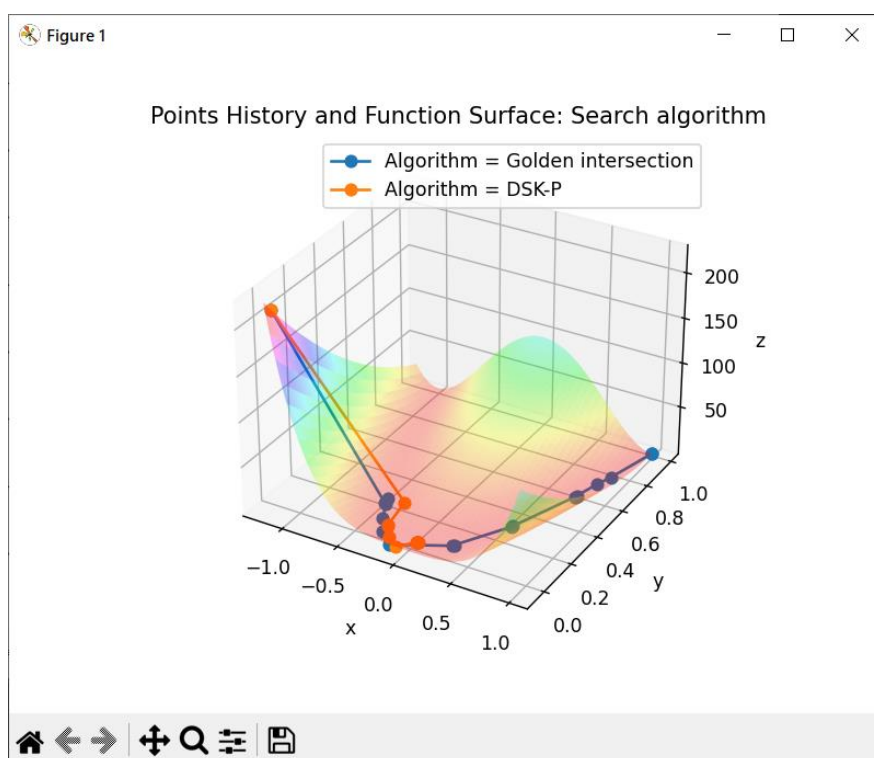


Рисунок 9 - шлях пошуку мінімуму в залежності від обраного методу одновимірного пошуку

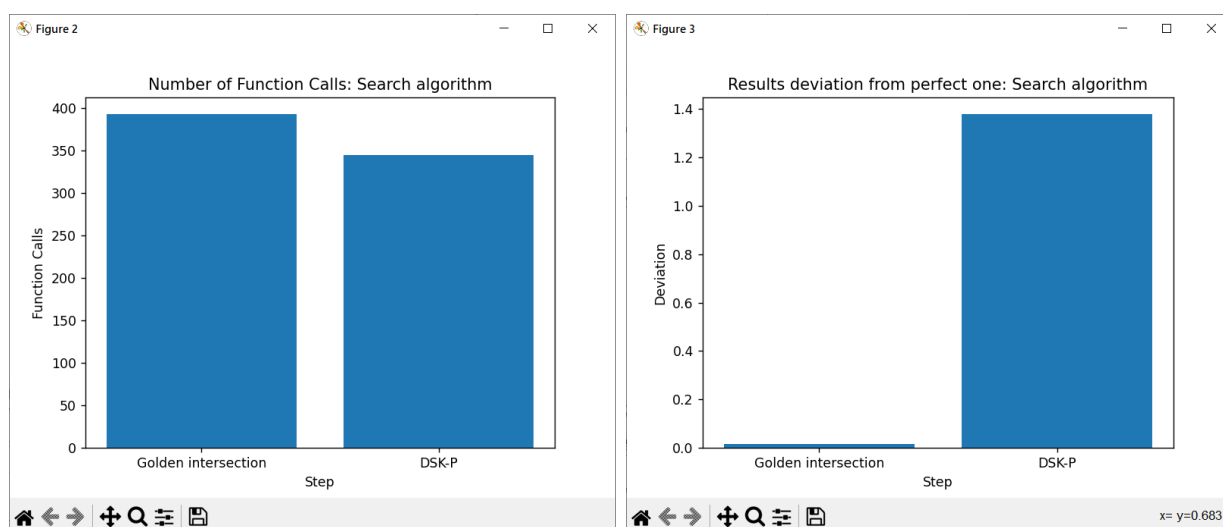


Рисунок 10 - кількість викликів функції в залежності від обраного методу одновимірного пошуку, та відхилення від ідеального розв'язку

Таблиця результатів:

Таблиця 5 - Результат тестування різних варіантів одновимірного пошуку

Метод пошуку	Результат	Кількість викликів функції	Відхилення	F(x)
Золотий перетин	[0.99381371 0.98704685]	393	0.014354590156989953	7.656691055730527e-05
ДСК-П	[0.03274737 0.01762627]	345	1.3786354865710493	0.9629807535231688

З результатів дослідження можна зробити висновок, що за використання методу золотого перетину отримуємо більшу точність, ціною невеликого зростання кількості викликів функції.

3.1.6. Визначення оптимального критерію закінчення

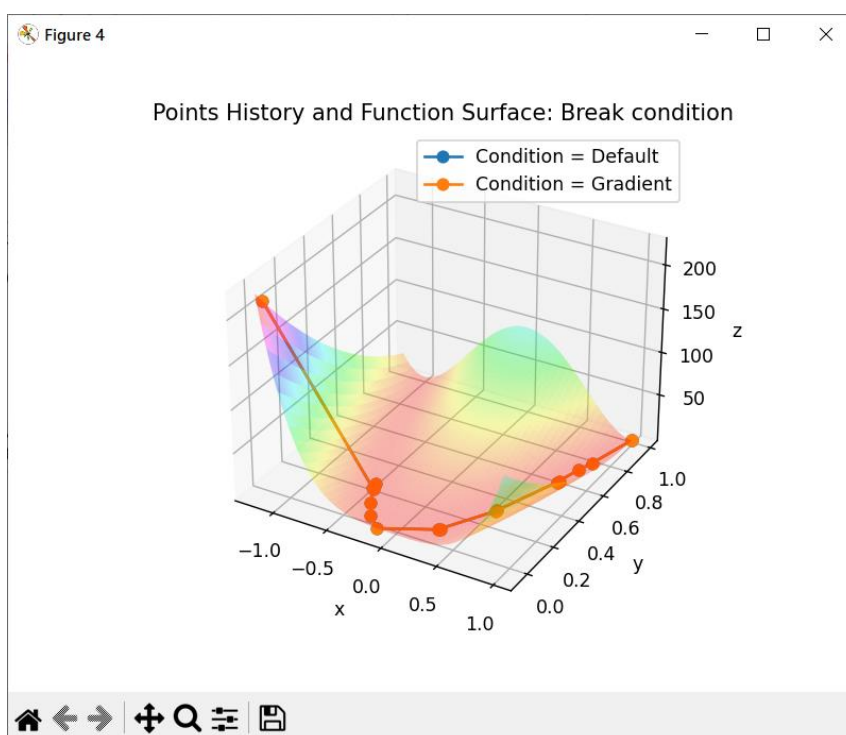


Рисунок 11 - шлях пошуку мінімуму в залежності від обраного критерію закінчення



Рисунок 12 - кількість викликів функції в залежності від обраного критерію закінчення, та відхилення від ідеального розв'язку

Таблиця результатів:

Таблиця 6 - Результат тестування різних варіантів одновимірного пошуку

Критерій	Результат	Кількість викликів функції	Відхилення	F(x)
Звичайний	[0.99381371 0.98704685]	396	0.0009525556292209975	7.656691055730527e-05
Градiєнт	[0.99381371 0.98704685]	393	0.0009525556292209975	7.656691055730527e-05

Результати є майже однаковими, хіба що використання методу з порівнянням значення градієнту дає на 3 (!) виклики функції менше.

3.1.7. Визначення необхідності наявності рестартів

Під час попередніх досліджень, рестарт не відбувся жодного разу. Умовою початку рестарту є невиконання обраного критерію закінчення та напрямок пошуку = 0, або знаменник з алгоритму Пірсона = 0, або кількість ітерацій більша за максимальну. Рестартом називаємо скидання матриці A до початкового значення та, відповідно, перерахунок напрямку руху. Однак новим початковим значенням x буде останнє обчислене значення перед початком рестарту.

Єдиним варіантом дослідження є: прибрати критерії виходу та замінити їх на перевірку кількості здійснених рестартів. Протестуємо це на кількості рестартів у 1, 10, 100, 1000 та 10000.

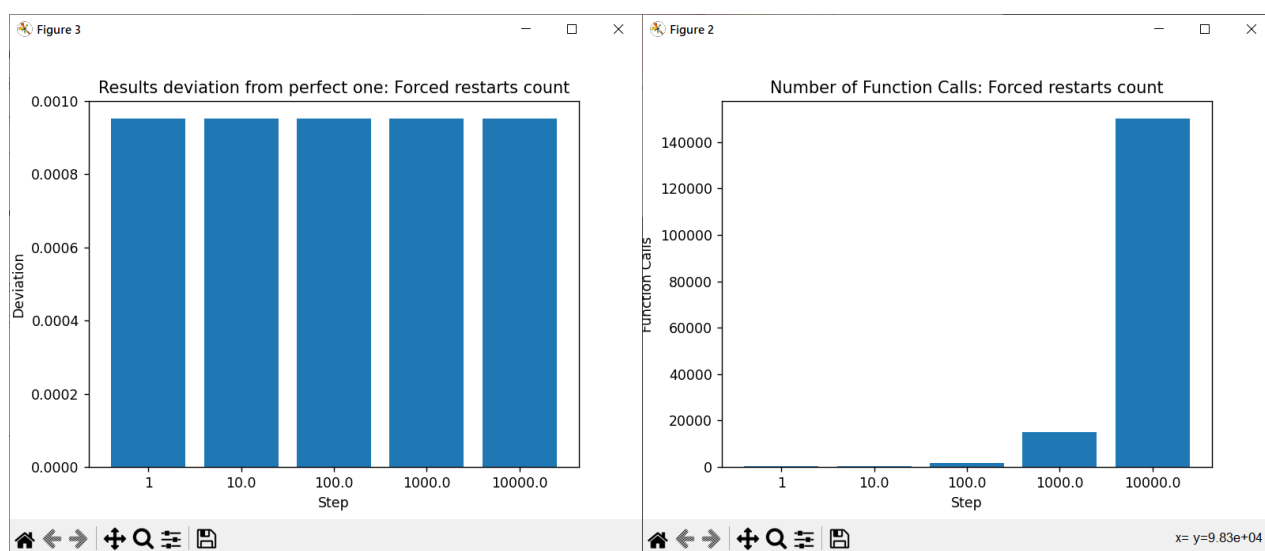


Рисунок 13 - кількість викликів функції в залежності від кількості рестартів, та відхилення від ідеального розв'язку

Результати з точки зору точності є абсолютно ідентичними, кількість же викликів функції поступово зростає в залежності від кількості рестартів.

3.2. Умовна оптимізація

3.2.1. Метод штрафних функцій в залежності від розташування локального мінімуму

Нехай допустима область задається наступним рівнянням:

$$4 - x_1^2 - x_2^2 \geq 0$$

Тоді цільова функція з урахуванням штрафу типу квадрату зрізки виглядатиме наступним чином:

$$P(x, R) = (1 - x_1)^2 + 100 * (x_2 - x_1^2)^2 + R(4 - (x_1^2 + x_2^2))^2$$

В такому випадку локальний мінімум розташований всередині допустимої області.

Оскільки досліджується метод зовнішньої точки, початкова точка має також бути розташована поза допустимою областю.

Візьмемо початковою точкою, наприклад, $x^{(0)} = (1.7, 1.7)$. зі зростаючими значеннями R :

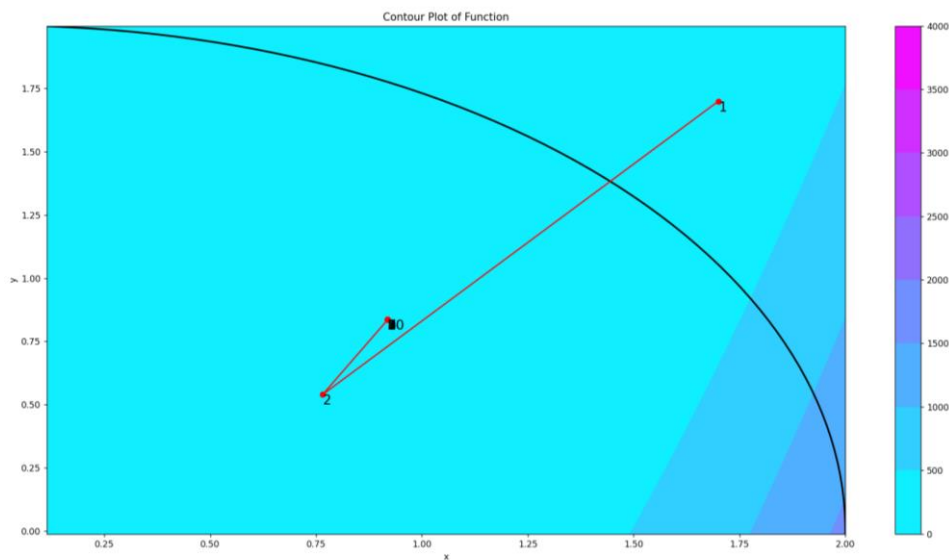


Рисунок 14 - шлях пошуку мінімуму в залежності від значення R . Початкова точка поза допустимою областю. Значення R від 1 до $1e+08$

Таблиця 7 - Результат тестування в залежності від значення R . Початкова точка поза допустимою областю. Значення R від 1 до $1e+08$

R	Результат	Кількість викликів функції	Відхилення	F(x)
1	[0.76580934 0.53994042]	415	0.5162364564838374	0.27128902933094684
10	[0.919196 0.83570803]	105	0.18308778962686673	0.015017694334651818
100	[0.919196 0.83570803]	18	0.18308778962686673	0.015017694334651818
1000	[0.919196 0.83570803]	18	0.18308778962686673	0.015017694334651818
10000	[0.919196 0.83570803]	18	0.18308778962686673	0.015017694334651818
1e+05	[0.919196 0.83570803]	18	0.18308778962686673	0.015017694334651818
1e+06	[0.919196 0.83570803]	18	0.18308778962686673	0.015017694334651818
1e+07	[0.919196 0.83570803]	18	0.18308778962686673	0.015017694334651818
1e+08	[0.919196 0.83570803]	18	0.18308778962686673	0.015017694334651818

Підбиваючи підсумок, у випадку з розташуванням точки локального мінімуму всередині допустимої області, а початкової точки - поза допустимою областю, найкращий результат було отримано за значення $R \geq 1$.

Нехай допустима область задається наступним рівнянням:

$$1 - x_1^2 - x_2^2 \geq 0$$

Тоді цільова функція з урахуванням штрафу типу квадрату зрізки виглядатиме наступним чином:

$$P(x, R) = (1 - x_1)^2 + 100 * (x_2 - x_1^2)^2 + R(1 - (x_1^2 + x_2^2))^2$$

Тоді точка локального мінімуму розташована поза допустимою областю.

Візьмемо початковою точкою $x^{(0)} = (1.7, 1.7)$.

У випадку, коли значення R постійно зростають, пошук відбувається всередині області, отже справжній мінімум знайдено не буде:

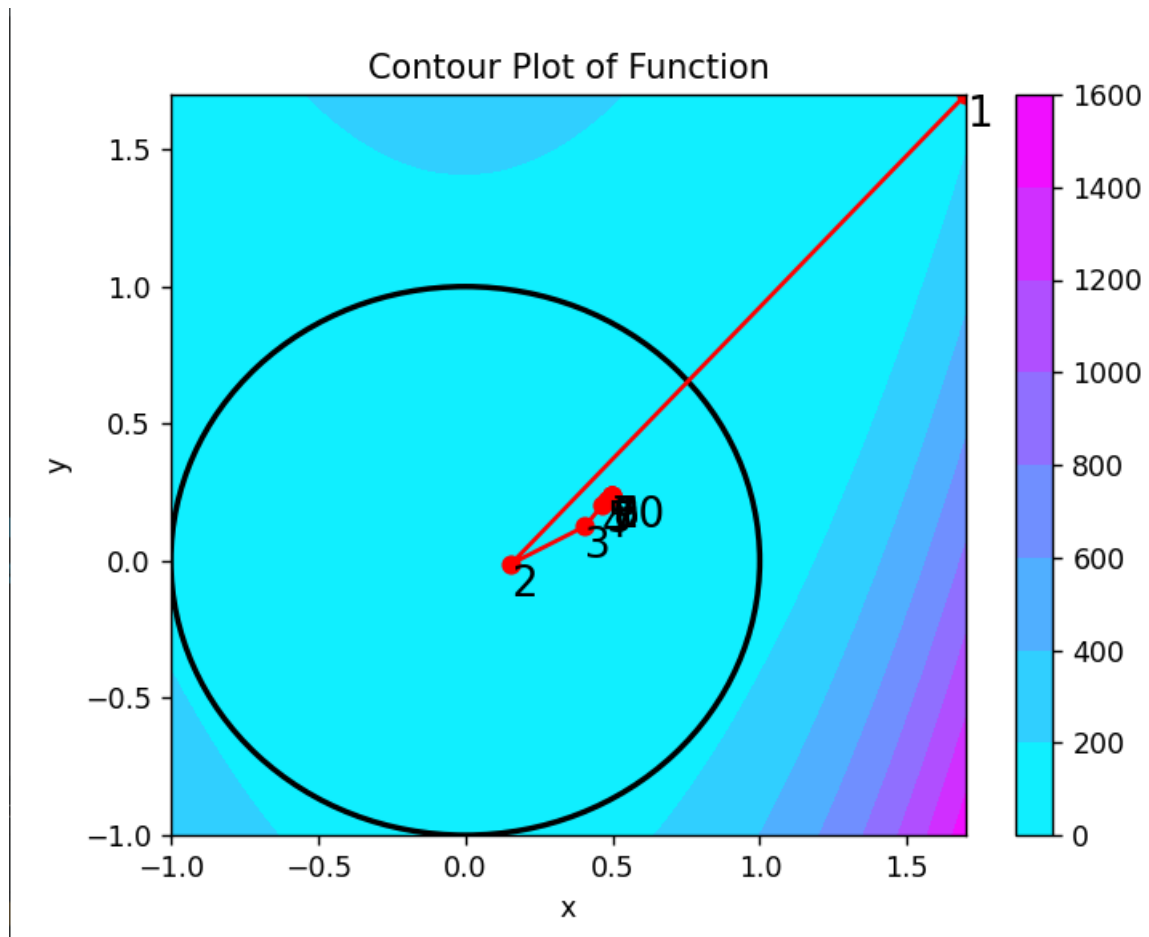


Рисунок 15 - шлях пошуку мінімуму в залежності від значення R . Початкова точка поза допустимою областю. Значення R від 1 до $1e+08$

Таблиця 8 - Результат тестування в залежності від значення R . Початкова точка поза допустимою областю. Значення R від 1 до $1e+08$

R	Результат	Кількість викликів функції	Відхилення	F(x)
1	[0.1554375 -0.0109447]	1129	1.3173059611695257	0.8365255253848294
10	[0.40700282 0.12654332]	217	1.0557330272846817	0.5045890563490031
100	[0.46484953 0.20279299]	93	0.9601692786592931	0.3040539962209005
1000	[0.4791991 0.219545]	396	0.938266267538847	0.28140788886438683
10000	[0.49761613 0.23801054]	68	0.9126979159376047	0.2616272131001298
1e+05	[0.49761613 0.23801054]	18	0.9126979159376047	0.2616272131001298
1e+06	[0.49761613 0.23801054]	18	0.9126979159376047	0.2616272131001298
1e+07	[0.49761613 0.23801054]	18	0.9126979159376047	0.2616272131001298
1e+08	[0.49761613 0.23801054]	18	0.9126979159376047	0.2616272131001298

3.2.2. Метод штрафних функцій в залежності від виду допустимої області (невипукла чи випукла)

Оскільки у п. 3.2.1. Метод штрафних функцій в залежності від розташування локального мінімуму фактично й досліджувались випуклі області, у цьому розділі розглядатимуться лише невивуклі допустимі області.

Візьмемо для прикладу наступне обмеження:

$$\begin{cases} x_1^2 + x_2^2 \geq 1 \\ x_1^2 + x_2^2 \leq 4 \end{cases}$$

За такого обмеження значення мінімуму знаходиться всередині досліджуваної області. Після додавання штрафних функцій цільова функція виглядатиме наступним чином:

$$f(x) + R(x_1^2 + x_2^2 - 1)^2 + R(4 - (x_1^2 + x_2^2))^2$$

За значень R прямуючих до нуля вирішення задачі майже не відрізняється від вирішення задачі БО через зменшення впливу штрафу:

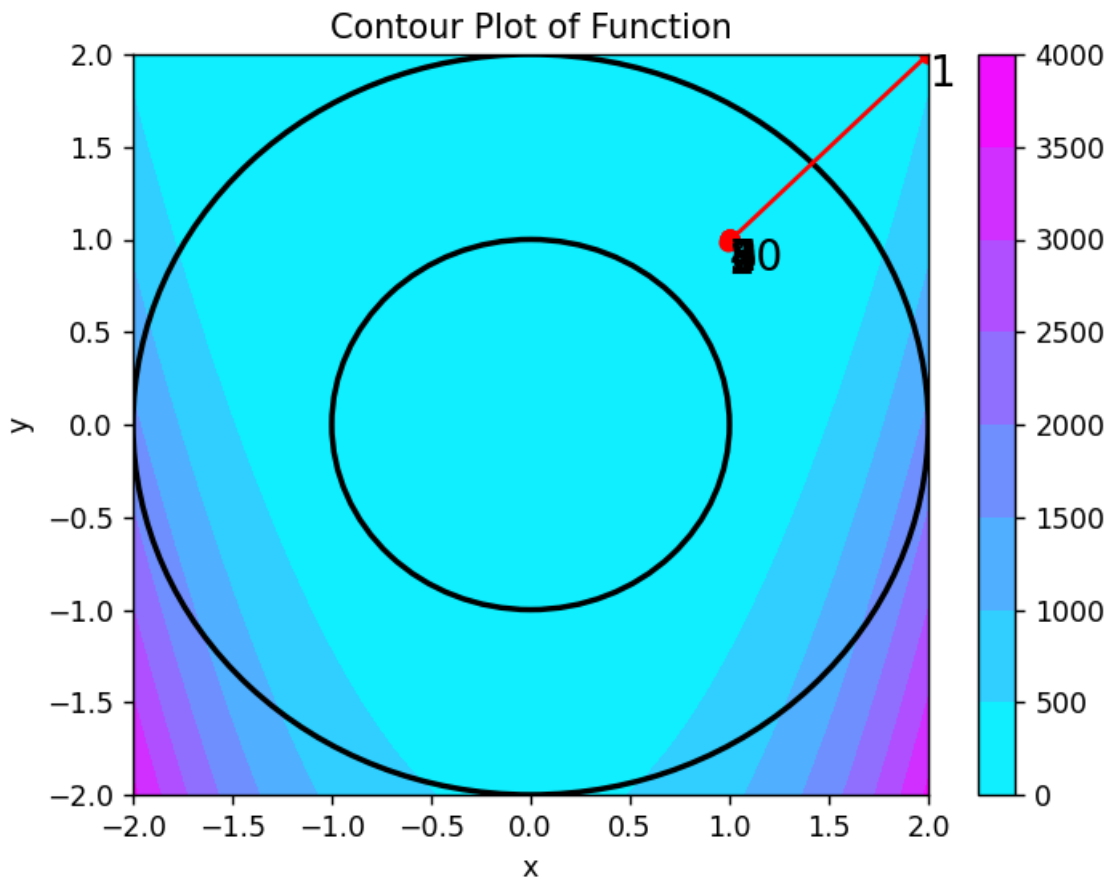


Рисунок 16 - шлях пошуку мінімуму в залежності від значення R . Початкова точка поза допустимою областю. Значення R від 0.1 до $1e-08$

За зростаючих значень R картина буде доволі схожою, адже доволі швидко шукана точка потрапляє до допустимої області. Тоді штраф втрачає власну вагу.

Взявши більш віддалену від мінімуму початкову точку, наприклад,

$$x_0 = (3, 3)$$

Можна побачити, що за великих значень R значення шукаються вздовж границі допустимої області, під кінець звалюючись в бік справжнього мінімуму:

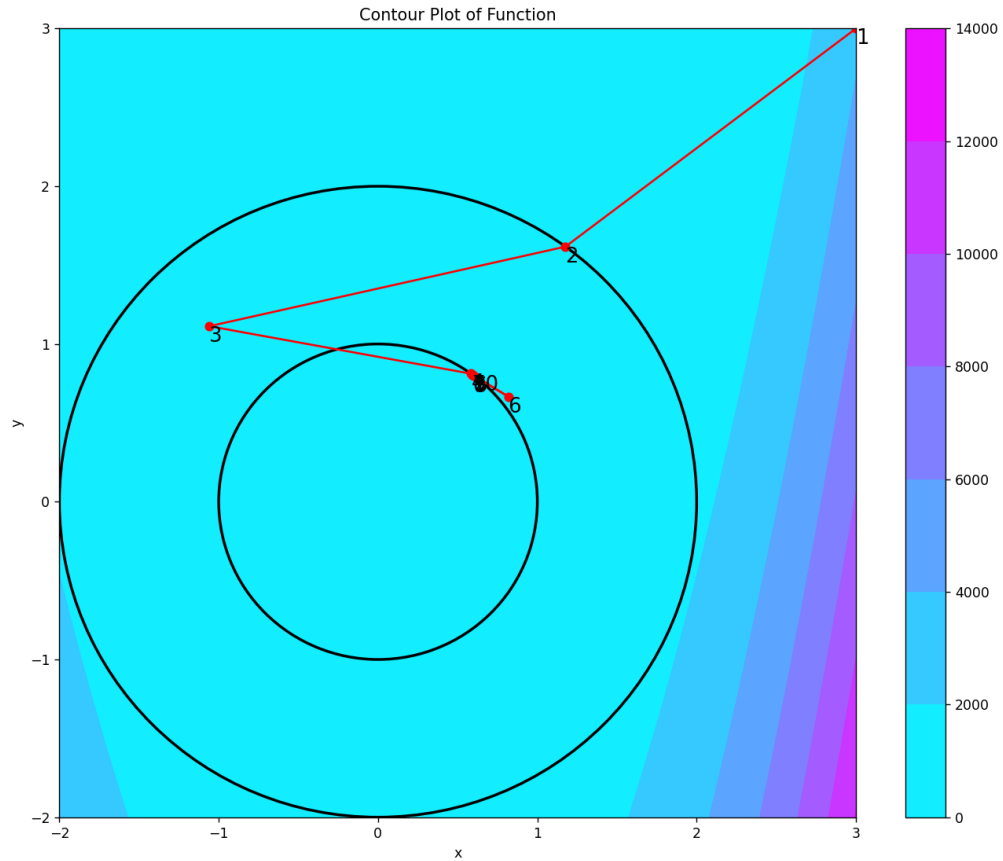


Рисунок 17 - шлях пошуку мінімуму в залежності від значення R . Початкова точка поза допустимою областю. Значення R від 1 до $1e+08$

Таблиця 9

R	Результат	Кількість викликів функції	Відхилення	F(x)
1	[1.17669486 1.61721176]	109	0.6420057826374193	5.441542369441039
10	[-1.05833993 1.11283463]	151	2.061430314024951	4.24201777087583
100	[0.58440315 0.8116889]	190	0.4562694511013784	22.277937989590388
1000	[0.60689718 0.79478711]	81	0.4434435234537494	18.341592557795654
10000	[0.81710816 0.66685309]	151	0.3800477432074451	0.033515465552758764
1e+05	[0.59273039 0.80542429]	98	0.4513626472409924	20.786092776864024
1e+06	[0.59244842 0.8056137]	97	0.4515355237064239	20.83390276271967
1e+07	[0.59240617 0.80564669]	97	0.4515594554810616	20.8414893554086
1e+08	[0.59240855 0.80564454]	97	0.45155823131547024	20.841035348279757

ВИСНОВКИ

У курсовій роботі проведено дослідження збіжності метода Пірсона (алгоритм 2) при мінімізації функції Розенброка. Також вивчено вплив різних факторів на швидкість збіжності та точність отриманого розв'язку.

Нами досліджено параметри, що використовувались в основній формулі та обрано найкращі значення, беручи до уваги кількість ітерацій та точність:

Крок обчислення похідних: 1

Метод обчислення похідних: центральний

Коефіцієнт алгоритму Свенна: 0.01

Точність одновимірного пошуку: 0.01

Алгоритм одновимірного пошуку: золотий перетин

Критерій закінчення: градієнт (другий)

Загалом метод є дуже чутливим до початкової точки, а також вразливим до похибки обчислення матриці A . Однак, метод може бути використаний для знаходження приблизного значення мінімуму функції, у випадку, коли будь-які аналітичні розрахунки, крім обрахунку значення функції, провести неможливо. Варто також зазначити, що не зважаючи на часом низьку точність, метод завжди забезпечував відносно невелику кількість обчислень цільової функції (приблизно 200-300 обчислень в залежності від заданих параметрів та початкової точки).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Методи оптимізації без використання похідних: практикум з дисципліни «Дослідження операцій»[Електронний ресурс]: навч. посіб. для студ. спеціальності 113 «Прикладна математика», спеціалізації «Наука про дані та математичне моделювання» / Т. С. Ладогубець, О. Д. Фіногенов; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 493 Кбайт). – Київ: КПІ ім. Ігоря Сікорського, 2020. – 45 с.
2. Химмельблау Д. Прикладное нелинейное программирование / Химмельблау Д. — М. : Мир, 1975. — 535 с.

ДОДАТКИ

Додаток А. Лістинг коду:

Код для програми, використаної для виконання курсової можна знайти за посиланням: <https://github.com/Retro52/MOCourse> (дата звернення: 30.05.2023)