НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

Кафедра прикладної математики

**КУРСОВА РОБОТА**

з дисципліни “Дослідження операцій”

на тему: МЕТОД НЬЮТОНА

Студента ІІІ курсу, групи КМ-83

напряму підготовки – прикладна математика

ЧЕБАКОВ АНДРІЙ ОЛЕКСІЙОВИЧ

Керівник

ЛАДОГУБЕЦЬ Т. С.

Національна оцінка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_\_\_\_

ECTS \_\_\_\_\_

Київ - 2021

ЗМІСТ

[ВСТУП 2](#_Toc41868080)

[1 Постановка задачі 3](#_Toc41868081)

[2 Теоретична частина 4](#_Toc41868082)

[2.1. Теоретичні основи методу Ньютона 5](#_Toc41868083)

[2.2. Теоретичні основи методів штрафних функцій. Метод зовнішньої точки 6](#_Toc41868084)

[2.3. Приведення матриці Гессе до додатньо визначеного вигляду 8](#_Toc41868085)

[2.4. Методи зовнішніх штрафних функцій 9](#_Toc41868086)

[3 Практична частина 10](#_Toc41868087)

[3.1. Загальні прийоми обчислення похідних у програмному коді 10](#_Toc41868088)

[3.2. Порівняльна характеристика впливу різних параметрів на роботу програми 11](#_Toc41868089)

[3.3. Вплив критерія закінчення на збіжність методу 15](#_Toc41868090)

[3. 3. 1. Норма градієнта 15](#_Toc41868091)

[3.3.2. Абсолютна похибка та відносна похибка 15](#_Toc41868092)

[ВИСНОВОК 16](#_Toc41868093)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 17](#_Toc41868094)

[ДОДАТКИ 18](#_Toc41868095)

# ВСТУП

В ході курсової роботи планується проведення досліджень роботи методу Ньютона мінімізації функцій, а саме як різні параметри згідно постановки задачі впливають на збіжність методу до теоретичного мінімуму.

Метою проведення досліджень у роботі є отримати такий набір параметрів, при якому метод оптимізації дає найкращий результат.

Завдання — провести дослідження збіжності методу, варіюючи при цьому схеми обчислення І-ої та ІІ-ої похідних, величину кроку в різнецевих схемах та три типи критеріїв закінчення. Детальна постановка задачі вказана у розділі 1.

Об’єктом дослідження є метод Ньютона мінімізації функцій, а предметом — функція Розенброка та усі параметри, які впливають на роботу методу.

Метод роботи — пошук та систематизація інформації щодо роботи методу Ньютона, написання програми для спрощення та прискорення проведення обрахунків.

# 1 Постановка задачі

Дослідити збіжність методу Ньютона при мінімізації функції Розенброка в залежності від:

1. Величини кроку h при обчисленні першої та другої похідних.
2. Схеми обчислення першої та другої похідних.
3. Вигляду критерію закінчення. .
4. Способу приведення матриці Гессе до додатньо визначеного виду.

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

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

# 2 Теоретична частина

## 2.1. Теоретичні основи методу Ньютона

Розкладемо цільову функцію в ряд Тейлора, при цьому всі члени порядку 3 і вище відкидаємо — отримаємо квадратичну апроксимацію :

,

де — апроксимуюча функція змінної в точці .

Якщо — напрямок пошуку в методі Ньютона, тоді

Мінімум функції за напрямком визначається диференціюванням за кожною із компонент, після чого отриманий вираз прирівнюється до 0

.

Звідки

,

де — матриця, обернена до матриці Гессе в точці .

Перехід з точки в точку у методі Ньютона здійснюється у вказаний далі спосіб:

.

Напрямок і величина кроку точно визначені. У задачі пошуку мінімуму довільної квадратичної функції з додатньо визначеною матрицею Гессе метод Ньютона дає рішення за одну ітерацію, вибір початкової точки ролі не грає. У разі загальної нелінійної цільової функції метод Ньютона зійдеться до шуканої точки із квадратичною швидкістю при наступних умовах: матриця Гессе функції, що мінімізується в точці повинна бути додатньо визначена, точка початку ітераційного процесу повинна знаходитися досить близько до. Так як метод Ньютона базується на квадратичній апроксимації, він має квадратичну швидкість збіжності (згадану вище), тобто виконується нерівність

,

де стала *С* пов'язана з обумовленістю матриці Гессе. Алгоритму не властиве зменшення значень цільової функції від ітерації до ітерації. Для того, щоб напрямок пошуку був напрямком спуску, повинна виконуватись нерівність

.

Припустимо, що поточне наближення не є стаціонарною точкою (тобто ), і знайдемо проекцію напрямку пошуку за методом Ньютона на напрям, що задається градієнтом в точці

.

У випадку коли матриця Гессе додатньо визначена в точці , наведена умова виконується, отже, напрямок пошуку за методом Ньютона буде напрямком спуску. Якщо в деякій точці від’ємно визначена, то зазначений напрямок є напрямком підйому. У разі невизначеності матриці Гессе до однозначного висновку не приходять. За мінімізації, коли всі власні значення додатні (матриця додатньо визначена), локальна квадратична апроксимація відповідає круговій чи еліптичній впадині, у якій наявний мінімум. Якщо пара власних значень має протилежні знаки (матриця невизначена), квадратична апроксимація являє собою сідло, що не має локального мінімуму. У цьому випадку рух у напрямку пошуку по методу Ньютона приведе у сідлову точку. Критерій, що гарантує збіжність методу Ньютона в припущенні, що функція двічі диференційовна, полягає в тому, що матриця, обернена матриці Гессе цільової функції, повинна бути позитивно визначеною.

## 2.2. Теоретичні основи методів штрафних функцій. Метод зовнішньої точки

Методи штрафних функцій використовуються для вирішення завдань нелінійного програмування. Загальна задача нелінійного програмування виглядає наступним чином:

Мінімізувати

з обмеженнями

Передбачається, що відоме початкове наближення , можливо недопустиме. Будується послідовність , яка починається з і закінчується точкою , що дає найкраще наближення до (рішення задачі) серед усіх точок побудованої послідовності.

У якості беруть стаціонарні точки штрафної функції. Штрафна функція — цільова функція вихідної задачі безумовної оптимізації. За допомогою штрафної функції вихідна задача умовної оптимізації перетворюється в послідовність задач безумовної оптимізації. Методи, засновані на даному підході, визначаються видом штрафної функції, а також правилами, за якими проводиться перерахунок штрафних параметрів на кожному циклі безумовної оптимізації. Штрафна функція, що дозволяє обмежитися рішенням тільки одного завдання безумовної оптимізації, називається точною. При використанні методів штрафних функцій виходить максимальний оптимізуючий ефект за рахунок постійного компромісу між необхідністю виконання обмежень і процесом мінімізації цільової функції , який досягається шляхом присвоєння належних вагів цільової функції і функціям, що задають обмеження. Вплив компонент штрафної функції в процесі оптимізації слабшає, а в границі повністю зникає, так що послідовність проміжних значень штрафної функції сходиться до того ж значення, що і послідовність значень і, отже, їх екстремуми однакові.

Штрафна функція визначається наступним виразом:

 ,

де

R — набір штрафних параметрів;

Ф — штраф, функція від R і функцій, які задають обмеження. Функція Ф визначається таким чином, щоб допустимі точки мали перевагу перед недопустимими щодо безумовної оптимізації штрафної функції.

Методи, засновані на перетворенні задач умовної оптимізації в задачі безумовної оптимізації повинні відповідати таким критеріям:

1) Рішення підзадач повинні збігатися до вирішення вихідної задачі.

.

2) Складність мінімізації повинна бути того ж порядку, що і мінімізації функції .

3) Правило перерахунку штрафних параметрів повинно бути простим (іноді складність перерахунку може бути виправдана за наявності складних функцій).

## 2.3. Приведення матриці Гессе до додатньо визначеного вигляду

Деякі автори запропонували додаткові перетворення, що роблять матрицю Гессе додатньо визначеною на кожному з етапів мінімізації. Грінштадт побудував схему на основі аналізу власних значень, що забезпечує додатню визначеність наближення. Нехай матриця H\*(х) апроксимує Н(х). Масштабуємо матрицю Н\*(х) наступним чином:

де С(х) — діагональна матриця, елементи якої рівні , тобто являють собою додатні квадратні корені із абсолютних значень елементів головної діагоналі Н\*(х). Тоді у матриці П всі елементи головної діагоналі або додатні, або від’ємні.

Оскільки невласні і порядку n, то обернена матриця до їх добутку є добутком їх обернених матриць, взятих у зворотному порядку, або

.

Тоді може бути вирахувана по масштабованій матриці:

.

У своїй роботі Грінштадт вказав, що можна виразити через власні значення і власні вектори матриці П(х). Власні вектори матриці, оберненої до даної,дорівнюють власним векторам вихідної матриці, а власні значення оберненої матриці — просто обернені величини власних значень вихідної матриці . Таким чином,

де — нормований власний вектор, відповідний власному значенню . Потім замість використаємо :

,

де будь яке заміняється малою додатною величиною, так що тепер гарантована додатна визначеність при обрахунку за формулою

## Методи зовнішніх штрафних функцій

Штрафи будуються таким чином, щоб для точок допустимої області вони дорівнювали 0, а для точок поза допустимої області > 0. . При *k*→∞, *R*→∞. В якості штрафів використовується штраф типу квадрат зрізки:

.

Якщо  - неперервні та диференційовані функції, то и Φ - також неперервна и диференційована функція, тому для пошуку мінімума  можно використовувати градієнтні. Алгоритм рішеня є аналогом алгоритму метода внутрішньої точки, тільки в 1) обирается монотонно зростаюча послідовність →∞.

# 3 Практична частина

## 3.1. Загальні прийоми обчислення похідних у програмному коді

Метод Ньютона у ході ітераційного процесу вимагає обчислення вектора градієнта та матриці Гессе, що базуються на обрахунку похідних. Похідні, в свою чергу, можуть бути обраховані як чисельно, так і засобами бібліотек мови Python.

Чисельні методи обчислення похідних вимагають знання точки, у якій проводиться обчислення та кроку для обрахунку однієї з трьох різнецевих схем. Використана в якості «скелета» бібліотека мови Python перетворює роботу із похідними на роботу із символьними змінними, таким чином роблячи програму частково автоматизованою. Зупинимося більш детально на символьному типі.

Спочатку імпортується пакет для роботи із символами конструкцією from sympy import Symbol. Далі наведена нижче конструкція задає символьні змінні х1 та х2 і створює функцію у, від якої далі братиметься похідна за допомогою методу diff, який застосовується до у.

x1 = Symbol('x1')

x2 = Symbol('x2')

y = (((1 - x1) \*\* 2) + (100 \* (x2 - (x1 \*\* 2)) \*\* 2))

x1d = y.diff(x1)

x2d = y.diff(x2)

Вивід програми у відповідь на наведені команди зображена на рис. 1.



Рисунок 1. Вивід похідної функції символьного типу

Отримані коефіцієнти виписуються у програмний код і згодом будуть використані для обрахунку градієнта в точці — це і є часткова автоматизація процесу. Отримуємо результат, записуємо його в код.

Так само можна сформувати матрицю Гессе, застосувавши diff вже до x1d та x2d.

Проте за постановкою задачі необхідно проаналізувати поведінку методу в залежності від схеми обчислення похідних (першої та другої). Тож було програмно реалізовано функції, що обраховують різнецеві схеми(права, ліваб центральна). Аргументи — значення в координаті х1, координаті х2 та змінна, за якою береться похідна. Програмна реалізація виглядає так:

def dev\_left\_func(x1, x2, s):

if s == 'x':

fs\_x = (func\_rosenbroks(x1, x2) - func\_rosenbroks(x1 - h, x2)) / h

elif s == 'y':

fs\_x = (func\_rosenbroks(x1, x2) - func\_rosenbroks(x1, x2 - h)) / h

else:

print('we have trouble')

return fs\_x

def dev\_right\_func(x1, x2, s):

if s == 'x':

fs\_x = (func\_rosenbroks(x1 + h, x2) - func\_rosenbroks(x1, x2)) / h

elif s == 'y':

fs\_x = (func\_rosenbroks(x1, x2 + h) - func\_rosenbroks(x1, x2)) / h

else:

print('we have trouble')

return fs\_x

def dev\_midle\_func(x1, x2, s):

if s == 'x':

fs\_x = (func\_rosenbroks(x1 + h, x2) - func\_rosenbroks(x1 - h, x2)) / (2 \* h)

elif s == 'y':

fs\_x = (func\_rosenbroks(x1, x2 + h) - func\_rosenbroks(x1, x2 - h)) / (2 \* h)

else:

print('we have trouble')

return fs\_x

Оскільки чисельна апроксимація другої похідної дає значення чистих похідних у точці, для обчислення мішаних було використано роботу із символьними функціями.

## 3.2. Порівняльна характеристика впливу різних параметрів на роботу програми

Подивимося, як поводить себе програма, отримуючи на вхід різні дані: схеми обчислення похідних та крок. Також у програмі встановлюємо лічильник, що накопичуватиме кількість обчислень значень цільової функції. У порівняння включимо і додаткову інформацію. Порівняльну характеристику різних комбінацій параметрів можна подивитись у табл.1.

*Таблиця 1.* **Результати запуску програми для бузумовної оптимізації**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Схема** | **Крок** | **Критерій зупинки та точність** | **К-сть ітерацій** | **К-сть обчислень цільової функції** | **Додатково** |
| Центральна | 0.000097 | е<0.001  abs and real | 6 | 96 | New X = [0.9999981230796285, 0.9999962461627785] |
| Центральна | 0.00001 | e<0.001  abs and rel | 6 | 96 | New X = [0.9999999796739858, 0.9999999593479691] |
| Центральна | 0.0001 | e<0.001  abs and rel | 6 | 96 | New X = [0.9999980052141819, 0.9999960104323411] |
| Центральна | 0.0005 | e<0.001  norm grad | 7 | 115 | New X = [0.9999500024929464, 0.9999000074856244] |
| Центральна | 0.001 | e<0.001  norm grad | 7 | 115 | New X = [0.9998000398852487, 0.9996001197542598] |
| Ліва | 0.0005 | e<0.001  norm grad | 9 | 153 | New X = [1.2058355866515997, 1.4542894380237057] |
| Ліва | 0.0001 | e<0.001  norm grad | 7 | 115 | New X = [1.0316466742572359, 1.0643448547745895] |
|  |  |  |  |  |  |
| Права | 0.00001 | e<0.001  norm grad | 7 | 115 | New X = [0.9970098635907303, 0.9940236679739325] |
| Права | 0.0005 | e<0.001  norm grad | 10 | 172 | New X = [0.8785801457711826, 0.7716530714173243] |
| Права | 0.0001 | e<0.001  norm grad | 7 | 115 | New X = [0.9713556875599274, 0.9434818324003665] |
| Права | 0.005 | e<0.001  norm grad | 18 | 324 | New X = [0.4976273556938812, 0.24513288365795807] |
| Права | 0.00001 | e<0.0001  norm grad | 7 | 115 | Absolute error = 6.626694831739814e-08  Relative error = 100.74648909134154 %  New X = [0.9970098635907303, 0.9940236679739325] |
| Центральна | 0.00001 | e<0.0001  norm grad | 7 | 115 | New X = [0.9999999799999765, 0.9999999599999534] |
| Ліва | 0.0001 | e<0.0001  norm grad | 7 | 115 | Absolute error = 6.036556179714122e-08  Relative error = 100.66607657367439 %  New X = [1.0030200532815536, 1.0060542271847412] |

Найкраще показала себе центральна схема обчислення похідних із кроком 0.00001 та критерієм зупину norm grad <0.0001.

Тепер розглянемо умовну оптимізації. Для початку будемо розглядати випуклу область. Наша функція:

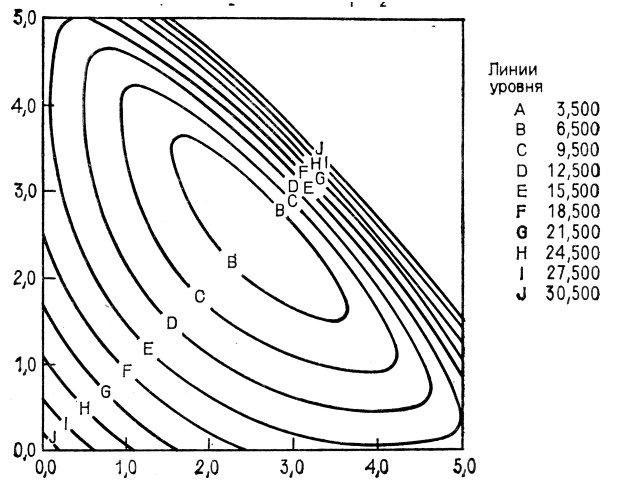
((1 - x1) \*\* 2) + (100 \* (x2 - (x1 \*\* 2)) \*\* 2) + R \* (1 - (x1 \*\* 2) - (x2 \*\* 2))

|  |  |
| --- | --- |
| **R** | **Точка** |
| 0.1 | X = [0.865717502130223, 0.749001707846897] |
| 1 | X = [0.801510094966007, 0.641722469423854] |
| 2 | X = [0.794454726571642, 0.630436866951462] |
| 3 | X = [0.791897491372868, 0.626370954710029] |
| 4 | X = [0.790575060442125, 0.624273467685579] |
| 5 | X = [0.789766785545381, 0.622993197774988] |
| 6 | X = [0.789221564096125, 0.622130330304443] |
| 7 | X = [0.788828934406059, 0.621509322802078] |
| 8 | X = [0.788532694289109, 0.621040975117976] |
| 9 | X = [0.788301226220116, 0.620675152460733] |
| 10 | X = [0.788115378876008, 0.620381508432737] |
| 11 | X = [0.787962874058760, 0.620140598138266] |
| 12 | X = [0.787835477075405, 0.619939386092382] |
| 13 | X = [0.787727458754618, 0.619768806327296] |
| 14 | X = [0.787634709846609, 0.619622358249984] |
| 100 | X = [0.786587458069243, 0.617969969297114] |
| 1000 | X= [0.786432407792303, 0.617725512205793] |
|  |  |

Тепер розглянемо умовну оптимізації. Для початку будемо розглядати невипуклу область. Наша функція:

((1 - x1) \*\* 2) + (100 \* (x2 - (x1 \*\* 2)) \*\* 2) + R \* (1 - (x1 \*\* 2) - (x2 \*\* 2)) +R \* ((x1 \*\* 2) + (x2 \*\* 2)-0,5)\*\*2

|  |  |
| --- | --- |
| **R** | **Точка** |
| 0.1 | X = [0.795755328617448, 0.632509795107829] |
| 1 | X = [0.723588239453218, 0.522603827107901] |
| 10 | X = [0.709272833358622, 0.502041128112582] |
| 100 | X = [0.707654222832219, 0.499741966891846] |
| 1000 | X = [0.707490132737848, 0.499509177356979] |
| 1000 | X = [0.707473700983785, 0.499485869145236] |



## 3.3. Вплив критерія закінчення на збіжність методу

### 3. 3. 1. Норма градієнта

За дослідження поведінки норми градієнта було виявлено, що не завжди напрям мінімізації функції є напрямом спуску. Продемонструємо це налаштувавши програму на роботу із методом diff, оскільки так у нас скінченне число ітерацій. Норма градієнта змінюється так: 752.8, 4.4, 2058.4, 0.05, 0.28, 8.007914331642496e-10. Градієнт не зменшується монотонно — є підйоми. Це пов’язане із тим, що не всі напрямки є напрямками спуску.

### 3.3.2. Абсолютна похибка та відносна похибка

Відносна похибка в рамках реалізації алгоритму метода Ньютона веде себе досить неоднозначно: переважно зменшується від ітерації до ітерації. А на кроці, де метод приходить до теоретичного мінімума становить близько 100%.

Абсолютна похибка в рамках реалізації алгоритму метода Ньютона веде себе так : різко зменшується в перших ітераціях. А на кроці, де метод приходить до теоретичного мінімума становить 1.6031672985731321e-19.

# ВИСНОВОК

Я провів дослідження у рамках курсової роботи метода Ньютона для мінімізації функції Розенброка.

Було виявлено, що є два способи знайти похідні у програмах на Python: чисельні методи апроксимації похідних та засоби стандартних символьних бібліотек (дають напівавтоматизовану програму). Найкращий результат досягнуто із застосуванням центральної схеми із кроком h = 0.0001, критерієм зупину «Норма градієнта менше 0.0001»: 7 ітерацій, 115 обрахунків значень цільової функції і точка, що відхиляється від теоретичного мінімуму на [2.0000023526201005e-08, 4.00000466083128e-08]. А метод diff символьної бібліотеки дає відповідь, рівну теоретичному мінімуму, вже на шостій ітерації. Ймовірно, це відбувається через те, що з наведеним методом ми маємо змогу отримати точний аналітичний вираз для похідної, а вся похибка, що при цьому накопичується, це похибка вхідних даних, на відміну від різнецевих схем.

Найкращими критеріями закінчення виявились норма градієнта, але за умови, що ми перейшли через стрибок: за цього викиду різко зростає як норма градієнта, так і абсолютна похибка. Відбувається це тому, що не всі напрямки мінімізації функції в ході виконання досліджень були напрямками спуску.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

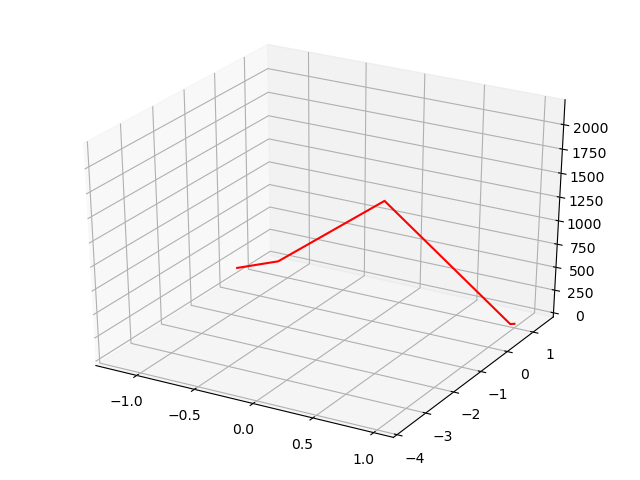
1. В. Реклейтис «Оптимизация в технике» часть 1, М .; «Мир», 1986

2. Д. Химмельблау. «Прикладное нелинейное программирование». М .; «Мир», 1975.

3. Т. С. Ладогубец. «Методы оптимизации» - электронный конспект. Киев.

# ДОДАТКИ

Додаток. Візуалізація результатів



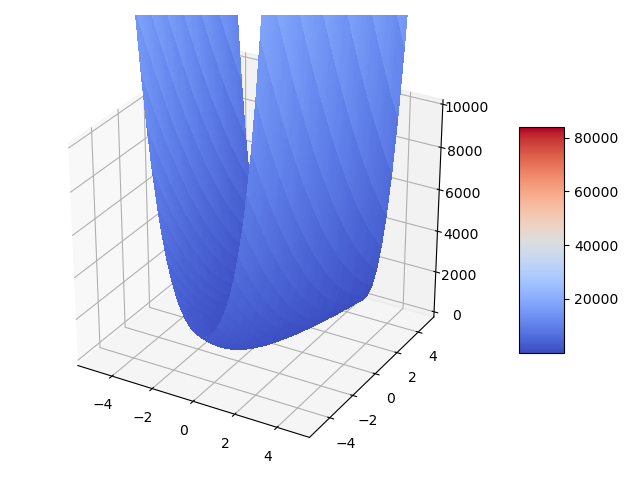


Рисунок Б.1. Функція Розенброка і траєкторія руху