Одеський національний університет імені І. І. Мечникова

Факультет математики, фізики та інформаційних технологій

Кафедра оптимального керування і економічної кібернетики

Кваліфікаційна робота

на здобуття ступеня вищої освіти «бакалавр»

«Аналіз критеріїв зупинки для методів мінімізації функцій»

«Analysis of stop criteria for methods of minimizing functions»

Виконав: здобувач денної форми навчання
спеціальності 113 Прикладна математика
Освітня програма «Прикладна математика»

Красніков Владислав Олександрович

Керівник: кандидат фіз.-мат. наук, доцент
Яровий Анатолій Трохимович
Рецензент: кандидат фіз.-мат. наук, доцент
Васильєв Олександр Борисович

Рекомендовано до захисту:

Протокол засідання кафедри

№ ___ від _____________2023 р.

Завідувач кафедри

_________ ____________________

(підпис) (прізвище, ініціали)

Захищено на засіданні ЕК №_______

протокол № ____ від ________ 2023 р.

Оцінка_____________/___ ___/_____

(за національною шкалою, шкалою ECTS, бали)

Голова ЕК

___________ _______________________

(підпис) (прізвище, ініціали)

Одеса – 2023

ЗМІСТ

ВСТУП

Оптимізація - це одна з найважливіших областей сучасної математики, яка має величезну кількість застосувань у науці та індустрії. Вона займається розв'язанням проблеми пошуку найкращого (оптимального) рішення з усіх можливих рішень. Цей процес часто включає мінімізацію або максимізацію певної функції.

При мінімізації функцій, важливим аспектом є вибір критеріїв зупинки, які визначають, коли процес оптимізації слід зупинити. Застосування правильного критерію зупинки важливе для забезпечення ефективності оптимізації і уникнення зайвих обчислень.

Дипломна робота зосереджується на аналізі різних критеріїв зупинки, які застосовуються в методах мінімізації функцій. Робота має наступну структуру: у першому розділі оглядаються основні методи мінімізації функцій. У другому розділі розглядаються 3 критерії зупинки для перевірки. У третьому розділі пропонується серія числових експериментів, які демонструють ефективність різних критеріїв зупинки.

ОСНОВНА ЧАСТИНА

1.1 Методи мінімізації функції

Методи мінімізації функцій відіграють ключову роль в численних наукових дисциплінах та практичних застосуваннях, включаючи машинне навчання, фінанси, медицину, економіку, фізику та інженерію. Їх мета полягає в пошуку значень вхідних змінних, які мінімізують (або максимізують) вихідну функцію, яка визначає проблему оптимізації.

У відповідності до вимог до похідних функції та до її структури, методи мінімізації функцій можуть бути розділені на різні категорії, такі як методи нульового порядку (які не використовують похідні), методи першого порядку (які використовують перші похідні або градієнти) і методи другого порядку (які використовують другі похідні або матриці Гессе).

Прикладами методів першого порядку є метод спуску градієнта, метод спряжених градієнтів і методи змінної метрики. До методів другого порядку відносяться Метод Ньютона і його модифікації, які використовують наближення других похідних, такі як квазі-Ньютонові методи.

Важливо відмітити, що обрання відповідного методу мінімізації функцій залежить від конкретної задачі оптимізації: структури функції, наявності похідних, розміру проблеми та інших факторів. Крім того, усі ці методи пов'язані з вибором відповідного критерію зупинки, який визначає, коли процес оптимізації можна вважати завершеним.

Як вже було зазначено, методи мінімізації функцій можна класифікувати за різними критеріями. Один з таких критеріїв - це кількість змінних, які оптимізуються. У випадку одновимірної оптимізації мінімізація функції виконується за однієї змінної, тоді як багатовимірна оптимізація включає пошук мінімуму функції, що залежить від двох або більше змінних.

Інший критерій - це використання похідних. Методи першого порядку використовують градієнт функції, тобто вектор її перших похідних, для визначення напрямку пошуку. Методи другого порядку, також відомі як методи Ньютона, використовують інформацію про кривизну функції, що отримана з її других похідних, для прискорення процесу оптимізації.

Додатково варто відмітити, що деякі методи оптимізації можуть бути адаптовані для рішення задач з обмеженнями, де деякі або всі змінні мають бути в межах вказаних обмежень. Ці методи включають методи штрафів та методи бар'єрів, які модифікують цільову функцію таким чином, що мінімуми, які не задовольняють обмеженням, стають менш привабливими або неможливими.

1.1.1 **Метод спряжених градієнтів**

Метод спряжених градієнтів є одним з найпопулярніших методів оптимізації, особливо для великих задач. Він використовується у багатьох областях, включаючи машинне навчання, фізику, інженерію та комп'ютерний дизайн.

Основний принцип методу спряжених градієнтів полягає у використанні градієнтів функції, яка має бути мінімізована, для визначення напрямку, в якому слід рухатися для досягнення мінімуму. Однак, на відміну від простого методу градієнтного спуску, метод спряжених градієнтів використовує інформацію з попередніх кроків для визначення наступного напрямку.

Основні кроки методу спряжених градієнтів включають:

1. Вибір початкової точки та визначення напрямку пошуку, як правило, від'ємного градієнта функції в початковій точці.
2. Проведення пошуку вздовж визначеного напрямку для визначення оптимального кроку, який мінімізує значення функції.
3. Обчислення нового градієнта функції в оновленій точці.
4. Визначення нового напрямку пошуку, який є спряженим з попередніми напрямками.

Ці кроки повторюються, поки не буде досягнуто бажаного мінімуму або поки не буде виконано певне число ітерацій.

Однією з ключових переваг методу спряжених градієнтів є його ефективність для великих задач оптимізації, які можуть бути важкими для інших методів через обчислювальні вимоги. Він також добре підходить для оптимізації задач, де функція має складну форму або немонотонний профіль. Метод спряжених градієнтів здатний ефективно знаходити мінімум в таких випадках, оскільки він комбінує інформацію з попередніх ітерацій.

Крім того, метод спряжених градієнтів має деякі інші переваги. Він не вимагає зберігання повної матриці Гессе або інших великих матриць, що робить його практичним для використання в задачах з великою кількістю змінних. Крім того, цей метод є ітеративним, тому його можна застосовувати для поступового покращення розв'язку з кожною ітерацією.

Незважаючи на свою ефективність, метод спряжених градієнтів також має свої обмеження. Він не гарантує знаходження глобального мінімуму, а лише збігається до локального мінімуму функції. Крім того, метод може бути чутливим до вибору початкової точки та параметрів, і неправильний вибір може призвести до повільної збіжності або незбіжності.

Узагальнюючи, метод спряжених градієнтів є потужним і популярним методом оптимізації, який знайшов широке застосування в різних галузях. Він поєднує ефективність ітераційних методів з використанням інформації з попередніх кроків, що робить його особливо корисним для великих задач оптимізації з складними функціями.

Приклад коду для методу спряжених градієнтів є у Додатку А.

1.1.2 Метод Ньютона

Метод Ньютона є одним з найпоширеніших методів оптимізації, який базується на використанні інформації про другу похідну функції для знаходження мінімуму. Цей метод отримав свою назву на честь великого фізика та математика Ісаака Ньютона, який вперше розробив його.

Основна ідея методу Ньютона полягає в тому, що він апроксимує функцію другим порядком у околі точці. Матриця других похідних, також відома як матриця Гессе, використовується для побудови цієї квадратичної апроксимації. Потім знаходяться точки екстремуму цієї апроксимації, які є кандидатами на мінімум функції.

Щоб більш детально описати метод Ньютона, розглянемо основні кроки:

1. Вибір початкової точки: Початкова точка обирається в заданому просторі змінних.
2. Обчислення градієнта та матриці Гессе: Обчислюються градієнт та матриця Гессе функції в поточній точці.
3. Розв’язання системи лінійних рівнянь: Застосовуючи матрицю Гессе та градієнт, розв'язується система лінійних рівнянь для знаходження напрямку спуску.
4. Оновлення поточної точки: Переміщення до нової точки, обчисленої шляхом додавання оптимального кроку до поточної точки.
5. Перевірка умови зупинки: Перевіряється умова зупинки, яка може включати досягнення заданої точності або заданої кількості ітерацій.
6. Повторення кроків: Якщо умова зупинки не виконана, повторюються кроки 2-6.

Однією з головних переваг методу Ньютона є його швидкість збіжності. Він може досить швидко знаходити мінімум функції, особливо в тих випадках, коли функція має квадратичну форму.

Однак, метод Ньютона також має свої обмеження. Він може бути обчислювально витратним, оскільки вимагає обчислення інверсії матриці Гессе. Крім того, метод Ньютона може бути нестійким у випадках, коли матриця Гессе є невизначеною або погано умовленою.

Узагальнюючи, метод Ньютона є потужним методом оптимізації, який може швидко знаходити мінімум функції, використовуючи інформацію про другу похідну. Він знайшов широке застосування у багатьох галузях, але його ефективність може залежати від властивостей функції та вибору початкової точки. Приклад коду для методу Ньютона є у Додатку Б.

1.1.3 Метод змінної метрики

Метод змінної метрики (або метод квазі-Ньютона) є іншим ефективним методом оптимізації, який базується на апроксимації матриці Гессе функції. Він розроблений для подолання обчислювальних обмежень методу Ньютона, а саме необхідності обчислювати та інвертувати матрицю Гессе. Метод змінної метрики поєднує швидкість збіжності методу Ньютона з обчислювальною ефективністю.

Основна ідея методу змінної метрики полягає в тому, що він створює апроксимацію матриці Гессе, використовуючи інформацію про градієнти функції в попередніх точках. Ця апроксимація матриці Гессе змінюється з кожною ітерацією, що дозволяє адаптуватися до локальної геометрії функції та покращувати ефективність оптимізації.

Розглянемо основні кроки методу змінної метрики:

1. Вибір початкової точки: Початкова точка обирається в заданому просторі змінних.
2. Ініціалізація матриці апроксимації Гессе: Початкова матриця апроксимації Гессе ініціалізується як одинична матриця або як будь-яка інша початкова матриця.
3. Обчислення градієнта функції: Обчислюється градієнт функції в поточній точці.
4. Розв'язання системи лінійних рівнянь: Застосовуючи матрицю апроксимації Гессе та градієнт, розв'язується система лінійних рівнянь для знаходження напрямку спуску.
5. Обчислення оптимального кроку: Визначається оптимальний крок, який мінімізує значення функції.
6. Оновлення поточної точки: Переміщення до нової точки.
7. Оновлення матриці апроксимації Гессе: Матриця апроксимації Гессе оновлюється з урахуванням різниці між градієнтами функції в поточній і попередніх точках.
8. Перевірка умови зупинки: Перевіряється умова зупинки, яка може включати досягнення заданої точності або заданої кількості ітерацій.
9. Повторення кроків: Якщо умова зупинки не виконана, повторюються кроки 3-8.

Метод змінної метрики має декілька варіацій, таких як метод Бройдена-Флетчера-Гольдфарба-Шанно (BFGS) та метод оберненої матриці Гессе (L-

BFGS), які використовують різні стратегії оновлення матриці апроксимації Гессе.

Узагальнюючи, метод змінної метрики є потужним методом оптимізації, який комбінує ефективність методу Ньютона з обчислювальною ефективністю. Він є популярним і широко використовується для рішення різноманітних задач оптимізації, особливо в тих випадках, коли обчислення матриці Гессе є обтяжливим завданням.

Приклад коду для методу змінної метрики є у Додатку В.

1.2 Критерії зупинки

Критерії зупинки, або критерії зупинення, це ключова частина будь-якої оптимізаційної задачі, але вони особливо важливі в контексті машинного навчання і алгоритмічної торгівлі.

Критерії зупинки визначають, коли алгоритм повинен припинити ітераційний процес. Це важливо, бо допомагає визначити, коли алгоритм досягнув свого оптимального рішення, або коли він продовжує робити ітерації без значимого поліпшення результату. Без визначення цих критеріїв, алгоритм може непотрібно витрачати час і ресурси на додаткові ітерації.

Критерії зупинки можуть бути визначені на основі різних параметрів, включаючи:

1. Максимальну кількість ітерацій: це простий і часто використовуваний критерій зупинки. Якщо алгоритм не досягне прийнятного рішення за певну кількість ітерацій, він зупиняється.
2. Певний рівень точності: алгоритм зупиняється, коли він досягає вказаного рівня точності.
3. Зміна функції втрати: в алгоритмах машинного навчання, критерій зупинки може бути визначений як певна зміна в функції втрати. Якщо функція втрати перестає значно зменшуватися з кожною новою ітерацією, алгоритм може бути зупинений.

Основна мета критеріїв зупинки - забезпечити ефективність алгоритму, зменшуючи непотрібні ітерації, і забезпечити його здатність досягати оптимального рішення. Вони також допомагають запобігти перенавчанню, яке відбувається, коли алгоритм надто добре пристосовується до навчальних даних і стає менш ефективним при генералізації на нових даних.

Важливо враховувати, що правильний вибір критерію зупинки може залежати від контексту задачі та вимог до точності. Вибір критерію зупинки - це баланс між часом обчислень, точністю рішення і ризиком перенавчання.

1.2.1 1-ий критерій зупинки

$$A1.\, f(x^{k-1}) - f(x^k) < \ \tau_f(1 + \left|f(x^k)\right|$$

$$A2.\, \left|\left|x^{k-1} - \ x^k\right|\right| < \sqrt{\tau_f}(1 + \left|\left|x^k\right|\right|)$$

$$A3.\, \left|\left|f'(x^k)\right|\right| \leq \sqrt[3]{\tau_f}(1 + \left|f(x^k)\right|$$

Цей критерій зупинки призначений для гарантії того, що процес оптимізації зупиняється, коли він досягає впевненої точності розв'язку. Він складається з трьох умов:

- Зміна значення цільової функції між двома послідовними ітераціями повинна бути меншою за деяку задану порогову величину, кориговану на поточне значення функції.
- Різниця між двома послідовними рішеннями повинна бути меншою за корінь квадратний від заданої порогової величини, коригований на поточне значення розв’язку.
- Норма градієнта цільової функції повинна бути меншою за кубічний корінь від заданої порогової величини, коригованої на поточне значення функції.

Всі ці умови слід задовольняти одночасно для зупинки ітеративного процесу.

«τ_f» в даному випадку є пороговим значенням, яке контролює точність алгоритму. Змінюючи це значення, можна контролювати швидкість збіжності та точність алгоритму. Зменшення «τ_f» призведе до більш точного розв'язку, але збільшить кількість ітерацій. Збільшення «τ_f» прискорить збіжність, але може призвести до менш точного розв'язку.

1.2.2 2-ий критерій зупинки

$$\left|\vec{x}^{k+1} - \vec{x}^{k}\right| \leq \varepsilon$$

$$\left|F(\vec{x}^{k+1}) - F(\vec{x}^{k})\right| \leq \varepsilon$$

Цей критерій зупинки також використовується в ітеративних методах оптимізації, таких як метод градієнтного спуску, для перевірки збіжності алгоритму. Він складається з двох умов:

Різниця між двома послідовними рішеннями повинна бути меншою або рівною заданій пороговій величині ε. Це означає, що алгоритм повинен зупинитися, коли зміни в рішенні між ітераціями стають незначними.

Різниця між значеннями цільової функції (F) для двох послідовних розв’язків повинна бути меншою або рівною ε. Це означає, що алгоритм повинен зупинитися, коли зміни в цільовій функції між ітераціями стають незначними. '

Цей критерій зупинки гарантує, що алгоритм зупиняється, коли досягає достатньої точності в розв'язку. Значення ε визначає потрібну точність: менше значення ε призводить до більш точного розв'язку, але збільшує кількість ітерацій. Більше значення ε зменшує кількість ітерацій, але може призвести до менш точного розв'язку.

1.2.3 3-ій критерій зупинки

$$\left|F(x^{k-1}) - F(x^k)\right| < 2^{-\tau_f}\ (1 + \left|F(x^{k-1}) - F(x^k)\right|)$$

$$\left|\left|h^k\right|\right| \leq 2^{\frac{-\tau_f}{3}}(1 +\ \left|F(x^{k-1}) - F(x^k)\right|)$$

$$\left|\left|h^k\right|\right| \leq 2^{\frac{-\tau_f}{3}}(1 +\ \left|F(x^k)\right|)$$

Цей критерій зупинки є популярним методом оптимізації для знаходження мінімуму або максимуму цільової функції. Він складається з трьох умов:

1. Різниця між значеннями цільової функції (F) для двох послідовних рішень повинна бути меншою за 2 в ступені «τ_f», помножене на суму одиниці та абсолютне значення різниці значень цільової функції. Це означає, що зміни в цільовій функції між ітераціями повинні бути незначні, але точність контролюється параметром «τ_f».
2. Норма кроку ітерації «h^k» повинна бути меншою або рівною 2 в ступені «τ$\frac{-\tau_f}{3}$», помноженому на суму одиниці та абсолютне значення різниці значень цільової функції. Це гарантує, що розмір кроку ітерації не стає надто великим і дозволяє контролювати швидкість сходимості алгоритму.
3. Норма кроку ітерації також повинна бути меншою або рівною 2 в ступені «$\frac{-\tau_f}{3}$», помноженому на суму одиниці та абсолютне значення значення цільової функції. Це допомагає контролювати величину кроку ітерації в залежності від поточного значення цільової функції.

Всі ці умови повинні бути виконані одночасно, щоб зупинити ітеративний процес.

Значення «τ_f» в даному випадку є пороговими значеннями, які контролюють точність алгоритму. Змінюючи ці значення, можна контролювати швидкість сходимості.

Чисельні розрахунки

$f(x) = 100 \times (x_2 - x_1{}^2)^2 + (1 - x_1)^2$; x^0 = [-1.2;1]							
метод	1ий кр	к-сть іт	2ий кр	к-сть іт	3ий кр	к-сть іт	min
Метод спряжених градієнтів	[1.00000005 1.0000001]	42	[1.00000004 1.00000009]	10	[0.99999678 0.99999355]	37	
Метод Ньютона	[1.00000003 1.00000006]	7	[1.00000002 1.000000016]	7	[1.00000001 1.00000002]	5	[1;1]
Метод змінної метрики	[0.99937275 0.99870949]	11	[0.99999997 0.99999995]	32	[0.99999997 0.99999995]	26	

Таблиця 1

$f(x) = ((x_1{}^2 + x_2 - 11)^2 + (x_1 + x_1{}^2 - 7)^2;$ x^0 = [1;1]							
метод	1ий кр	к-сть іт	2ий кр	к-сть іт	3ий кр	к-сть іт	min
Метод спряжених градієнтів	[2.99999968 1.99999987]	73	[3.00000051 2.00000011]	18	[3. 1.99999999]	7	
Метод Ньютона	[3.3844315 -1.84812481]	7	[3.58442834 -1.84812653]	6	[3.00000003 2.00000002]	6	[3.38443;-1.84813]
Метод змінної метрики	[3.4385793 -2.6422054]	14	[3.2551 -2.5124]	18	[3. 2.]	15	

Таблиця 2

$f(x) = (x_1{}^2 + 12x_2 - 1)^2 + (49x_1{}^2 + 49x_2{}^2 + 84x_1 + 2324x_2 - 681)^2$; $x^0 = [1;1]$							
метод	1ий кр	к-сть іт	2ий кр	к-сть іт	3ий кр	к-сть іт	min
Метод спряжених градієнтів	[0.28580858 0.27936017]	91	[0.28581; 0.27936]	15	[0.2864123 0.27441215]	12	[0.28581;0.27936])
Метод Ньютона	[0.28614928 0.27612755]	9	[0.2851215 0.2766541]	5	[0.2851665 0.2791766]	6	
Метод змінної метрики	[0.28578807 0.27933525]	6	[0.2898151 0.27151162]	5	[0.2895856 0.2754340]	11	

Таблиця 3

$f(x) = (x_1{}^2 + 12x_2 - 1)^2 + (x_1 + x_1{}^2 - 7)^2$; $x^0 = [1;1]$							
метод	1ий кр	к-сть іт	2ий кр	к-сть іт	3ий кр	к-сть іт	min
Метод спряжених градієнтів	[4.471635 -1.586245]	13	[4.4701951 -1.5866314]	10	[4.47835206 -1.5879697]	10	[4.47; -1.58]
Метод Ньютона	[4.476115 -1.5812451]	7	[4.47815112 -1.5860194]	4	[4.4691701 -1.5794414]	51	
Метод змінної метрики	[4.477134 -1.58178501]	21	[4.47835181 -1.58796959]	12	[4.4783518 -1.5879697]	6	

Таблиця 4

$f(x) = (x_2 - x_1{}^2)^2 + (1 - x_1)^2$; x^0 = [-1.2;1]							
метод	1ий кр	к-сть іт	2ий кр	к-сть іт	3ий кр	к-сть іт	min
Метод спряжених градієнтів	[1.00000021 1.00000044]	16	[0.99999995 0.99999989]	15	[0.99999851 0.99999595]	12	[1;1]
Метод Ньютона	[1.00000006 1.00000011]	5	[1.00000001 1.00000002]	5	[1.00000021 1.00000001]	41	
Метод змінної метрики	[1.00000011 1.00000022]	19	[1.00000003 1.00000006]	9	[1.00000009 1.00000017]	21	

Таблиця 5

$f(x) = (x_1 - x_1{}^2)^2 + 100 \times (1 - x_1)^2$; x^0 = [-1.2;1]							
метод	1ий кр	к-сть іт	2ий кр	к-сть іт	3ий кр	к-сть іт	min
Метод спряжених градієнтів	[1.00000007 1.00000014]	35	[0.99999999 1.00000001]	6	[0.99999999 1.00000001]	6	[1;1]
Метод Ньютона	[0.99999446 0.99998888]	33	[1.00000001 1.00000002]	7	[1.00000002 1.00000001]	66	
Метод змінної метрики	[0.99999726 0.99999448]	52	[0.99999999 1.00000004]	6	[1.0000000; 1.00000137]	14	

Таблиця 6

$f(x) = 100 \times (x_2 - x_1{}^3)^2 + (1 - x_1)^2;\ x^0$ = [-1.2;1]							
метод	1ий кр	к-сть іт	2ий кр	к-сть іт	3ий кр	к-сть іт	min
Метод спряжених градієнтів	[0.9999946 1.0000149]	13	[0.99999086 0.99997239]	25	[0.99997838 0.9999351]	24	[1;1]
Метод Ньютона	[0.99999579 0.99999157]	10	[1.00000014 1.00000027]	6	[0.99999995 0.99999999]	51	
Метод змінної метрики	[0.99999955 0.99999909]	69	[0.99999103 0.99997309]	43	[0.99999929 0.99999954]	26	

Таблиця 7

$f(x)$ =$(x_1 + 40x_2)^2 + 5 \times (x_1 - x_2)^2$; x^0 = [-1.2;1]							
метод	1ий кр	к-сть іт	2ий кр	к-сть іт	3ий кр	к-сть іт	min
Метод спряжених градієнтів	[1.00000002 1.00000001]	5	[1.000000031 0.09999997]	16	[1.000000009 0.999999994]	37	[1;1]
Метод Ньютона	[1.000000022 1.000000071]	11	[1.00000001 0.99999998]	5	[1.00000001 1.000000002]	15	
Метод змінної метрики	[1.10000003 1.09999997]	23	[1.000000011 0.999999984]	6	[0.99999997 0.99999995]	26	

Таблиця 8

<table>
<tr><td colspan="8">$f(x)$ = -100 $*$ ($x2$ − ($x1$ − 1)^3)^2 + (2 − $x1$)^2 ; x^0 =[-1.2;1]</td></tr>
<tr><td>метод</td><td>1ий кр</td><td>к-сть іт</td><td>2ий кр</td><td>к-сть іт</td><td>3ий кр</td><td>к-сть іт</td><td>min</td></tr>
<tr><td>Метод спряжених градієнтів</td><td>[1.00000015 1.0000002]</td><td>42</td><td>[1.00000005 1.00000009]</td><td>10</td><td>[0.99999918 0.99999125]</td><td>16</td><td rowspan="3">[1;1]</td></tr>
<tr><td>Метод Ньютона</td><td>[1.00000003 1.00000006]</td><td>7</td><td>[1.000000021 1.000000016]</td><td>7</td><td>[1.0000001 1.00000001]</td><td>9</td></tr>
<tr><td>Метод змінної метрики</td><td>[1.00000009 1.00000001]</td><td>11</td><td>[1.00000012 1.00000031]</td><td>32</td><td>[0.99999999 0.99999995]</td><td>21</td></tr>
</table>

Таблиця 9

Евклідова відстань

<table>
<tr><td colspan="8">$f(x) = 100 \times (x_2 - {x_1}^2)^2 + (1 - x_1)^2$; x^0 = [-1.2;1]</td></tr>
<tr><td>метод</td><td>е.в.1ий кр</td><td>ср к-сть іт</td><td>е.в.2 ий кр</td><td>ср к-сть іт</td><td>е.в.3ий кр</td><td>ср к-сть іт</td><td>min</td></tr>
<tr><td>Метод спряжених градієнтів</td><td>0.0000141</td><td rowspan="3">20</td><td>0.0000131</td><td rowspan="3">16.3</td><td>0.0000138</td><td rowspan="3">22.6</td><td rowspan="3">[1;1]</td></tr>
<tr><td>Метод Ньютона</td><td>0.000085</td><td>0.000064</td><td>0.000014</td></tr>
<tr><td>Метод змінної метрики</td><td>0.001411</td><td>0.000050</td><td>0.000050</td></tr>
</table>

Таблиця 10

$f(x) = ((x_1{}^2 + x_2 - 11)^2 + (x_1 + x_1{}^2 - 7)^2;\ x^0 = [1;1]$							
метод	е.в.1ий кр	ср к-сть іт	е.в.2 ий кр	ср к-сть іт	е.в.3ий кр	ср к-сть іт	min
Метод спряжених градієнтів	0.4133414	31.3	0.413340734	14	0.41334116	9.3	[3.38443;-1.84813]
Метод Ньютона	0.4024112		0.199998340		0.413341		
Метод змінної метрики	0.7959195		0.676742847		0.413341		

Таблиця 11

$f(x) = (x_1{}^2 + 12x_2 - 1)^2 + (49x_1{}^2 + 49x_2{}^2 + 84x_1 + 2324x_2 - 681)^2\ ;\ x^0 = [1;1]$							
метод	е.в.1ий кр	ср к-сть іт	е.в.2ий кр	ср к-сть іт	е.в.3ий кр	ср к-сть іт	min
Метод спряжених градієнтів	0.00000143	35.3	0.001512792	8.1	0.004984374	9.6	[0.28581;0.27936]
Метод Ньютона	0.330679210		0.002792		0.0019578		
Метод змінної метрики	0.000033067		0.0088112		0.0054468		

Таблиця 12

$f(x) = (x_1{}^2 + 12x_2 - 1)^2 + (x_1 + x_1{}^2 - 7)^2$; x^0 = [1;1]							
метод	е.в.1ий кр	ср к-сть іт	е.в.2 ий кр	ср к-сть іт	е.в.3ий кр	ср к-сть іт	min
Метод спряжених градієнтів	0.00645548	13.6	0.0066342	8.6	0.0115443	22.3	[4.47; -1.58]
Метод Ньютона	0.00624047		0.010132		0.00176512		
Метод змінної метрики	0.00735392		0.0115441		0.0115442		

Таблиця 13

$f(x) = (x_2 - x_1{}^2)^2 + (1 - x_1)^2$; x^0 = [-1.2;1]							
метод	е.в.1ий кр	ср к-сть іт	е.в.2 ий кр	ср к-сть іт	е.в.3ий кр	ср к-сть іт	min
Метод спряжених градієнтів	0.0000004875	13.3	0.00000012	9.6	0.00000431	24.6	[1;1]
Метод Ньютона	0.000000125		0.00000002		0.000000001		
Метод змінної метрики	0.000000245		0.00000006		0.00000019		

Таблиця 14

$f(x) = (x_1 - x_1{}^2)^2 + 100 \times (1 - x_1)^2$; x^0 = [-1.2;1]							
метод	е.в.1ий кр	ср к-сть іт	е.в.2 ий кр	ср к-сть іт	е.в.3ий кр	ср к-сть іт	min
Метод спряжених градієнтів	0.00000015652	40	0.000000014	6.3	0.0000000141	28.6	[1;1]
Метод Ньютона	0.00001242		0.000000022		0.00000017		
Метод змінної метрики	0.00000616		0.00000134		0.0000013		

Таблиця 15

$f(x) = 100 \times (x_2 - x_1{}^3)^2 + (1 - x_1)^2$; x^0 = [-1.2;1]							
метод	е.в.1ий кр	ср к-сть іт	е.в.2 ий кр	ср к-сть іт	е.в.3ий кр	ср к-сть іт	min
Метод спряжених градієнтів	0.000015848	30.6	0.0000290	24.6	0.0000684	33.6	[1;1]
Метод Ньютона	0.000009422		0.000000304		0.000000014		
Метод змінної метрики	0.000001015		0.000028		0.000000845		

Таблиця 16

$f(x)$ =$(x_1 + 40x_2)^2 + 5 \times (x_1 - x_2)^2$; x^0 = [-1.2;1]							
метод	е.в.1ий кр	ср к-сть іт	е.в.2 ий кр	ср к-сть іт	е.в.3ий кр	ср к-сть іт	min
Метод спряжених градієнтів	0.000000022	13	0.90000003	9	0.00000001081	26	[1;1]
Метод Ньютона	0.000000074		0.00000002236		0.0000000286		
Метод змінної метрики	0.1414213		0.0000000194		0.000000058		

Таблиця 17

$f(x)$ = -100 $*$ $(x2 - (x1 - 1)\text{^}3)\text{^}2 + (2 - x1)\text{^}2$; x^0 =[-1.2;1]							
метод	е.в.1ий кр	ср к-сть іт	е.в.2 ий кр	ср к-сть іт	е.в.3ий кр	ср к-сть іт	min
Метод спряжених градієнтів	0.00000024999	20	0.000000102	16.3	0.000008788	22.6	[1;1]
Метод Ньютона	0.00000006		0.000000026		0.000000141		
Метод змінної метрики	0.000000090		0.0000003		0.000000050		

Таблиця 18

Висновки

В результаті проведених розрахунків було встановлено, що для методу спряжених градієнтів найбільш відповідним є перший критерій зупинки. Для методу змінної метрики найкраще підходить другий критерій зупинки. Щодо методу Ньютона, для нього найбільш придатним виявився третій критерій зупинки. Найкраще це видно в 1, 5-9 функціях. Найімовірніше через прості відповіді, адже мінімум у цих функціях [1;1]. В інших функціях відповідь не була такою ж точною, але там теж можна бачити такий самий результат евклідової відстані в тому чи іншому методі при кожному критерію зупинки. Але кожен метод все також дає хороші результати і їх також можна використовувати в будь-якому випадку.

СПИСОК ЛІТЕРАТУРИ

1. А.Т. Яровий Методи оптимізації та варіаційне числення: Навчально-методичний посібник для студентів спеціальності “Математика” / А.Т.Яровий, Є.М.Страхов, 2017р. - С. 84-86
2. Яровий, Анатолій Трохимович. Методи оптимізації та варіаційне числення : навч.- метод. посіб. для студ. спец. "Математика" / А. Т. Яровий, Є. М. Страхов; ОНУ ім. І.І. Мечникова, ІМЕМ . – Одеса : Освіта України, 2017 . – 153 с.

ДОДАТКИ

Додаток А

```
import numpy as np

def f(X):
    x1, x2 = X
    return (x1**2 + 12*x2 - 1)**2 + (x1 + x2**2 - 7)**2

def grad_f(X):
    x1, x2 = X
    df_dx1 = 2*(2*x1)*(x1**2 + 12*x2 - 1) + 2*(x1 + x2**2 - 7)
    df_dx2 = 2*(12)*(x1**2 + 12*x2 - 1) + 2*(2*x2)*(x1 + x2**2 - 7)
    return np.array([df_dx1, df_dx2])

def conj_gradient_method(X0, epsilon):
    X_prev = X0
    grad_prev = grad_f(X_prev)
    d_prev = -grad_prev
    iteration = 0

    while True:
        alpha = -np.dot(grad_prev, d_prev) / np.dot(d_prev, d_prev)
        X_next = X_prev + alpha * d_prev
        grad_next = grad_f(X_next)
        beta = np.dot(grad_next, grad_next) / np.dot(grad_prev, grad_prev)
        d_next = -grad_next + beta * d_prev

        if np.linalg.norm(X_next - X_prev) <= epsilon and np.abs(f(X_next) -
f(X_prev)) <= epsilon:
            break

        X_prev = X_next
        grad_prev = grad_next
        d_prev = d_next
        iteration += 1

    return X_next, iteration

initial_point = np.array([1, 1])
```

```
epsilon = 1e-6

solution, num_iterations = conj_gradient_method(initial_point, epsilon)

print("Решение: ", solution)
print("Количество итераций:", num_iterations)
```

Додаток Б

```
import numpy as np

def f(x):
    return 100 * (x[1] - x[0]**2)**2 + (1 - x[0])**2

def grad_f(x):
    df_dx1 = 400 * (x[0]**3 - x[0]*x[1]) + 2 * (x[0] - 1)
    df_dx2 = 200 * (x[1] - x[0]**2)
    return np.array([df_dx1, df_dx2])

def hessian_f(x):
    d2f_dx1dx1 = 1200 * x[0]**2 - 400 * x[1] + 2
    d2f_dx1dx2 = -400 * x[0]
    d2f_dx2dx1 = -400 * x[0]
    d2f_dx2dx2 = 200
    return np.array([[d2f_dx1dx1, d2f_dx1dx2], [d2f_dx2dx1, d2f_dx2dx2]])

def newton_method(f, grad_f, hessian_f, x0, tol, max_iter):
    x = x0
    for i in range(max_iter):
        grad = grad_f(x)
        hessian = hessian_f(x)
        h = np.linalg.solve(hessian, -grad)
        x_new = x + h
        if np.linalg.norm(x_new - x) < tol:
            return x_new
        x = x_new
    return x

x0 = np.array([-1.2, 1.0])
tol = 1e-6
max_iter = 1000
```

```
solution = newton_method(f, grad_f, hessian_f, x0, tol, max_iter)

print("Solution: x =", solution)
print("Activation: f(x) =", f(solution))
```

Додаток В

```
import numpy as np
from scipy.optimize import minimize

def himmelblau(x):
    return (x[0]**2 + x[1] - 11)**2 + (x[0] + x[1]**2 - 7)**2

def himmelblau_grad(x):
    return np.array([4*x[0]*(x[0]**2 + x[1] - 11) + 2*(x[0] + x[1]**2 - 7),
                     2*(x[0]**2 + x[1] - 11) + 4*x[1]*(x[0] + x[1]**2 - 7)])

x0 = np.array([1, 1])
tolerance = 1e-5
previous_grad = himmelblau_grad(x0)
iterations = 0

while True:
    res = minimize(himmelblau, x0, method='BFGS', jac=himmelblau_grad)

    if np.linalg.norm(himmelblau_grad(res.x) - previous_grad) < tolerance:
        break

    previous_grad = himmelblau_grad(res.x)
    iterations += 1

print(f'Optimized point: {res.x}')
print(f'Iterations: {iterations}')
```