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

Фізико-математичний факультет

Кафедра математичної фізики

«На правах рукопису»

УДК 519.6

«До захисту допущено»

Завідувач кафедри

_________ Горбачук В. М.

“13” грудня 2018 р.

Магістерська дисертація

зі спеціальності (спеціалізації) 111 математика (математичні та комп’ютерні методи в моделюванні динамічних систем).

на тему: «**Оптимізація функції двох змінних за допомогою аналізу елементів внутрішньої геометрії поверхні**»

Виконав: студент II курсу магістратури, групи ОМ-72мп

Петриняк Денис Юрійович _________

Науковий керівник доцент, кандидат фізико-математичних наук

Селезньова Надія Петрівна __________

Рецензент: доцент, кандидат фізико-математичних наук

Гончаренко Юрій Вікторович ___________

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент _____________

Київ – 2018

РЕФЕРАТ

Обсяг: 72с., 4 ч., 1 табл., 19 рис., 2 дод., 10 джерел.

Актуальність теми - процес відшукання глобального екстремуму функцій багатьох змінних є достатньо складним. Відомі числові методи оптимізації при обчисленні екстремальних точок вимагають додаткових досліджень достатності умов, які для неквадратичних функцій є досить складними, тому пропонується на кожному кроці оптимізації додатково обчислювати кривини поверхні в кожній точці. Це дозволяє визначати тип цих точок, що в свою чергу дає більш точне уявлення про рельєф поверхні і задача оптимізації уже не потребує додаткових досліджень достатності умов екстремуму.

Робота має зв'язок з науковими програмами: в рамках теми «Розв'язок методів дослідження розвязків диференціально-операторних рівнянь і рівнянь із частинними похідними параболічного типу», державний реєстраційний номер 0117u003173, 1.01.2017 – 12.2021. в Національному технічному університеті України «Київський політехнічний інститут імені Ігоря Сікорського».

Мета роботи - покращити стандартний метод оптимізації функції двох змінних за допомогою введення достатніх умов екстремуму шляхом аналізу кривин поверхні в поточних точках та визначення типу цих точок на поверхні.

Об'єкт дослідження – функція двох змінних.

Предмет дослідження – кривини поверхні (Гауссова, середня та головні кривини).

Методами дослідження екстремуму функції двох змінних в даній роботі є чисельні методи: ґрадієнтний метод, метод Хука-Дживса, метод спряжених напрямків Пауелла, геометричні характеристики поверхонь - кривини.

Наукова новизна одержаних результатів – включення до чисельного методу оптимізації достатніх умов за допомогою кривин поверхні.

В роботі проведена геометрична інтерпретація квадратичної апроксимації поверхні стичним параболоїдом, і перехід від параболоїду до індикатриси Дюпена. Це зроблено у зв'язку з тим, що в загальному випадку задача відшукання головних кривин поверхні та головних напрямків у кожній поточній точці на сьогоднішній день в загальному випадку не вирішена. Як правило, її вирішують у внутрішніх координатах поверхні і перехід до зовнішньої системи координат є досить складним, а інколи і неможливим. Маючи рівняння індикатриси Дюпена - кривої другого порядку, можна відомими методами знайти головні напрямки відносно цієї кривої і далі рухатись по поверхні в одному із цих напрямків, адже, як відомо, в цих напрямках головні кривини поверхні набувають своїх екстремальних значень. В результаті таких досліджень, доповнено чисельний ґрадієнтний метод оптимізації достатніми умовами екстремуму, а саме обчислено кривини поверхні в поточних точках, що дає можливість класифікувати поточні точки, а в підсумку зрозуміти чи є отримана точка екстремальною., що продемонстровано на прикладах.

Практичне значення – доповнений за допомогою обчислення кривин чисельний метод дозволить відразу вияснити, чи є дана точка екстремальною чи сідловою, чи точкою сплощення, чи омбілічною.

Апробація роботи - зроблено доповідь на науковому семінарі кафедри математичної фізики. Стаття " Кривини поверхні та оптимізація функції двох змінних" (обсягом 9с.) прийнята до друку в матеріали VII Міжнар. наук.-практ. конф. «Математика в сучасному технічному університеті», Київ, 27—28 грудня 2018 р. — Київ: КПІ ім. Ігоря Сікорського, 2019.

Ключові слова: НОРМАЛЬНА КРИВИНА, ГАУССОВА КРИВИНА, СЕРЕДНЯ КРИВИНА, І ТА ІІ КВАДРАТИЧНІ ФОРМИ ПОВЕРХНІ, ҐРАДІЄНТНІ МЕТОДИ ОПТИМІЗАЦІЇ, СПРЯЖЕНІ НАПРЯМКИ, ГОЛОВНІ НАПРЯМКИ, ЛІНІЇ КРИВИНИ, СТИЧНИЙ ПАРАБОЛОЇД, ІНДИКАТРИСА ДЮПЕНА.

ABSTRACT

Volume: 72 p., 4 parts, 1 table, 19 illustrations, 2 attachments, 10 sources.

Topicality of the topic **-** the process of finding the global extremum of the functions of many variables is rather complicated. Known numerical optimization methods for calculating extreme points require additional studies of the sufficiency of conditions that are rather complex for non-quadratic functions, therefore, at each optimization step, it is proposed to further calculate the curvatures of the surface at each point. This allows us to determine the type of these points, which in turn provides a more accurate representation of the surface relief, and the optimization problem no longer requires additional studies of the sufficiency of extreme conditions.

The work has a connection with scientific programs within the topic «Solving methods for investigating the differences between differential operator equations and equations with partial derivatives of a parabolic type», state registration number 0117u003173, 01.01.2017 - 12.2021. in the National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute».

The purpose of the work is to improve the standard method for the optimization of the function of two variables by analyzing the surface curvature at current points.

Object of study – is a function of two variables.

The subject of research is the curvature of the surface (Gauss curvature, middle curvature and main curvatures).

Research Methods extremum function of two variables in this study is numerical methods: gradient method, pattern search method (Hooke and Jeeves),

method of conjugate directions Powell, geometric properties of surfaces - curvature.

The scientific novelty of the obtained results is the inclusion in the numerical method of optimization of sufficient conditions with the help of surface curvature.

In this paper, a geometric interpretation of the quadratic approximation of the surface with a contiguous paraboloid is carried out, and the transition from paraboloid to the Dupin indicatrix. This is due to the fact that in the general case the task of finding the main curvatures of the surface and the main directions at each current point to date is generally not solved. As a rule, it is solved in the internal coordinates of the surface and the transition to the external coordinate system is rather complicated, and sometimes impossible. With the equation of the Dupin indicatrix of the second order curve, one can find, by known methods, the main directions in relation to this curve and then move along the surface in one of these directions, since in these directions the main curvatures of the surface acquire their extreme values. As a result, complemented numerical gradient optimization method is sufficient extremum conditions such as curvature surface is calculated at current locations, which allows to classify the current point, and eventually got to understand whether the extreme point, as demonstrated by examples.

Practical value - in general, in numerical methods of optimizing the function there is no procedure for verifying sufficient conditions of extremum; in the paper, we propose a method for checking the sufficient conditions of an extremum.

Test work - a report was presented at the scientific seminar of the Department of Mathematical Physics. Article "Curvature of the surface and optimization of the function of two variables" (volume is 9 p.) Accepted for printing in materials VII Intern. scient.- practical. conf . "Mathematics in the

Modern Technical University", Kyiv, December 27-28, 2018 - Kyiv: Igor Sikorsky KPI, 2019.

Key words: MAIN CURVATURES, GAUSS CURVATURE, MIDDLE CURVATURE, FIRST AND SECOND QUADRATIC FORMS OF SURFACE, GRADIENT METHODS OF OPTIMIZATION, CONJUGATE DIRECTIONS, MAIN DIRECTIONS, CURVATURE LINES, CONTIGUOUS PARABOLOIDS, DUPIN INDICATRIX.

Зміст

Частина 1

Перелік основних термінів

Основні терміни даної роботи –це терміни диференціальної геометрії: нормальні кривини кривої на поверхні, гауссова кривина, середня кривина, перша та друга квадратичні форми поверхні, ґрадієнтні методи оптимізації, спряжені напрями, головні напрямки на поверхні, лінії кривини, сідлові, еліптичні, омбілічні, асимптотичні точки поверхні.

Вступ

На сьогоднішній день створено та досліджено велику кількість методів мінімізації функцій багатьох змінних. До класичних методів відносяться ґрадієнтні методи та метод Ньютона. Вони мають велике значення в ідейному відношенні. В основі цих методів лежить ідея заміни функції, мінімум якої шукають, в околі поточної точки x_k першими членами її розвинення в ряд Тейлора. В ґрадієнтному методі беруть лінійну частину розвинення, а в методі Ньютона – квадратичну. Більшість методів оптимізації будують на тій же ідеї апроксимації функцій.

В представленій роботі реалізовано ґрадієнтний метод в Excel та метод Хука-Дживса в PascalABC. Знайдено мінімум різних нелінійних функцій в тому числі і функцій з яружним рельєфом. Ґрадієнтний метод доповнено аналізом кривин поверхні в поточних точках, що дозволило на кожному кроці зрозуміти рельєф функції та проаналізувати отриману точку на екстремум.

Актуальність теми

На практиці виникають все нові і нові задачі оптимізації, при чому складність цих задач постійно зростає. Тому виникає необхідність у створенні та вдосконаленні нових математичних моделей та методів, що враховують наявність багатьох критеріїв, проводять глобальний пошук оптимуму. Потреба у чисельних методах знаходження екстремуму функцій виникає, коли система із частинних похідних не має аналітичного розв'язку або містить складну нелінійність. Також однією із найбільш вагомих складових у дослідженні операцій є оптимізація поставленої задачі.

Аналітично можна розв'язати тільки невелику частину задач оптимізації. Реальні прикладні задачі оптимізації є достатньо складними. Сучасні методи оптимізації далеко не завжди справляються із розв'язуванням таких задач. Поки що не існує такої теорії, яка б врахувала будь-які особливості функцій, що описують оптимізаційні задачі. Отже, пріоритетними є ті методи, якими найпростіше управляти в процесі розв'язування задачі.

Історична довідка

До другої половини XVII не існувало ніяких загальних прийомів розв’язування задач на екстремум. Прагнення їх знайти в значній мірі стимулювало створення математичного аналізу. Перший загальний метод дослідження задач на екстремум відкрив П. Ферма (близько 1630 р.). Сучасною мовою його можна сформулювати так: у точці екстремуму деякої функції однієї змінної похідна дорівнює нулеві, тому екстремуми слід шукати серед коренів похідної. Цей результат включено зараз до шкільного курсу математики під назвою «теорема Ферма». Фактично Ферма описав цей прийом лише для алгебраїчних многочленів. У загальному вигляді його вперше отримав видатний англійський фізик, механік, астроном і математик Ісаак Ньютон (60-ті роки XVII ст.). Потім його перевідкрив відомий німецький математик, фізик і філософ Готфрід Вільгельм Ляйбніц і вперше опублікував у знаменитій статті, з якої починається історія математичного аналізу. Примітним є початок назви статті Лейбніца: «Новий метод знаходження найбільших і найменших величин… » [1; с.10]. У XVIII столітті зусиллями швейцарського математика, фізика, механіка і астронома Леонарда Ейлера та французького математика і механіка Жозефа Луі Лагранжа, були розроблені методи розв'язування екстремальних задач із цільовими функціями від кількох змінних без обмежень на аргументи та з обмеженнями типу рівностей. Основним серед таких методів є метод множників Лагранжа, який зараз входить до програми кожного математичного курсу вищих навчальних закладів. Пізніше ці дослідження були доповнені методами розв'язування задач, в яких обмеження на аргументи задаються як рівностями, так і нерівностями. Розділ прикладної математики, де розглядаються проблеми, пов'язані з такими задачами, одержав назву *математичного програмування*.

Пошуки шляхів розв'язування конкретних практичних екстремальних задач, що виникали у геометрії, фізиці, механіці, призвели [1; с.11] до створення нового розділу математичного аналізу, який одержав назву *варіаційного числення.* Почалось з того, що в першому науковому журналі *«Акта Ерудиторум»* за червень 1696 р. було розміщено замітку відомого швейцарського математика, учня і послідовника Г. В. Лейбніца-Іоганна Бернуллі. Замітка мала назву *«Нова задача, до розв'язування якої запрошуються математики».* Задача формулювалась так*: «У вертикальній площині дано дві точки А та В. Визначити шлях АМВ, спускаючись яким під дією власної ваги, тіло М, починаючи рух з точки А, дійде до точки В за найменший проміжок часу».* Розв'язок цієї задачі, за словами Г. В. Лейбніца, «настільки чудової і досі невідомої», було дано самим І. Бернуллі, а також Г. В. Лейбніцем, Якобом Бернуллі і ще одним анонімним автором, у якому знавці, за словами І. Бернуллі, «ex unge leonem» (за кігтями впізнають лева) відразу впізнали І. Ньютона. Крива найшвидшого спуску або брахістохрона виявилась названа також циклоїдою. Саме із задачі про брахістохрону почалась історія класичного варіаційного числення. Загальні методи розв'язування задач варіаційного числення були розроблені у XVIII столітті Л. Ейлером і Ж. Лагранжем. Над теорією варіаційного числення вчені працювали більше двох століть. Окрім необхідних умов першого порядку (рівнянь Ейлера-Лагранжа) були знайдені необхідні і достатні умови другого порядку для двох типів екстремумів - сильного і слабкого (А. Лежандр, К. Якобі, К. Вейєрштрасс), запропонований новий підхід до розв'язування варіаційних проблем (теорія Гамільтона-Якобі), побудована теорія поля (А. Кнезер, Д. Гільберт).

До середини 30-х років XX століття більшість вчених вважали, що проблематика задач на екстремум практично вичерпана. Все змінилося, коли в 1939 році до професора Л. В. Канторовича прийшли на консультацію представники фанерного тресту і запропонували його увазі кілька задач, що виникли у них на виробництві. При математичній формалізації задач виявилось, що вони зводяться до знаходження екстремуму лінійної функції на множині точок многогранника [1; с.11]. Перебрати всі вершини многогранника було майже неможливо через їх велику кількість. Канторович Л.В. дослідив такі задачі та запропонував метод для розв'язування, заклавши основи нового напряму в теорії екстремальних задач. Цей напрям отримав назву *лінійного програмування.* Термін «лінійне програмування» з'явився в середині 40-х років XX століття в працях Т. Ч. Купманса.

Перша праця, де розглядались загальні питання лінійного програмування і викладені основні ідеї *методу послідовного поліпшення плану* для невироджених задач лінійного програмування, була надрукована Дж. Б. Данцігом у 1949 році [1; с.12]. Метод Данціга увійшов в математику під назвою *симплексний метод.*

Значний внесок у дослідження загальної задачі лінійного програмування зробив відомий [1; с.12] американський математик і фізик Джон фон Нейман, який встановив, що будь-яку матричну гру двох осіб з нульовою сумою можна подати у вигляді задачі лінійного програмування і навпаки.

Методи лінійного програмування знайшли широке використання на практиці. Зокрема за розробку математичних методів та їх впровадження в економіку Л. В. Канторовичу разом з американським економістом Купмансом у 1975 році була присуджена [1; с.12] Нобелевська премія. У 40-х роках XX століття теорія екстремальних задач пережила своє друге народження. Створення теорії лінійного програмування дало поштовх розвиткові інших розділів теорії оптимізації, насамперед - *опуклого аналізу* і *нелінійного програмування.*

Зараз математичне програмування розвивається у напрямі дослідження все більш спеціальних класів екстремальних задач (у відповідності до потреб [1; с.13] різноманітних застосувань), розробки чисельних методів, які долають негладкість, неопуклість, яружніість, розривність цільових функцій, а також враховують багатоекстремальність, можливу нечітку і стохастичну природу оптимізаційних задач.

Значний внесок у розвиток зазначених напрямів теорії оптимізації зробили також українські [1; с.13] вчені: В. М. Глушков, В.С. Міхалевич, Ю.М. Єрмольєв, Б.М. Пшеничний та ін.

Багато задач оптимізації пов'язано з управлінням різноманітними процесами, приладами, системами. Наприклад, візок, рухається прямолінійно без тертя на горизонтальних рейках і управляється зовнішньою силою, яку можна змінювати в [1; с.13] заданих межах. Потрібно зупинити візок у визначеному положенні за найкоротший час. Ця задача називається найпростішою *задачею про швидкодію* в автоматичному регулюванні.

Особливість постановки технічних екстремальних задач полягає в тому, що діючі сили поділяються на два типи: одні з них – нерегульовані сили природи [1; с.13] (наприклад, сила тяжіння), інші - регульовані (наприклад, сила тяги). Крім того, існують обмеження на управляючі дії, пов'язані з технічними характеристиками систем.

Значна кількість подібних задач виникла в хімічній промисловості, у військовій справі, космонавтиці і т.п. При цьому виявилося, що методів варіаційного числення не достатньо для розв'язування цих задач. У кінці сорокових і на початку п'ятидесятих років XX століття зусиллями Л.С. Понтрягіна та його учнів знайдено формалізацію цілого класу задач, що охоплювали більшість актуальних проблем техніки, а потім була побудована теорія цього класу задач. Вона [1; с.14] отримала назву *теорії оптимального управління*. Основним результатом цієї теорії є принцип максимуму Понтрягіна, який встановлює необхідні умови оптимальності в задачах такого класу. Цікаво відмітити, що перша задача, яка відноситься до оптимального управління, була поставлена Ньютоном у «Математичних початках натуральної філософії» (1687 р.) ще до задачі про брахістохрону.

Так сформувалися основні розділи сучасної теорії [1; с.14] екстремальних задач: *математичне програмування*, *варіаційне числення* та *теорія оптимального управління*.

Постановка задачі оптимізації

Екстремальна задача в математичному поданні є задачею [1; с.26] відшукання екстремуму (мінімуму чи максимуму) деякої функції $f(x)$ на деякій множині X .

Серед екстремальних задач виділяють задачі мінімізації або задачі максимізації.

Задача мінімізації записується у вигляді

$$f(x^*) \to \min, x \in X \qquad (1)$$

або

$$\min_{x \in X} f(x^*) ,$$

де функція $f(x)$ називається *цільовою функцією*, X - *допустимою множиною*, а будь-який елемент $x \in X$ - *допустимою точкою* цієї множини.

Задача [1; с.26] пошуку максимуму функції $f(x)$ зводиться до еквівалентної задачі пошуку мінімуму шляхом зміни знаку перед функцією на протилежний:

$$f(x^*) \to \max f(x) = -\min(-f(x)), x \in X .$$

Означення

Точка $x^* \in X$ називається *розв'язком задачі* (1) або *точкою глобального мінімуму*, якщо

$$f(x^*) \le f(x) \ \forall x \in X \qquad (2)$$

Означення

Точка $x^* \in X$ називається *точкою локального мінімуму* або *локальним розв'язком* задачі (1.1), якщо існує число $\exists \varepsilon > 0$ таке, що

$$f(x^*) \le f(x) \ \forall x \in X \cap U_\varepsilon(x^*), \ (3)$$

де $U_\varepsilon(x^*)$ - ε -окіл точки x^*.

Означення

Якщо в (2) і (3) нерівність є «<» при $x \ne x^*$, то x^* називається відповідно *точкою строгого глобального* чи *строгого локального мінімуму.*

Той факт, що точка $x^* \in X$ є точкою глобального [1; с.27] мінімуму функції $f(x)$ на множині X, записують у вигляді

$$f(x^*) = \min_{x \in X} f(x)$$

або

$$x^* = \arg\min_{x \in X} f(x) .$$

Основні класи екстремальних задач

Навіть неповний аналіз формальних моделей екстремальних задач показує, що класифікацію задач [1; с.57] оптимізації можна проводити за кількома ознаками в залежності від типу і властивостей цільової функції, способів подання і властивостей допустимої множини.

Розглянемо найбільш важливі для теорії і практики класи екстремальних задач.

1. Без урахування специфіки цільової функції $f(x)$ задачі оптимізації поділяються на два класи: задачі безумовної оптимізації та задачі умовної оптимізації.

Задача

$$f(x) \to \min(\max), x \in X \qquad (1)$$

називається *задачею безумовної мінімізації (максимізації)*, якщо $X = R^n$, тобто якщо задача має вигляд

$$f(x) \to \min(\max), x \in R^n .$$

Серед чисельних методів безумовної оптимізації найбільш розповсюдженими є: метод покоординатного спуску, ґрадієнтні методи, метод Ньютона та його модифікації, метод спряжених ґрадієнтів.

Задача (1) називається *задачею умовної мінімізації (максимізації)*, якщо X - власна підмножина простору R^n, тобто X - непорожня підмножина R^n, яка не співпадає з R^n.

Для розв’язування задач умовної оптимізації використовують, зокрема, метод проекції ґрадієнта, метод умовного ґрадієнта, метод можливих напрямів, метод штрафних функцій.

Одним із найважливіших класів задач умовної оптимізації, який є узагальненням класичної задачі на умовний екстремум, є *задачі* *математичного програмування*. Якщо в задачі математичного програмування цільова функція лінійна і допустима множина визначається системою лінійних рівнянь і нерівностей, то вона називається *задачею лінійного програмування.* Якщо цільова функція нелінійна, то задача математичного програмування називається задачею *нелінійного програмування.* Серед задач нелінійного програмування найбільш важливими і добре вивченими є задачі квадратичного і опуклого програмування.

Серед прикладних задач оптимізації важливе місце займають також задачі дискретного, цілочислового, дробово-лінійного, параметричного і стохастичного програмування.

2. *Задача класичного варіаційного числення* полягає в знаходженні [1; с.64] мінімуму функціоналу виду

$$I = \int_{t_0}^{t_1} L(t, x(t), \dot{x}(t))dt \quad (2)$$

при обмеженнях:

$$M_i(t, x(t), \dot{x}(t)) = 0, i = \overline{1,k} \quad (3),$$

$$\Psi_j(x(t_0), x(t_1)) = 0, j = \overline{1,s} \quad (4),$$

де $x(t) = (x_1(t), ..., x_n(t))$ - вектор-функція з простору $C^1\left(\left[t_0, t_1\right], R^n\right)$ неперервно диференційованих n - вимірних вектор-функцій; $\dot{x}(t) = (\dot{x}_1(t), ..., \dot{x}_n(t)),$ при цьому $\dot{x}_i(t) = \frac{dx_i(t)}{dt}, (i = \overline{1,n})$; $L(t, x(t), \dot{x}(t)),$ $M_i(t, x(t), \dot{x}(t))$ $(i = \overline{1,k})$ - диференційовані функції від $2n+1$ змінної; $\Psi_j(x(t_0), x(t_1)),$ $(j = \overline{1,s})$ - диференційовані функції від $2n$ змінних.

Рівняння (3) називаються *диференціальними зв'язками*, а умови (4) - *граничними умовами*.

Задача мінімізації функціоналу (2) при обмеженнях (3), (4) називається *задачею Лагранжа*.

Задача

$$I = \int_{t_0}^{t_1} L(t, x(t), \dot{x}(t))dt \to \min, x(t_0) = x_0, x(t_1) = x_1$$

називається *найпростішою векторною задачею класичного варіаційного числення*, а при $n = 1$ - *найпростішою задачею класичного варіаційного числення.*

При розв'язуванні задач варіаційного числення використовують метод невизначених множників Лагранжа, а також наближені методи, за якими знаходять розв'язок із наперед заданою точністю. Серед наближених методів найбільш відомими є методи Рітца, Гальоркіна, Анторовича.

3. Одним із найпоширеніших класів екстремальних задач, які мають вагоме практичне значення, є *задачі оптимального управління*. Вони пов'язані, наприклад, з механікою польотів космічних апаратів, з роботою електроприводів, [1; с.65] хімічних і ядерних реакторів, з питаннями віброзахисту і амортизації, математичною економікою тощо.

Огляд основних методів оптимізації

Чисельні методи розв'язування задач оптимізації базуються, як правило, на наближеному обчисленні [1; с.327] значень цільової функції, значень функцій, які визначають допустиму множину точок, значень похідних (частинних похідних) цих функцій або їх аналогів. На основі отриманих результатів будується наближення до розв'язку задачі - шуканої точки мінімуму x^* або, якщо така точка не єдина, до однієї з точок множини точок мінімуму X^*. Іноді будується наближення до мінімального значення цільової функції

$$f(x^*) = \min_{x \in X} f(x).$$

Всі відомі чисельні методи пошуку екстремумів нелінійних функцій, залежно від використаної [7; с. 321] інформації для отримання наступної точки, можна поділити на три основні групи. Методи, які використовують відомості лише про значення цільової функції, називаються *методами нульового порядку*, методи, які використовують також відомості про значення похідних (частинних похідних) першого порядку або їх аналогів, - *методами першого порядку*, методи, які використовують, крім того, відомості про похідні (частинні похідні) другого порядку, - *методами другого порядку*.

Методи, які використовуються для розв'язування задач виду

$$f(x) \to \min, x \in R^n ,$$

називаються *методами безумовної мінімізації*, а методи, які використовуються для розв'язування задач виду

$$f(x) \to \min,\ x \in X\ ,$$

називаються *методами умовної мінімізації.*

Найбільш відомими методами безумовної мінімізації є: метод *покоординатного спуску* і *симплексний метод* (не слід плутати з симплекс-методом розв'язування задачі лінійного програмування), які відносяться до методів нульового порядку, *ґрадієнтні* і *субґрадієнтні методи*, *метод спряжених ґрадієнтів*, які відносяться до методів першого порядку, метод Ньютона та його модифікації, які відносяться до методів другого порядку.

Серед методів умовної мінімізації найбільш відомими є: *методи проекції ґрадієнта* і *субґрадієнта*, *метод можливих напрямів*, *метод умовного ґрадієнта*, *методи штрафних функцій*.

Найбільш розповсюдженими наближеними методами безумовної та умовної оптимізації є так звані *методи послідовних наближень* або *ітераційні методи*. Багато ітераційних методів мінімізації відносяться до *методів спуску*. В цих методах [1; с.328] напрям руху на кожному кроці ітераційного процесу обирається з числа напрямів спадання функції, що мінімізується.

Частина 2. Опис чисельних методів оптимізації, використаних у роботі

Метод Хука-Дживса

Постановка задачі

Потрібно знайти безумовний [6; с.199] мінімум функції $f(x)$ багатьох змінних, тобто знайти таку точку $x^* \in R^n$, щоб $f(x^*) = \min\limits_{x \in R} f(x)$.

Стратегія розв'язання задачі

Метод конфігурацій є модифікацією методу циклічного покоординатного пошуку. Цей метод відомий як метод Хука-Дживса (R.Hooke, T.A. Jeeves) представляє собою комбінацію *пошуку, що досліджує*, із циклічною зміною змінних, і *прискорювального пошуку за зразком*. Досліджуваний пошук орієнтований на виявлення локального поводження цільової функції і визначення напрямку її спадання вздовж «ярів». Отримана інформація використовується при пошуку за зразком при русі вздовж «ярів».

Досліджуваний пошук починається в деякій початковій точці x^0, яка називається *старим базисом*. Як множина напрямків пошуку вибирається множина *координатних* напрямків e^i, $i = 1,...,n$. Задається величина кроку, яка може бути різною для різних координатних напрямків і змінною в процесі пошуку. Фіксується перший координатний напрямок і робиться крок вбік збільшення відповідної змінної. Якщо значення функції в пробній точці менше значення функції у вихідній точці, крок вважається вдалим. В протилежному випадку необхідно повернутися в попередню точку і зробити крок в протилежному напрямку з наступною перевіркою поводження функції. Після перебору всіх координат пошук, що досліджує, завершується. Отримана точка називається *новим базисом* x^1. Якщо досліджуваний пошук з даною величиною кроку невдалий, то вона зменшується і процедура триває.

Пошук закінчується, коли поточна величина кроку стане меншою за деяку величину.

Пошук за зразком полягає в русі за напрямком від старого базису до нового (від точки x^0 через точку x^1, із точки x^1 через точку x^2, із x^2 через x^3 і т.д.). Величина прискорювального кроку задається прискорювальним множником λ. Успіх пошуку за зразком визначається за допомогою досліджуваного пошуку з отриманої точки. Якщо при цьому значення в найкращій точці менше, ніж у точці попереднього базису, то пошук за зразком вдалий, інакше пошук невдалий. Якщо пошук за зразком невдалий, відбувається повернення в новий базис, де продовжується досліджуваний пошук зі зменшеним кроком.

При пошуку за напрямком e^i змінюється тільки змінна x_i, а інші змінні залишаються фіксованими.

Алгоритм

1. Задати початкову точку x^0, число $\varepsilon > 0$ для закінчення обчислювального процесу, початкові величини кроків по координатних [6; с.200] напрямках $\Delta_1,...,\Delta_n \geq \varepsilon$, множник $\lambda > 0$, коефіцієнт зменшення кроку $\alpha > 1$. Вважаємо, що $y^1 = x^0,\ i = 1,\ k = 0$.
2. Здійснити досліджуваний пошук по обраному координатному напрямку:

   а) якщо $f(y^i + \Delta_i e^i) < f(y^i)$, тобто $f(y_1^i,...,y_i^i + \Delta_i,...,y_n^i) < f(y_1^i,...,y_i^i,...,y_n^i)$, то крок вважається вдалим. У цьому випадку варто вважати, що $y^{i+1} = y^i + \Delta_i e^i$ і перейти до кроку 3;

   б) якщо в пункті а) крок невдалий, то робиться крок в протилежному напрямку. Якщо $f(y^i - \Delta_i e^i) < f(y^i)$, тобто $f(y_1^i,...,y_i^i - \Delta_i,...,y_n^i) < f(y_1^i,...,y_i^i,...,y_n^i)$, то крок вважається вдалим. У цьому випадку варто вважати, що $y^{i+1} = y^i - \Delta_i e^i$ і перейти до кроку 3;

в) якщо в пунктах а) і б) кроки невдалі, то вважаємо $y^{i+1} = y^{i}$.

3. Перевірити умови:

   а) якщо $i < n$, то вважаємо, що $i = i + 1$ і переходимо до кроку 2 (продовжити досліджувальний пошук по напрямкам, що залишилися);

   б) якщо $i = n$, то перевірити успішність досліджувального пошуку, що досліджує:

   - якщо $f(y^{k+1}) < f(y^{k})$, то перейти до кроку 4;
   - якщо $f(y^{k+1}) \geq f(y^{k})$, то перейти до кроку 5.

4. Провести пошук за зразком. Вважати, що $x^{k+1} = y^{n+1}$,

   $y^{1} = x^{k+1} + \lambda(x^{k+1} - x^{k}), i = k, k = k + 1$ і перейти до кроку 2.

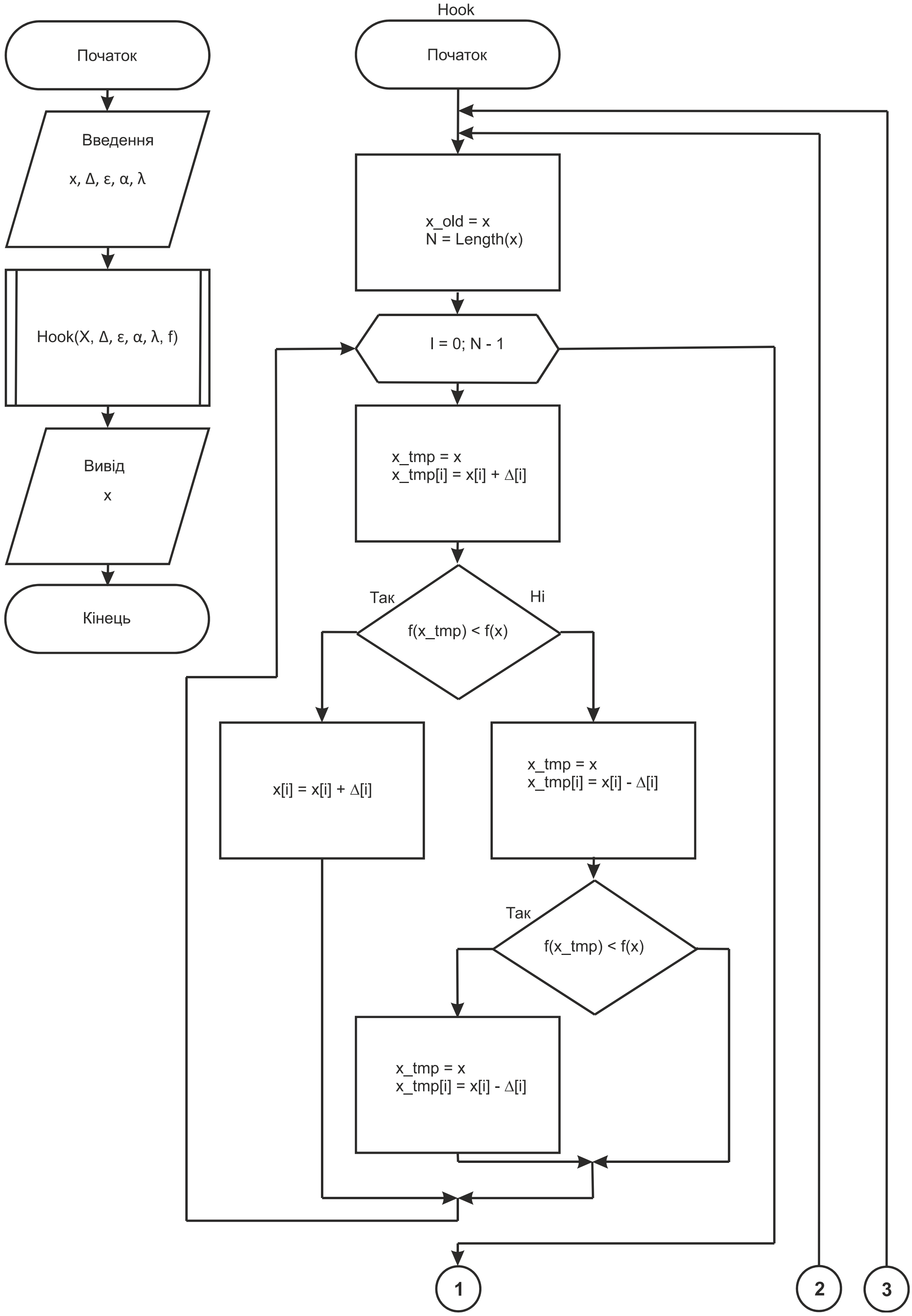
5. Перевірити умови закінчення:

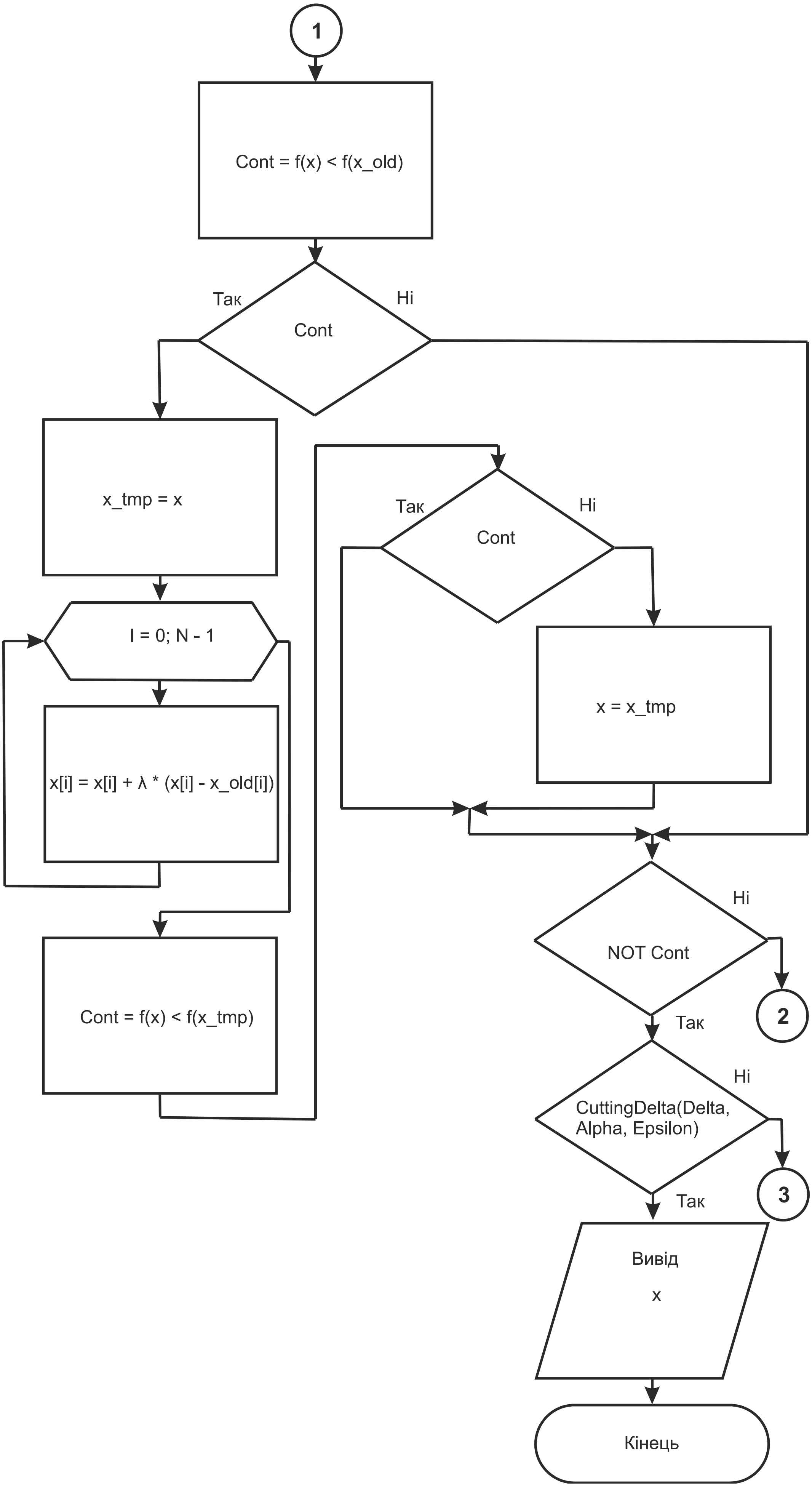
   а) якщо всі $\Delta_i > \varepsilon$, то пошук закінчити: $x^{*} \cong x^{k}$;

   б) для тих i, для яких $\Delta_i > \varepsilon$, зменшити величину кроку: $\Delta_i = \dfrac{\Delta_i}{\alpha}$.

   Вважати, що $y^{1} = x^{k}$, $x^{k+1} = x^{k}$, $k = k + 1, i = 1$ і перейти до кроку 2.

Блок-схема програми методу Хука-Дживса

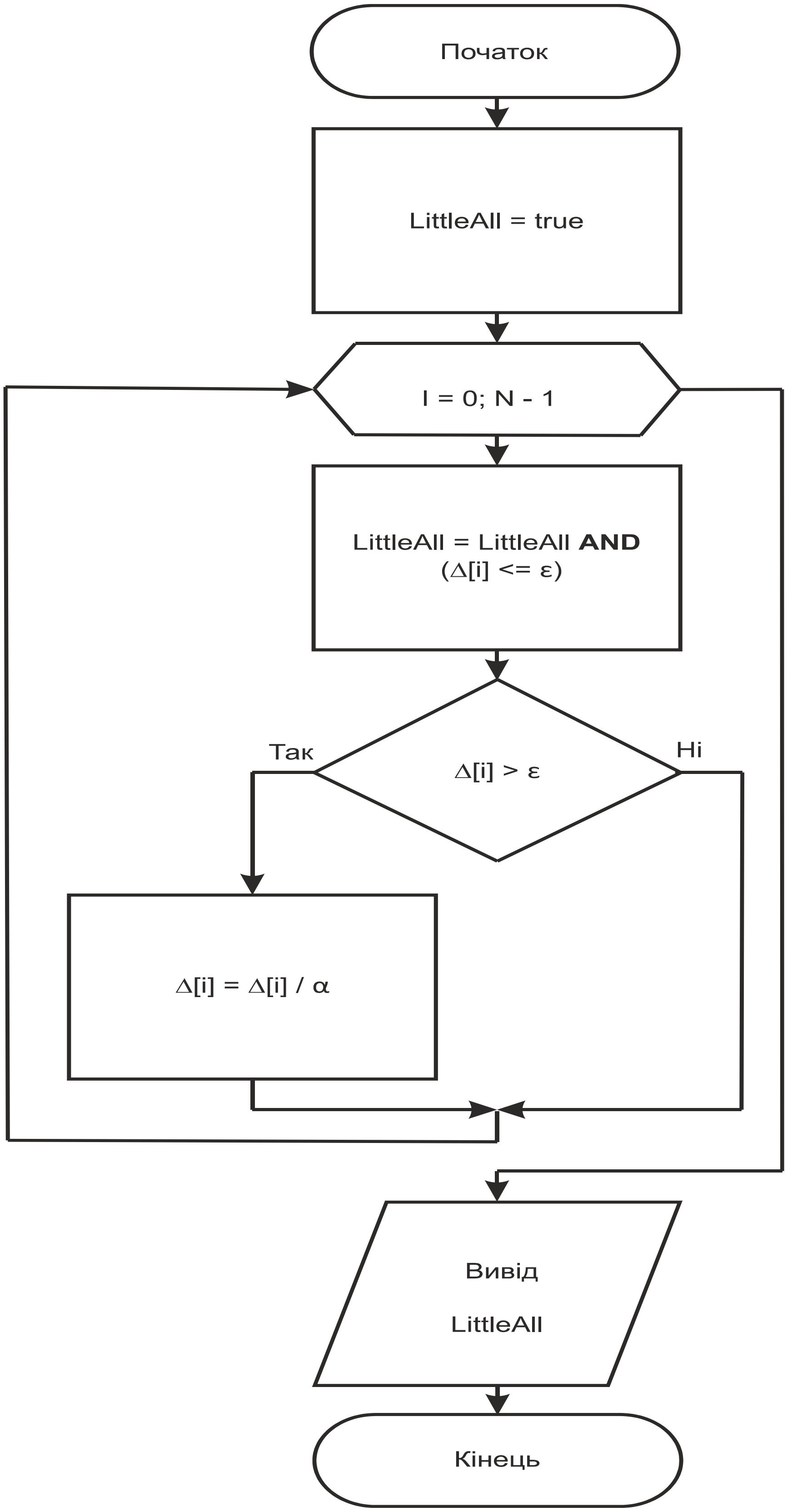



1
Cont = f(x) < f(x_old)
Так
Ні
Cont
x_tmp = x
I = 0; N - 1
x[i] = x[i] + λ * (x[i] - x_old[i])
Cont = f(x) < f(x_tmp)
Так
Ні
Cont
x = x_tmp
Ні
NOT Cont
Так
2
Ні
CuttingDelta(Delta, Alpha, Epsilon)
Так
3
Вивід
x
Кінець

// Зменшує кожен елемент Delta > Epsilon у Alpha раз.

// Істинна, якщо кожен елемент Delta став <= Epsilon

CuttingDelta(Delta, Alpha, Epsilon)



Метод спряжених напрямків Пауелла

Рекурентна схема більшості методів багатовимірної оптимізації побудована на формулі: $X_{k+1} = X_k + \lambda_k \vec{p}_k$. Вони відрізняються одне від одного способом вибору довжини кроку λ_k та напрямком пошуку $\vec{p}_k$. Перший із них – це скаляр, а другий – вектор одиничної довжини. Усі такі методи потребують певний степінь гладкості цільової функції. Також вони не дають гарантованої збіжності до глобального мінімуму, а в кращому випадку збігаються до одного із локальних мінімумів або до сідлової точки.

У 1962 році К.Сміт запропонував, а М. Пауелл в 1965 році вдосконалив процедуру пошукової оптимізації, яка використовує спряжені напрямки [10].

Одне із означень *спряжених напрямків*: система n лінійно незалежних напрямків $\vec{e}_1, \vec{e}_2, ..., \vec{e}_n$ називається спряженою по відношенню до додатно визначеної матриці H, якщо $\vec{e}_i^T \cdot \mathrm{H} \cdot \vec{e}_j = 0,\ \forall i \neq j.$ Це поняття спряженості цілком відповідає поняттю спряженості напрямків на поверхні, описане в класичному курсі диференціальної геометрії.

У методах оптимізації за матрицю H обирають матрицю Гессе складену із других похідних. Ця матриця є додатно визначеною, якщо цільова функція є опуклою. В такому випадку перехід до спряжених напрямків з геометричної точки зору означає вибір нових напрямків на поверхні, які більше відповідають топології поверхні функції, адже напрямки пошуку екстремуму витягуються вздовж головних осей квадратичної апроксимації функції.

Алгоритм Пауелла в деяких джерелах вважають методом нульового порядку, тому що він не застосовує похідні у явному вигляді. В інших джерелах його вважають ґрадієнтним методом, тому що неявно шукається оцінка матриці Гессе та використовується інформація про поведінку похідних. Крім того, як метод апроксимації цільової функції, його можна вважати методом Ньютонівського типу.

Теорема

Якщо при початковій точці X_0 пошуку в напрямку $\vec{P}$ мінімуму $f(X)$ міститься в деякій точці X_i і якщо при початковій точці пошуку $X^1 \neq X^0$ в тому ж самому напрямку $\vec{P}$ мінімум $f(X)$ міститься в точці X_j, то при $f(X_i) < f(X_j)$ напрямок $X_i - X_j$ є спряженим з напрямком $\vec{P}$ по відношенню до матриці Гессе функції $f(X)$.

Ця теорема має розширення на випадок декількох спряжених напрямків.

Ідея алгоритму Пауелла полягає в тому, що на кожному кроці пошуку визначається мінімум квадратичної функції, якою апроксимується цільова функція, вздовж кожного із спряжених напрямків до всіх попередніх. Після цього обирається нова система напрямків із застосуванням результатів пошуку і наступного *твердження:*

нехай $\vec{P}_1, \vec{P}_2, ..., \vec{P}_n$ деякі напрямки оптимізаційного пошуку функції $f(X)$ в R^n, B - матриця, стовпці якої є $\vec{P}_1, \vec{P}_2, ..., \vec{P}_n$. Тоді визначник матриці B набуває максимального значення тоді і лише тоді, коли є спряженими напрямки $\vec{P}_1, \vec{P}_2, ..., \vec{P}_n$ відносно матриці Гессе функції $f(X)$.

Доведено, що алгоритм Пауелла є збіжним до точки, в якій ґрадієнт функції дорівнює нулю та $f(X)$ є строго опуклою. Ця точка є локальним екстремумом. У методі Пауелла застосовують послідовність квадратичних інтерполяцій зі спеціальною процедурою настройки параметрів.

Фактично цей метод працює тільки з квадратичними функціями. Якщо його модернізувати так, щоб у кожній точці ітерації поверхні вписувати стичний параболоїд, а потім будувати індикатрису Дюпена, то, обчислюючи в кожній точці додатково кривини поверхні можна зрозуміти топологічний характер цих точок, аналізуючи які за кривинами, визначати екстремум функції, маючи уже достатні умови.

Алгоритм

1. Починаючи з X^0, за допомогою якого-небудь одновимірного пошуку визначається λ_1 так, щоб значення $f(X^0 + \lambda_1 \bar{S}_1)$ приймало мінімальне значення. Вважається, що $X^1 = X^0 + \lambda_1 \bar{S}_1$. Починаючи з X^1 здійснюється одновимірний пошук в напрямку $\bar{S}_2$ і визначається $X^2 = X^1 + \lambda_2 \bar{S}_2$. Пошук триває послідовно в кожному напрямку $\bar{S}_1, \bar{S}_2, ..., \bar{S}_n$, завжди починаючи з самої останньої точки, поки не будуть отримані всі $\lambda_i,\ i = 1, ..., n$.
2. Пошук, здійснений відповідно до кроку 1, може привести до лінійно-залежних напрямків (якщо в одному з напрямків неможливе просування, то одна з компонент $\bar{S}_j$ стає нулем і можуть з'явитися колінеарні напрямки). Тому після виконання кроку 1 проводиться додатковий крок величиною $(X^n - X^0)$, що відповідає повному переміщенню на кроці 1 і приводить в точку $(2X^n - X^0)$. Потім проводиться тест (крок 3), щоб переконатися, чи зменшується визначник матриці напрямків пошуку після включення нового напрямку і відкидання старого.
3. Нехай Δ - максимальне зменшення в будь-якому напрямку пошуку на кроці 1.

$$\Delta = \max_{i=1,...,n} \left\{ f(X^{i-1}) - f(X^i) \right\}.$$

Напрямок пошуку, відповідний Δ, позначимо через S_m. Якщо

$$f(2X^n - X^0) \geq f(X^1)$$

або

$$(f(X^1) - 2f(X^n) + f(2X^n - X^0))(f(X^1) - f(X^n) - \Delta) \leq \Delta(f(X^0) - (2X^n - X^0))^2$$

то перейти до кроку 1, причому з тими ж самими напрямками $\bar{S}_1, \bar{S}_2, ..., \bar{S}_n$, починаючи пошук із точки X^n або з точки $(2X^n - X^0)$, в залежності від того, в якій точці функція приймає найменше значення.

4. Якщо тест на кроці 3 не задоволений, то визначається мінімум в напрямку $\bar{S} = X^n - X^0$, точка цього мінімуму вибирається в якості нової початкової точки X^0 для кроку 1. Система напрямків змінюється таким чином: напрямок $\bar{S}_m$ (визначений Δ) відкидається і додається напрямок $\bar{S}$, який ставиться не на місце $\bar{S}_m$, а в останній стовпець матриці напрямків.

$$A = (\bar{S}_1 \bar{S}_2, ..., \bar{S}_{m-1} \bar{S}_{m+1}, ..., \bar{S}_n \bar{S})$$

5. Перевірка критерію зупинки: зміна по кожній незалежній змінній менша, ніж задана точність

$$\varepsilon_i,\ i = 1, ..., n \text{ або } \left\| X^n - X^0 \right\| \le 0{,}1\varepsilon .$$

Може здатися нераціональним відкидання самого вдалого напрямку поточної ітерації і встановлення нового перспективного напрямку на останнє місце, замість першого. Однак, неважко побачити, що найвдаліший напрямок, швидше за все вичерпав себе, а новий перспективний напрямок тільки що було використано для одновимірної оптимізації та використовувати його відразу ж немає ніякого сенсу, тому що просування просто не буде.

Обґрунтування методу Пауелла

Теорема

Якщо використовуються спряжені напрямки, то довільна квадратична функція n змінних може бути мінімізована рівно за n кроків, по одному кроку в кожному із спряжених напрямків. При цьому порядок використання напрямків несуттєвий. [7; с. 329]

Метод головних напрямків є частинним випадком методу спряжених напрямків. Отже, ця теорема розповсюджується на метод головних напрямків.

Ґрадієнтні методи

Ґрадієнтні методи відносяться до методів першого порядку. Особливість цих методів – це використання вектору ґрадієнта.

Означення

Ґрадієнтом функції $f(x)$ називається вектор, величина якого визначає максимальну швидкість зміни функції $f(x)$, а напрямок співпадає з напрямком найбільшого зростання цієї функції. Вектор, який протилежний за напрямком ґрадієнту називається *антиґрадієнтом.*

Всі методи першого порядку базуються на ітераційній процедурі, яка реалізується за формулою:

$$X^{(k+1)} = X^{(k)} + \lambda^{(k)} \cdot \nabla f(X^{(k)}),$$

де k - номер ітерації;

X^k – поточне наближення до розв’язку;

$\lambda^{(k)}$ – параметр, який характеризує довжину кроку;

$\nabla f(X^{(k)})$ – напрямок пошуку в n - мірному просторі змінних.

Ґрадієнтний метод – це метод розв'язування задач оптимізації з двічі неперервно диференційованими функціями. Основна ідея методу полягає в побудові послідовності ітераційних точок із врахуванням напрямку ґрадієнта функції, яку досліджуємо.

Метод ґрадієнтного спуску

Постановка задачі

Нехай задано функцію $f(x)$, яка обмежена [6; с.223] знизу на множині R^n і має неперервні частинні похідні у всіх її точках.

Потрібно знайти локальний безумовний мінімум функції $f(x)$ на множині припустимих розв'язків $X = R^n$, так щоб $f(x^*) = \min\limits_{x \in R^n} f(x)$.

Стратегія розв'язування задачі

Стратегія розв'язування задачі полягає в побудові послідовності точок $\left\{x^k\right\}, k = 0,1,2,\dots$ таких, що $f(x^{k+1}) < f(x^k), k = 0,1,2\dots$ Точки послідовності $\left\{x^k\right\}$ обчислюються за формулою

$$x^{k+1} = x^k - \alpha_k \nabla f(x^k),$$

де точка x^0 є початковою точкою, що задається дослідником; також задають величину кроку α_k, яку можна обрати, наприклад, так: на кожному кроці задаємо спочатку $\alpha_0 = 0,5$. Число α_k змінюємо (зменшуємо), якщо при наступних ітераціях функція $f(x^{k+1}) > f(x^k)$, до тих пір, поки не досягнемо критерію виходу.

Побудова послідовності $\left\{x^k\right\}, k = 0,1,2,\dots$ закінчується в точці x^k, для якої $\left\|\nabla f(x^k)\right\| < \varepsilon_1$, де ε_1 – наперед задане число, або, якщо $k \geq M$, M - граничне число ітерацій, або при дворазовому одночасному виконанні нерівностей $\left\|x^{k+1} - x^k\right\| < \varepsilon_2$, $\left|f(x^{k+1}) - f(x^k)\right| < \varepsilon_2$, де ε_2 – наперед задане достатньо мале додатне число.

Отже, для ґрадієнтного методу необхідно задати такі початкові дані: $x^0, \alpha_0, \varepsilon_1, \varepsilon_2, M$.

Алгоритм

1. Задати x^0, $\varepsilon_1 > 0$, $\varepsilon_2 > 0$, граничне число ітерацій M .Знайти ґрадієнт функції в початковій точці x^0: $\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1};\dots;\frac{\partial f(x)}{\partial x_n}\right)^T$
2. Покладемо $k = 0$

3. Обчислити $\nabla f(x^k)$.
4. Перевірити виконання критерію виходу

$$\left\|\nabla f(x^k)\right\| < \varepsilon_1, \text{ де } \left\|\nabla f(x^k)\right\| = \sqrt{\sum_{i=1}^{n}\left(\frac{\partial f(x)}{\partial x_i}\right)^2}:$$

а) якщо критерій виконаний, то $x^* = x^k$;

б) якщо критерій не виконаний, то перейти до кроку 5.

5. Перевірити виконання нерівності $k \geq M$:

а) якщо нерівність виконана, то $x^* = x^k$;

б) якщо ні, то перейти до кроку 6.

6. Задати крок $\alpha = 0.5$ при $k = 0$, якщо функція при наступному кроці не буде зменшуватись, то покласти $\alpha = \dfrac{1}{3}$, за цим же принципом далі зменшувати α.
7. Обчислити $x^{k+1} = x^k - \alpha_k \nabla f(x^k)$.
8. Перевірити виконання умов

$$\left\|x^{k+1} - x^k\right\| < \varepsilon_2, \ \left|f(x^{k+1}) - f(x^k)\right| < \varepsilon_2;$$

а) якщо обидві умови виконані при поточному значенні k, то покласти $k = k - 1$, то $x^* = x^k$;

б) якщо хоча б одна з умов не виконується, то вважати $k = k - 1$ і перейти до кроку 3.

Ґрадієнтний метод має свої недоліки. Для того, щоб рухатися завжди в напрямку ґрадієнта, крок повинен бути невеликим. Тому ітерацій буде багато. Але оскільки на кожному кроці необхідно постійно обчислювати ґрадієнт, то наближення до оптимуму буде сповільнюватись при наближенні до оптимуму. Також цей метод погано працює на яружних функціях.

Вплив величини кроку на ґрадієнтний пошук

Питання вибору величини кроку є досить важливим і в остаточному підсумку визначає ефективність і швидкість збіжності алгоритму [12].

Якщо розмір кроку обраний занадто малим, то рух до оптимуму буде довгим через необхідність обчислення частинних похідних у багатьох точках.

При великому кроці в околі оптимуму можуть виникнути незатухаючі коливання незалежних змінних і знижується точність знаходження екстремуму. При дуже великому кроці є можливими розбіжні коливання.

На рис.1 зображені лінії рівня функції $f(u_1,u_2)$. Процес пошуку при великому h зображений послідовністю точок A_0, A_1, A_2, A_3.

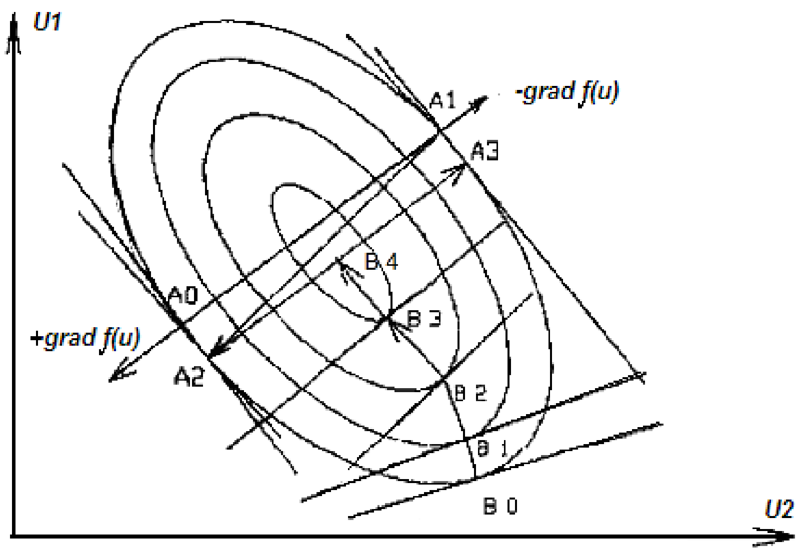


Рис. 1. [12]

Як видно з рис.3 при цьому можуть виникати незатухаючі коливання незалежних змінних. При меншому значенні параметра кроку α процес пошуку зображений послідовністю точок B_0, B_1, B_2, B_3

Також величину кроку α_k можна знайти за допомогою одного з методів одновимірного пошуку. У поточній точці ітерації визначається ґрадієнт і робиться наступний спуск. Цей метод є складнішим за методом координатного спуску, тому що вимагає обчислення похідних і ґрадієнта та переходу до інших змінних.

Критерій закінчення пошуку

Розглянемо одномірну функцію $f(u)$ (рис.2).

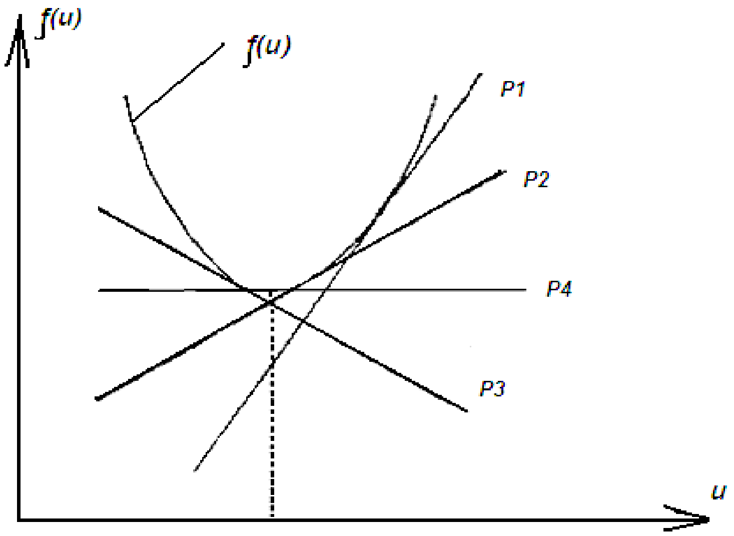


Рис. 2. [12]

Прямі $P_1 \div P_4$- дотичні до функції $f(u)$. В області екстремуму $\frac{df}{du} = 0$, що відповідає дотичній P_4.

Отже, при $\left|\frac{df}{du}\right| \leq \delta$, де δ-мале додатне, наперед задане число, процес пошуку може бути зупинений.

Для багатопараметричної функції умовою зупинки є

$$\sum_{j=1}^{n}\left(\frac{dF}{du_j}\right)^2 \leq \delta.$$

Умова перевіряється на кожному кроці пошуку й виконується, якщо екстремум лежить у допустимій області. Інший варіант визначення моменту закінчення пошуку полягає в наступному: після кожної серії із заданою кількістю кроків s1 запам'ятовується значення цільової функції.

Якщо наступна серія дає менше значення цільової функції, то пошук триває, у іншому випадку - зупинка. Рекомендується також наступний критерій зупинки:

$$f(\vec{u}^{(k+1)}) \geq f(\vec{u}^{(k)}); \quad \left|\vec{u}^{(k+1)} - \vec{u}^{(k)}\right| < \varepsilon_i, \; i = l, 2 \ldots n,$$

де ε_i, - малі, наперед задані величини.

Недоліки ґрадієнтного методу [12]

1. Можна виявити тільки локальний екстремум.
2. Невисока швидкість досягнення екстремуму.
3. Мала швидкість збіжності поблизу екстремальної точки.
4. Залежність методу від масштабу змінних. Якщо гіперпростір витягнутий так, що утвориться яр, то процедура ґрадієнтного методу збігається занадто повільно.
5. Відсутні методи обчислення оптимального значення параметра кроку α.

Чисельне диференціювання

Диференціюванням називається процес [8; с.61] знаходження похідної від заданої функції або ж її чисельного значення в заданій точці.

Необхідність виконання диференціювання виникає досить часто і викликана величезним поширенням цієї операції в сучасній математиці і її додатках. Похідна буває потрібна і сама по собі, як миттєва швидкість тих чи інших процесів, і як допоміжний засіб для побудови більш складних процедур, наприклад, в методі Ньютона для чисельного розв’язку рівнянь і систем рівнянь

В даний час найбільш поширені три способи обчислення похідних:

1) символьне (аналітичне) диференціювання;

2) чисельне диференціювання;
3) автоматичне (алгоритмічне) диференціювання.

Чисельним диференціюванням називається процес знаходження значення похідної від функції, що використовує значення цієї функції в деякому наборі точок її області визначення. Таким чином, якщо функція задана таблицею, тобто лише на скінченній множині значень аргументу, або процедура визначення значень цієї функції є «чорним ящиком» з невідомої структурою, то альтернатив чисельному диференціюванню немає. Зокрема, іноді ми змушені представляти у вигляді «чорного ящика» обчислення значень функції, аналітичне представлення для якої існує, але є занадто складним для диференціювання першими двома способами.

В основі чисельного диференціювання лежать різні ідеї. Найперша полягає в тому, щоб відновити функцію, що задана таблицею, до функції неперервного аргументу, до якої вже може бути застосована звичайна операція диференціювання.

В запропонованій роботі частинні похідні функції двох змінних знайдено за означенням, що є більш зручним в контексті наших задач, ніж, наприклад, інтерполяційний підхід.

Якщо аналітичний вид функції $f(\overrightarrow{u})$ є відомим, то обчислення похідних, які використовуються в алгоритмі ґрадієнтного методу, не викликає утруднень.

Часто $f(\vec{u})$ у явному аналітичному виді записати не можна, або ж вид $\frac{df}{du}$ занадто складний.

У таких випадках використовують різницеву апроксимацію похідних

$$\frac{df}{du_i} \cong \frac{\Delta f}{\Delta u_i} = \frac{f(u_1, u_2, ..., u_i + \delta, ...u_n) - f(u_1, ..., u_n)}{\delta},$$

де δ-величина приросту, яка є однаковою для всіх змінних.

Модифікації алгоритмів ґрадієнтного методу

Вибір величини параметра кроку в ґрадієнтному алгоритмі досить важливий. Відомі модифікації ґрадієнтного алгоритму, у яких параметр кроку α змінюється автоматично в процесі пошуку. У випадку погано організованої функції масштабують змінні [12].

Нижче наведені деякі алгоритми з адаптивним параметром кроку.

1. $u_i^{(k+1)} = u_i^{(k)} - h^{(k)} sign \dfrac{\partial f(\vec{u}^{(k)})}{\partial u_i}$;

$$h^{(k+1)} = \begin{cases} h^{(k)}, f(\vec{u}^{(k)}) \le f(\vec{u}^{(k-1)}) \\ -\dfrac{h^{(k)}}{2}, f(\vec{u}^{(k)}) > f(\vec{u}^{(k-1)}) \end{cases};$$

2. $u_i^{(k+1)} = u_i^{(k)} - h^{(k)} \dfrac{\partial f(\vec{u}^{(k)})}{\partial u_i}$;

$$h^{(k+1)} = \begin{cases} 2h^{(k)}, a^{(k)} < a_{\min} \\ h^{(k)}, a_{\min} \le a^{(k)} \le a_{\max} \\ \dfrac{h^{(k)}}{3}, a^{(k)} > a_{\max} \end{cases};$$

$$\cos a^{(k)} = \frac{\sum_i \dfrac{\partial f(\vec{u}^{(k)})}{\partial u_i} \cdot \dfrac{\partial \vec{u}^{(k-1)}}{\partial u_i}}{\sqrt{\sum_i \left(\dfrac{\partial f(\vec{u}^{(k)})}{\partial u_i}\right)^2 \cdot \sum_i \left(\dfrac{\partial f(\vec{u}^{(k-1)})}{\partial u_i}\right)^2}},$$

де a^k - кут повороту ґрадієнта на k-тому кроці пошуку, причому

$\cos(a_{\min}) \approx 0.2$, $\cos(a_{\max}) \approx 0.8$.

Застосування ґрадієнтних методів оптимізації функцій багатьох змінних у випадках яружних, улоговинних та невпорядкованих рельєфах функцій

Не існує універсального методу пошуку екстремуму функції багатьох змінних. Для різних видів функцій рекомендовано різні оптимальні методи знаходження екстремуму.

При розв'язуванні чисельними методами задач на оптимізацію функції в результаті отримуємо точку, яка до певної міри відповідає умовам екстремуму, але достеменно не відомо, чи в дійсності ця точка буде екстремальною, і якщо вона знайдена, то нам не відомо чи це є точка глобального чи локального екстремуму, чи вона є єдиною. Загалом, результат знаходження екстремуму функції багатьох змінних суттєво залежить від обрання початкової точки. Наскільки відомо, не існує загальновідомих методів чи рекомендацій для задання такої точки. Також не існує універсальних критеріїв виходу із алгоритму оптимізації. Особливо це проявляється при відшуканні екстремуму в випадках яружних, улоговинних та на невпорядкованих рельєфах функцій**.**

Постановка задачі

В топографії можна зобразити рельєф поверхні [9; с.43] лініями рівня. За видом ліній рівня умовно виділяють три типи рельєфу: улоговинний, яружний і неврегульований. Два останні типи рельєфу шляхом символьних перетворень можна перевести до улоговинного типу. При улоговинному рельєфі лінії рівнів подібні еліпсам. В малому околі невиродженого мінімуму рельєф функції є улоговинним. Це видно з наступних міркувань. У мінімуму перші похідні функції $F(x_m,\ y_m)$ дорівнюють нулю

$$\frac{\partial F}{\partial x} = \frac{\partial F}{\partial y} \ = \ 0, (1)$$

розвинення функції в ряд Тейлора поблизу мінімуму має вигляд :

$$F(x,y) = F(x_m, y_m) + \frac{1}{2}(\Delta x)^2 F_{xx}(x_m, y_m) + \Delta x \Delta y F_{xy}(x_m, y_m) +$$
$$+ \frac{1}{2}(\Delta y)^2 F_{yy}(x_m, y_m) + \ldots$$

де $\Delta x = x - x_m$, $\Delta y = y - y_m$. Тому що квадратична форма – додатньо визначена, а лінії рівня знакозмінної квадратичної форми - це еліпси. Яружний тип рельєфу маємо, коли лінії рівня шматково-гладкі, тобто мають точки зламу.

Геометричне місце точок зламу називається істинним яром, якщо кут напрямлений в сторону зростання функції, і гребенем – якщо в сторону зменшення. Найчастіше лінії рівня всюди гладкі, але на них є ділянки із значною кривиною - розв'язні яри або гребені. Наприклад,

$$f(x, y) = 10(y - \sin x)^2 + 0.2x^2$$

У фізичних задачах яружних рельєф вказує на те, що не врахована якась закономірність, що є наявною між змінними.

Невпорядкований тип рельєфу характеризується наявністю багатьох максимумів, мінімумів і сідловин. Наприклад,

$$f(x, y) = (2 + \sin 2x)(2 + \sin 2y)\,.$$

Усі ефективні методи пошуку мінімуму багатовимірної функції зводяться до побудови траєкторії, вздовж якої функція зменшується. Різні методи відрізняються лише засобами побудови цих траєкторій. Здавалося б, для знаходження мінімуму достатньо розв’язати систему рівнянь типу (1) і відкинути ті розв’язки, які є сідловими або максимумами. Проте в реальних задачах мінімізації методи розв’язання таких систем, як правило, є збіжними у настільки малому околі мінімуму, що вибрати відповідне нульове наближення не завжди вдається.

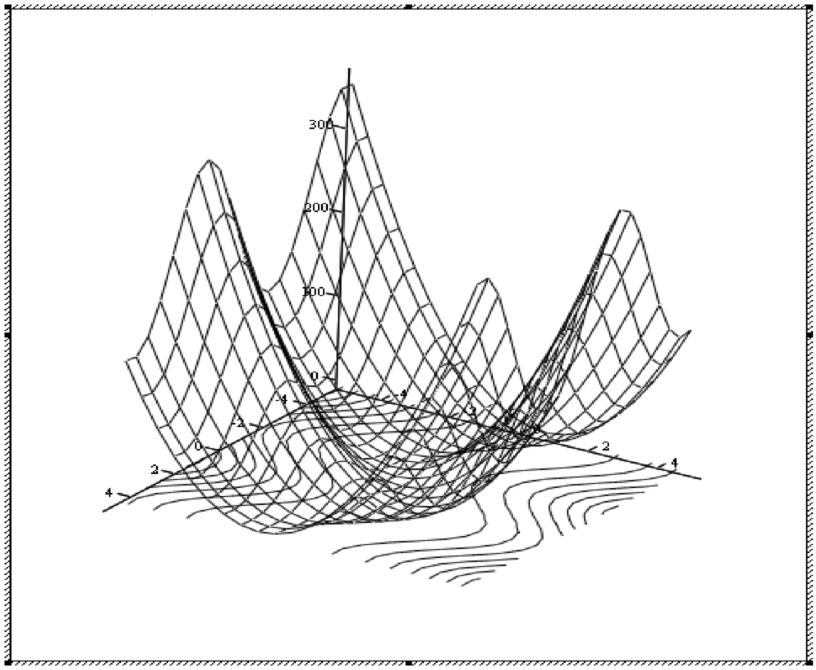


Рисунок 1 − Яружний рельєф

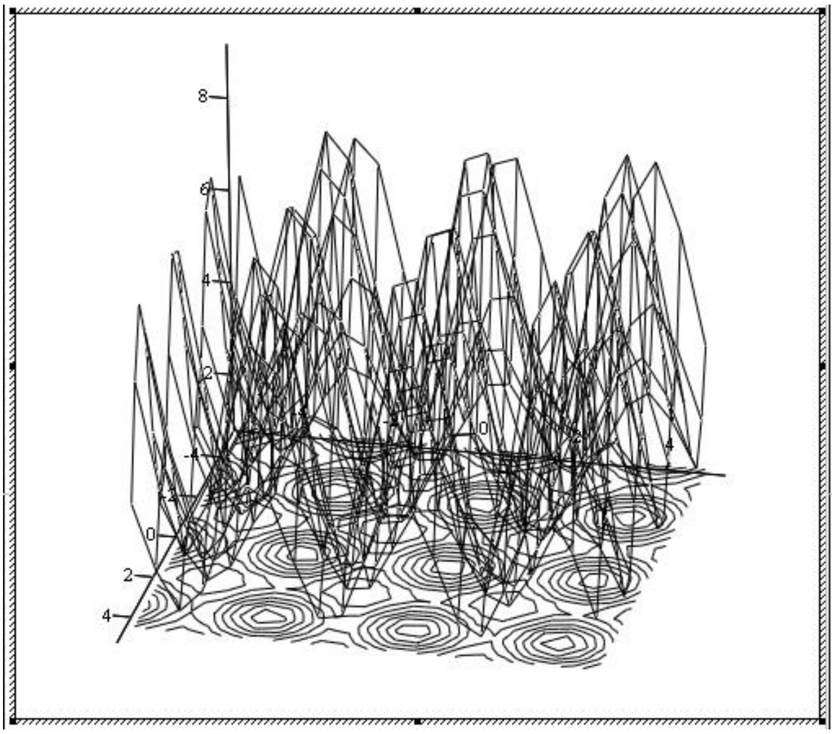


Рисунок 2 – Невпорядкований рельєф

Частина 3. Геометричний підхід до задач оптимізації.

Теоретичні відомості

Для поверхонь існує поняття *стичного параболоїда.* Це параболоїд, для якого нормаль до поверхні в даній точці є його віссю і який має дотик порядку $n \geq 2$ з поверхнею в цій точці. В кожній точці двічі диференційованої поверхні існує єдиний [2; с.52] стичний параболоїд, який може вироджуватися в параболоїд, циліндр чи площину. Якщо поверхню віднести до прямокутних декартових координат, взявши дану точку поверхні за початок координат, а стичну площину в ній за площину XY, то рівняння поверхні в околі точки дотику буде

$$z = f(x, y),$$

а рівняння стичного параболоїда в цій точці

$$z = \frac{1}{2}(f_{xx}x^2 + 2f_{xy}xy + f_{yy}y^2)$$

(похідні функції f беруться в точці дотику).

В залежності від вигляду стичного параболоїду точки поверхні поділяються на *еліптичні, гіперболічні, параболічні та точки сплощення.* Значення стичного параболоїду полягає в тому, що він відтворює форму поверхні з точністю до нескінченно малих 2-го порядку (стична поверхня відтворює її форму з точністю до нескінченно малих 1-го порядку).

За допомогою стичного параболоїду вводиться поняття *спряжених напрямків* на поверхні. Два напрямки на поверхні в даній точці називаються *спряженими*, якщо прямі, що їх містять, спряжені відносно стичного параболоїда в цій точці. Ортогональні спряжені напрямки називаються *головними.* В даній точці поверхні, як правило, два головних напрямки. Винятком є точки сплощення [2; с.53] і спеціальні еліптичні точки (точки округлення, або омбілічні), в яких кожен напрямок є головним. Лінія, у якої в кожній точці напрямок є головним, називається *лінією кривини.* В точках поверхні, які не є еліптичними, існують самоспряжені напрямки. Вони називаються *асимптотичними напрямками.* Лінія на поверхні, напрямок якої в кожній точці асимптотичний, називається *асимптотичною лінією.*

Базові поняття для кривих 2-го порядку

Розглянемо криву 2-го порядку Γ, задану в афінній [4; с.84] системі координат рівнянням

$$F(x,y) = a_{11}x^2 + 2a_{12}xy + a_{22}y^2 + 2a_1x + 2a_2y + a_0 = 0 \quad (1)$$

і пряму l, яка задана параметрично

$$\begin{cases} x = x_0 + \alpha t \\ y = y_0 + \beta t \end{cases}.$$

Для знаходження точок перетину Γ і l підставимо параметричне рівняння в $F = 0$:

$$F_2t^2 + 2F_1t + F_0 = 0,$$

де

$$F_2 = q(\alpha,\beta) = a_{11}\alpha^2 + 2a_{12}\alpha\beta + a_{22}\beta^2,$$

$$F_1 = \alpha(a_{11}x_0 + a_{12}y_0 + a_1) + \beta(a_{12}x_0 + a_{22}y_0 + a_2),$$

$$F_0 = F(x_0, y_0).$$

Означення

Ненульовий вектор (α,β) має *асимптотичний напрямок* щодо кривої Γ, якщо

$$F_2 = q(\alpha,\beta) = a_{11}\alpha^2 + 2a_{12}\alpha\beta + a_{22}\beta^2 = 0.$$

Знову розглянемо криву 2-го порядку Γ (1) і пряму l неасимптотичного напрямку ($F_2 \neq 0$), яка задана параметрично

$$\begin{cases} x = x_0 + \alpha t \\ y = y_0 + \beta t \end{cases}.$$

Означення

Пряма $\alpha(a_{11}x + a_{12}y + a_1) + \beta(a_{12}x + a_{22}y + a_2) = 0$ називається *діаметром* кривої 2-го порядку (1) *спряженим даному неасимптотичному напрямку* (α,β). [4; с.89]

Означення

Два діаметри кривої з єдиним центром, що ділять навпіл хорди, паралельні іншому діаметрові, називаються *спряженими діаметрами.* [4; с.92]

Означення

Два напрямки (α, β) і (α^*, β^*) називаються *спряженими напрямками* (відносно даної кривої), якщо вони задовольняють рівняння

$$\alpha(a_{11}\alpha^* + a_{12}\beta^*) + \beta(a_{12}\alpha^* + a_{22}\beta^*) = (\alpha, \beta)\mathrm{A}\begin{pmatrix} \alpha^* \\ \beta^* \end{pmatrix} = 0 .$$

Означення

Неасимптотичний напрямок, що перпендикулярний спряженому йому діаметру, називається *головним напрямком* кривої Γ, а діаметр називається *головним діаметром.* [4; с.94]

Теорема

Вектор (α, β) задає головний напрямок кривої (1), тоді і тільки тоді, коли він є власним вектором матриці

$$Q = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix},$$

що відповідає [4; с.95] ненульовому власному значенню, тобто ненульовому кореню рівняння

$$\det(Q - \lambda E) = \lambda^2 - S\lambda + \delta = 0 .$$

Базові поняття для поверхонь 2-го порядку

Розглянемо поверхню 2-го порядку, що задана в афінній системі координат рівнянням

$$F(x, y, z) = a_{11}x^2 + a_{22}y^2 + a_{33}z^2 + 2a_{12}xy + 2a_{13}xz + 2a_{23}yz + 2a_1x + 2a_2y + 2a_3z + a_0 = 0, \quad (1)$$

де $a_{11}x^2 + a_{22}y^2 + a_{33}z^2 + 2a_{12}xy + 2a_{13}xz + 2a_{23}yz =: q(x, y, z)$ - квадратична частина, $2a_1x + 2a_2y + 2a_3z =: l(x, y, z)$ - однорідна лінійна частина. [4; с.119]

При цьому квадратична частина має не дорівнювати нулю.

Введемо позначення $Q := \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{pmatrix}$.

Означення

Ненульовий вектор (α, β, γ) задає *асимптотичний напрямок* для поверхні $F = 0$, якщо

$$q(\alpha, \beta, \gamma) = a_{11}\alpha^2 + a_{22}\beta^2 + a_{33}\gamma^2 + 2a_{12}\alpha\beta + 2a_{13}\alpha\gamma + 2a_{23}\beta\gamma = 0 .$$

Розглянемо рівняння (1) і введемо позначення (частинні похідні):

$$F_x = 2(a_{11}x + a_{12}y + a_{13}z + a_1),$$

$$F_y = 2(a_{12}x + a_{22}y + a_{23}z + a_2),$$

$$F_z = 2(a_{13}x + a_{23}y + a_{33}z + a_3).$$

[4; с. 131]

Означення

Площина $\alpha F_x + \beta F_y + \gamma F_z = 0$ називається *діаметральною площиною спряженою неасимптотичному напрямку* (α, β, γ). [4; с.136]

Означення

Неасимптотичний напрямок називається *головним,* якщо спряжена йому діаметральна площина перпендикулярна йому. [4; с.138]

Теорема

Головні напрямки поверхні співпадають із власними векторами матриці Q квадратичної частини її рівняння. [4; с.138]

Теорема

Площина, спряжена головному напрямку, є площиною симетрії поверхні. [4; с.138]

Кривина поверхні

Нехай поверхня $\Phi \in C^1$ і $\vec{r} = \vec{r}(u,v)$ – її гладка параметризація. Квадрат диференціала $ds^2 = d\overrightarrow{r^2} = (\overrightarrow{r_u}du + \overrightarrow{r_v}dv)^2 = \overrightarrow{r_u^2}du^2 + 2(\overrightarrow{r_u} + \overrightarrow{r_v})dudv + \overrightarrow{r_v^2}dv^2$ є квадратичною формою, заданої на сукупності векторів $\overrightarrow{dr} = \overrightarrow{r_u}du + \overrightarrow{r_v}dv$ в дотичній площині поверхні, і називається *першою квадратичною формою поверхні.* Коефіцієнти цієї [3;с.74] форми позначаються відповідно $\overrightarrow{r_u^2} = E$, $(\overrightarrow{r_u} + \overrightarrow{r_v}) = F$, $\overrightarrow{r_v^2} = G$. Перша квадратична форма поверхні є *додатньо визначеною* квадратичною формою.

Якщо поверхня задана рівнянням $z = z(x,y)$, то

$$E = 1 + z_x^2,\ F = z_x z_y,\ G = 1 + z_y^2.$$

Внутрішня геометрія поверхні визначається її першою квадратичною формою.

Якщо поверхня представляє собою нерозтяжну, але гнучку плівку, то при її згинанні зберігається її внутрішня [3; с.75] геометрія, хоча просторова форма змінюється. Отже, перша квадратична форма поверхні визначає її з точністю до згинання.

Для однозначного визначення поверхні з точністю до руху і дзеркального відображення в просторі необхідно задати ще її другу квадратичну форму. *Другою квадратичною формою поверхні* називають вираз $(d^2\vec{r}, \vec{m}) = Ldu^2 + 2Mdudv + Ndv^2$, де $L = (\overrightarrow{r_{uu}}, \vec{m})$, $M = (\overrightarrow{r_{uv}}, \vec{m})$, $N = (\overrightarrow{r_{vv}}, \vec{m})$, а $\vec{m}(u,v)$ – нормаль до поверхні. Враховуючи що $\vec{m} = \dfrac{\left|\overrightarrow{r_u}, \overrightarrow{r_v}\right|}{\left|\left[\overrightarrow{r_u}, \overrightarrow{r_v}\right]\right|}$, маємо

$$L=\frac{(\overrightarrow{r_u},\overrightarrow{r_v},\overrightarrow{r_{uu}})}{\sqrt{EG-F^2}},\ M=\frac{(\overrightarrow{r_u},\overrightarrow{r_v},\overrightarrow{r_{uv}})}{\sqrt{EG-F^2}},\ N=\frac{(\overrightarrow{r_u},\overrightarrow{r_v},\overrightarrow{r_{vv}})}{\sqrt{EG-F^2}}.$$

Якщо поверхня задана рівнянням $z=z(x,y)$, то

$$L=\frac{z_{xx}}{\sqrt{1+z_x^2+z_y^2}},\ M=\frac{z_{xy}}{\sqrt{1+z_x^2+z_y^2}},\ N=\frac{z_{yy}}{\sqrt{1+z_x^2+z_y^2}}.$$

Нормальним перетином поверхні Φ в точці P називають лінію перетину поверхні з площиною, що проходить через нормаль [3; с.76] поверхні в цій точці. Кривина нормального перетину в точці P називається *нормальною кривиною поверхні* в цій точці в напрямку, дотичному до нормального перетину. Нормальна кривина обчислюється за формулою

$$k_H=\frac{Ldu^2+2Mdudv+Ndv^2}{Edu^2+2Fdudv+Gdv^2}.$$

Для довільної точки $P\in\Phi$ має місце наступне *твердження*: або існує два взаємно ортогональних напрямки, які називаються головними, в яких k_H має найбільше і найменше значення k_1 і k_2, або кривина всіх нормальних перетинів однакова, тобто $k_H=k_1=k_2$. В останньому випадку точка називається *сферичною* при $k_H\neq 0$ і *точкою сплощення* при $k_H=0$. Всі напрямки в сферичній точці або точці сплощення вважаються *головними*. Величини k_1 и k_2 називають *головними кривинами поверхні в точці* P.

Гауссовою кривиною поверхні називається величина

$$K=k_1k_2.$$

Середньою кривиною поверхні називається величина

$$H=\frac{1}{2}(k_1+k_2).$$

Залежно від значень $K(u,v)$ і $H(u,v)$ точки поверхні класифікують наступним чином.

Точка [3; с.77] називається *еліптичною*, якщо в ній $K>0$;

гіперболічною - якщо $K<0$;

параболічною – якщо $K=0,\ H\neq 0$;

точкою сплощення – якщо $K=H=0$.

Мають місце формули

$$K = \frac{LN - M^2}{EG - F^2}, \ H = \frac{EN - 2MF + GL}{2(EG - F^2)}.$$

Для задач оптимізації важливими є такі властивості нормальних кривин поверхні:

1) нормальна кривина поверхні в асимптотичному напрямку дорівнює нулю;
2) нормальна кривина поверхні по головним напрямкам досягає екстремальних значень. [5; с.144]

Обчислення головних кривин. Формули

$$\begin{vmatrix} L - Ek_n & M - Fk_n \\ M - Fk_n & N - Gk_n \end{vmatrix} = 0 \Rightarrow (EG - F^2)k_n^2 + (2MF - EN - LG)k_n + (LN - M^2) = 0.$$ [5; с.149]

$$(EG - F^2)k^2 + (2MF - EN - LG)k + (LN - M^2) = 0.$$

$$D = (2MF - EN - LG)^2 - 4(EG - F^2)(LN - M^2).$$

$$k_1 = \frac{-2MF + EN + LG + \sqrt{(2MF - EN - LG)^2 - 4(EG - F^2)(LN - M^2)}}{2(EG - F^2)};$$

$$k_2 = \frac{-2MF + EN + LG - \sqrt{(2MF - EN - LG)^2 - 4(EG - F^2)(LN - M^2)}}{2(EG - F^2)}.$$

Індикатриса Дюпена

Нехай в околі точки P гладка поверхня не є частиною площини, тобто виключаємо випадок $L = M = N = 0$.

У дотичній площині в точці P поверхні розглянемо в’язку прямих з центром P. На кожній з прямих цієї в’язки від точки P по обидва боки відкладемо відрізки довжиною $\frac{1}{\sqrt{|k_n|}}$, де k_n – нормальна кривина ліній на поверхні, що не дорівнює нулю, для яких дана пряма є дотичною.

Лінія, утворена кінцями відкладених таким способом відрізків, називається *індикатрисою кривини поверхні або індикатрисою Дюпена* в точці P.

Розглянемо [2; с.55] інший підхід до означення індикатриси Дюпена. Нехай задано рівняння поверхні $z = z(x, y)$, це еквівалентно параметричному заданню

$$x = u,\ y = v,\ z = z(u, v).$$

Візьмемо точку O поверхні за початок координат, а стичну поверхню в ній - за площину XY. Тоді поверхня в околі точки O задається рівнянням

$$z = \frac{1}{2}(rx^2 + 2sxy + ty^2) + \varepsilon(x, y)(x^2 + y^2),$$

де $\varepsilon(x, y) \to 0$ при $x, y \to 0$, p, q, r, s, t - позначення для перших і других похідних функції $\mathrm{z}(x, y)$: $\mathrm{p} = \mathrm{z}_{x,}$ $q = \mathrm{z}_{\mathrm{y},}$ $r = \mathrm{z}_{xx,}$ $s = \mathrm{z}_{xy,}$ $t = \mathrm{z}_{\mathrm{yy}}$.

Стичний параболоїд в точці O поверхні –

$$z = \frac{1}{2}(rx^2 + 2sxy + ty^2).$$

Індикатриса Дюпена в точці O -

$$rx^2 + 2sxy + ty^2 = \pm 1.$$

Перша і друга квадратична форми поверхні і її стичний параболоїд в точці O однакові. Перша квадратична форма –

$$dx^2 + dy^2,$$

друга квадратична форма –

$$r\,dx^2 + 2s\,dxdy + t\,dy^2.$$

Отже, у поверхні і стичного параболоїду нормальна кривина в одному і тому ж напрямку однакова. А саме

$$k_n = \frac{r\,dx^2 + 2s\,dxdy + t\,dy^2}{dx^2 + dy^2}.$$

Вид індикатриси залежить від дискримінанту її рівняння. Можливі випадки:

1. $LN - M^2 > 0$.

   Індикатриса розпадається на еліпс і уявний еліпс. Точка P називається точкою *еліптичного типу*. Окремим випадком точок еліптичного типу є омбілічна точка, в якій індикатриса Дюпена розпадається на два кола. У точці еліптичного типу немає асимптотичних напрямків.

2. $LN - M^2 < 0$.

   Індикатриса є парою спряжених гіпербол. Точка називається точкою *гіперболічного типу*. У точці гіперболічного типу існують два асимптотичних напрямки.

3. $LN - M^2 = 0$.

   Індикатриса є лінією параболічного типу, що має центр симетрії. Точка називається точкою *параболічного типу*. У точці параболічного типу існує один асимптотичний напрямок.

Геометрична інтерпретація квадратичної апроксимації стичним параболоїдом

Стичний параболоїд можна записати в новій системі координат шляхом переносу початку системи координат в точку екстремуму функції і наступного повороту осей для отримання симетрії. Позначимо початкові координати $x_1, x_2, ..., x_n$, а нові координати через $\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_n$ (їх називають головними осями). Власне кажучи, вони визначають головні напрямки індикатриси Дюпена. Це дозволяє отримати вираз для цільової функції $f(x)$ в канонічному вигляді відносно координат $x_1, x_2, ..., x_n$. Для функції двох змінних квадратична апроксимація має вигляд

$$f(\mathrm{x}_1, \mathrm{x}_2) = \mathrm{b}_0 + b_1 x_1 + b_2 x_2 + b_{11} x_1^2 + b_{12} x_1 x_2 + b_{22} x_2^2 \;,$$

це рівняння зводиться до канонічного виду

$$f(\tilde{\mathrm{x}}_1, \tilde{\mathrm{x}}_2) = \tilde{b}_{11} \tilde{x}_1^2 + \tilde{b}_{22} \tilde{x}_2^2 + f(x_1^*, x_2^*) \,,$$

де $f(x_1^*, x_2^*)$ - значення $f(x)$ в точці мінімуму x^*. Характер функції $f(x)$ в залежності від знаків $\tilde{b}_{11}$, $\tilde{b}_{22}$ і співвідношень між ними представлено у вигляді таблиці.

Таблиця [7; с. 328]

№	Співвідношення між головними кривинами поверхні, гауссова та середня кривини	Знаки		Тип лінії індикатриси Дюпена	Геометрична інтерпретація околу точки поверхні	Характеристика центральної точки
1	$k_1 = k_2$	-	-	Коло	Круговий пагорб	Максимум
2	$k_1 = k_2$	+	+	Коло	Кругова западина	Мінімум
3	$k_1 > k_2$, $K > 0$	-	-	Еліпс	Еліптичний пагорб	Максимум
4	$k_1 > k_2$, $K > 0$	+	+	Еліпс	Еліптична западина	Мінімум
5	$\lvert k_1 \rvert > \lvert k_2 \rvert$, $K < 0$	+	-	Гіпербола	Симетричне сідло	Сідлова точка

6	$\lvert k_1 \rvert = k_2$, $K < 0$	-	+	Гіпербола	Симетричне сідло	Сідлова точка
7	$k_1 > \lvert k_2 \rvert$, $K < 0$	+	-	Гіпербола	Витягнуте сідло	Сідлова точка
8	$k_2 = 0$, $K = 0$	-		Пряма	Стаціонарний хребет (вироджена поверхня)	Нема
9	$k_2 = 0$, $K = 0$	+		Пряма	Стаціонарний яр (вироджена поверхня)	Нема
10	$k_2 = 0$, $K = 0, H \neq 0$	-		Парабола	Хребет, що підіймається (вироджена поверхня)	на ∞
11	$k_2 = 0$, $K = 0, H \neq 0$	+		Парабола	Яр, що спадає (вироджена поверхня)	на ∞

Лінії кривини

Лінією кривини поверхні називається така лінія , у якій в кожній точці дотична напрямлена по головному напрямку.

Так як в кожній точці поверхні є два головних напрямки, то ми будемо мати два сімейства ліній кривини на поверхні, і ці сімейства будуть взаємно ортогональні. Отже, сукупність всіх ліній кривини дасть деяку ортогональну сітку на поверхні. Рівняння

$$(EM - FL)du^2 + (EN - GL)dudv + (FN - GM)dv^2 = 0$$

або еквівалентні рівняння

$$\frac{dv}{du} = \varphi_1(u,v), \ \frac{dv}{du} = \varphi_2(u,v)$$

- *диференціальні рівняння ліній кривини.* Інтегруючи їх, ми знайдемо v через u і, підставляючи цей вираз в рівняння поверхні, отримаємо рівняння ліній кривини.

Суть подальшої роботи полягає в тому, щоб координати точок ітерації при обчисленні екстремуму функції двох змінних обчислювати слідуючи в головних напрямках індикатриси Дюпена, тобто по лініям кривини. Це дасть можливість зменшити кількість ітерацій у порівнянні з ґрадієнтними методами. Цей метод є більш ефективним для поверхонь яружного типу.

Частина 4. Кривини поверхні та оптимізація функції двох змінних

Далі наводимо приклади, розв'язані методом Хука-Дживса та ґрадієнтним методом. Метод Хука-Дживса реалізовано в PascalABC (Додаток 1,2, написано програму), ґрадієнтний метод реалізовано в Excel. Використання Excel обумовлено тим, що було простіше в ньому реалізовувати обчислення кривин поверхні в кожній поточній точці, та аналізувати отримані результати.

В прикладах співставляються результати отримані цими методами. Тому початкові точки в реалізації цих методів обрані однаковими.

В методі спряжених напрямків Пауелла закладена основна ідея, шукати кожну поточну точку в напрямку мінімуму функції по спряженим напрямкам. Ці спряжені напрямки відповідають спряженим напрямкам індикатриси Дюпена в поточних точках. По суті, теореми обґрунтування методу Пауелла, доводять необхідність у кожній поточній точці шукати головні напрямки та далі іти по цим напрямкам. Це викладено за допомогою матричного числення. В нашій роботі надається геометрична інтерпретація цього підходу до чисельних задач оптимізації.

Приклади

Приклад 1

$$z(x,y) = x^4 + 2y^4 + x^2y^2 + 2x + y$$

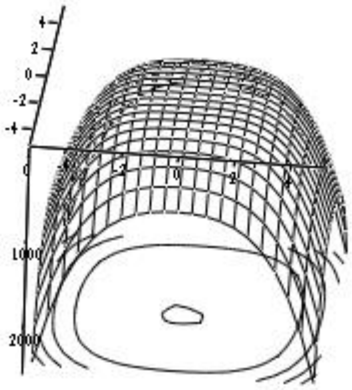


Рис.1

На рис. 1 представлена поверхня та її лінії рівня. Ця поверхня має однакове найменше значення не в одній точці, а в безлічі точок:

x	-0,75884	-0,75902	-0,75911	-0,75917	…
y	-0,40617	-0,40579	-0,40559	-0,40547	…

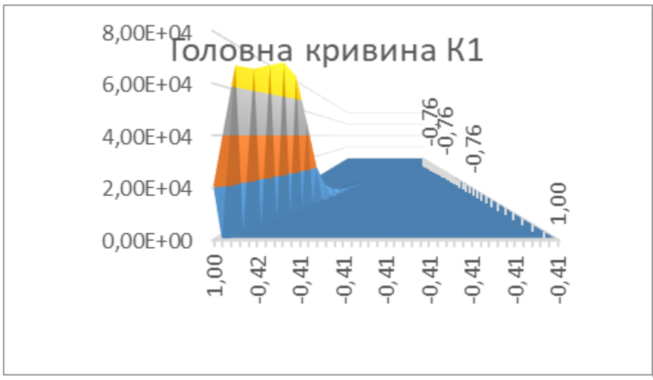
У всіх цих точках найменше значення функції дорівнює (-1,44283)

За допомогою EXCEL обчислено наступне:

- гауссова кривина в цих точках спадає від К=1354,26 до К=53,24;
- середня кривина в цих точках спадає від Н=97,26 до Н=7,40;
- $k_1 > k_2$ (k_1 суттєво більша за k_2) у всіх поточних точках, згідно з таблицею 1;
- всі поточні точки поверхні в околі мінімуму є еліптичними;
- частинні похідні в точках зменшуються від $z'_x = 0,001869$; $z'_y = 0,00195$ до $z'_x = 7,25E-05$; $z'_y = 7,56E-05$.

Отже функція досягає свого мінімуму не в одній точці, а в деякій області, як видно з рис.1.

Графіки головних кривин в околі цих точок характеризують рельєф поверхні в напрямках ліній кривини:



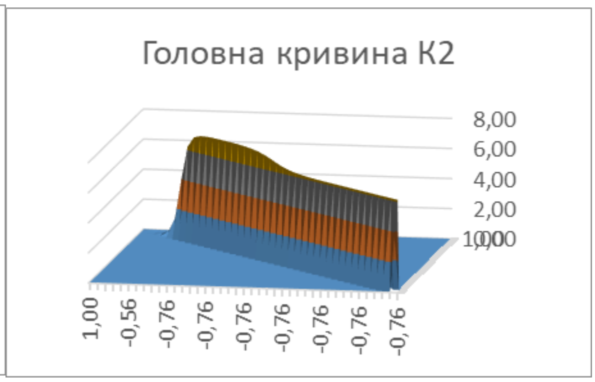


Рис.2

Гауссова та середня кривини характеризують нормальні перерізи поверхні:

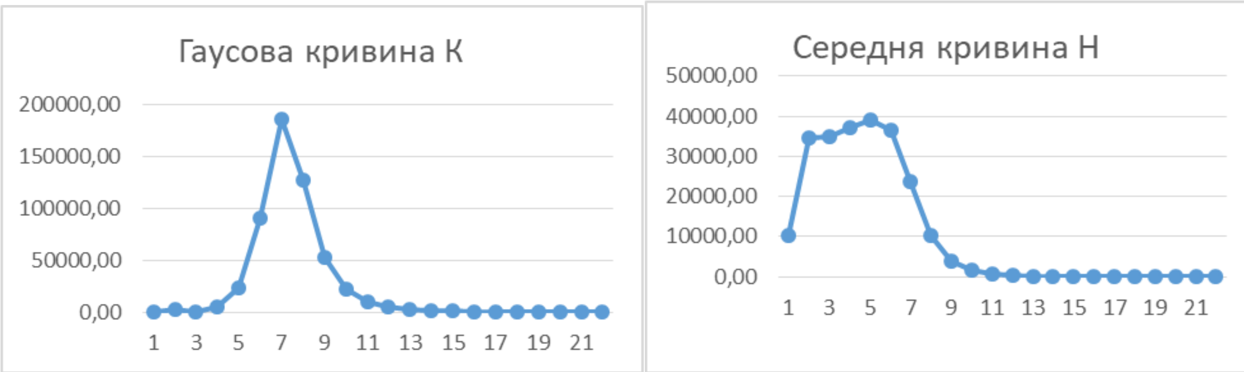
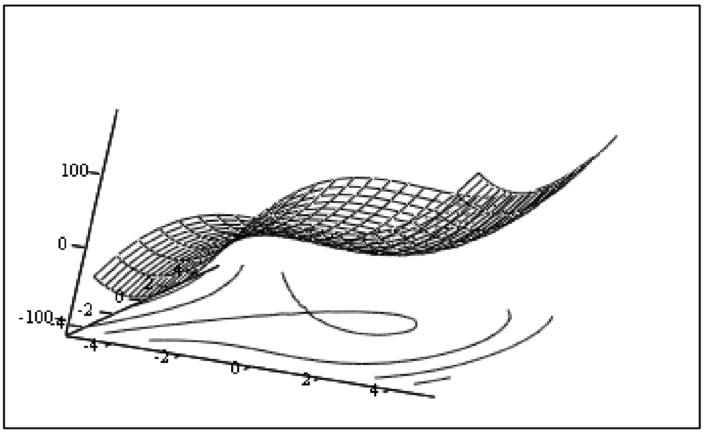


Рис. 3

Цей приклад також розв'язано методом Хука-Дживса (Додаток 2). Результат розв'язування: $z(-0,75;-0,4375) = -1.44015502929688$. Як бачимо результати співпадають, щоправда, додаткові дослідження дозволили отримати про точку мінімуму більше інформації, ніж просте розв'язування стандартним методом.

Приклад 2

$$z(x,y) = x^3 + 2y^2 - 3x - 4y$$



z, z

Рис. 4

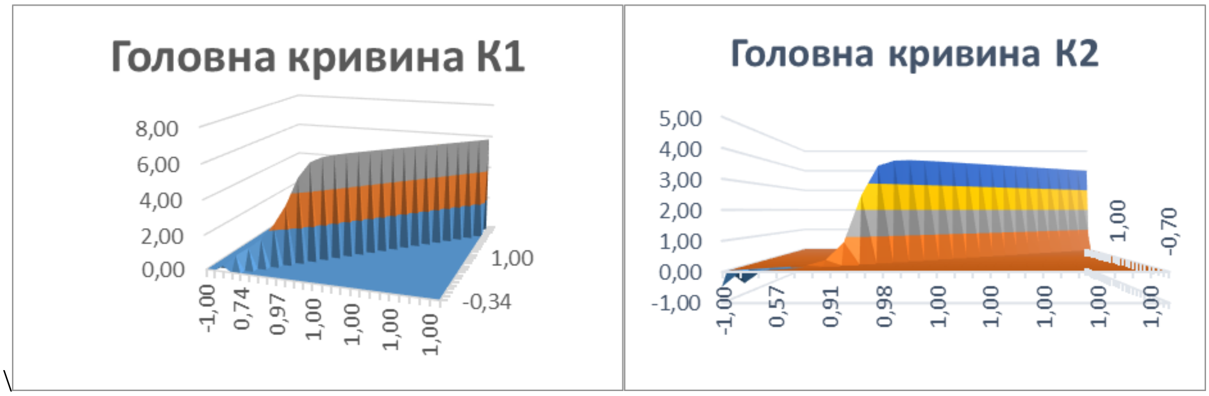


Рис.5

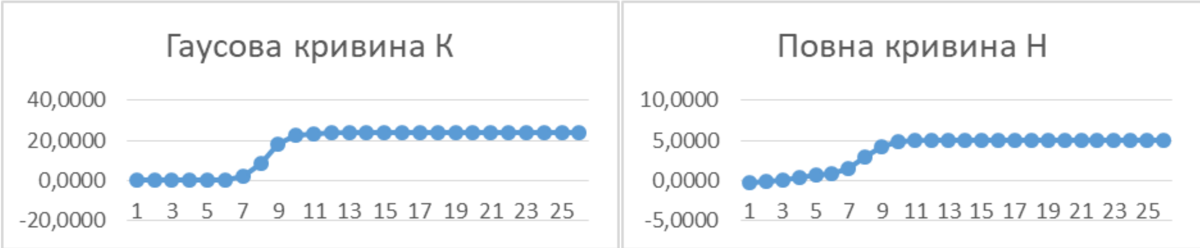


Рис.6

Мінімум функції дорівнює (-4) і досягається в точці (0,999;0,999). В цій точці $K = 24,\ H = 5,\ k_1 = 6,\ k_2 = 3{,}99.$ Отже точка мінімуму є еліптичною. В той же час по зміні у поточних точках кривин, можна сказати, що поверхня має яружний характер:

Таблиця 2

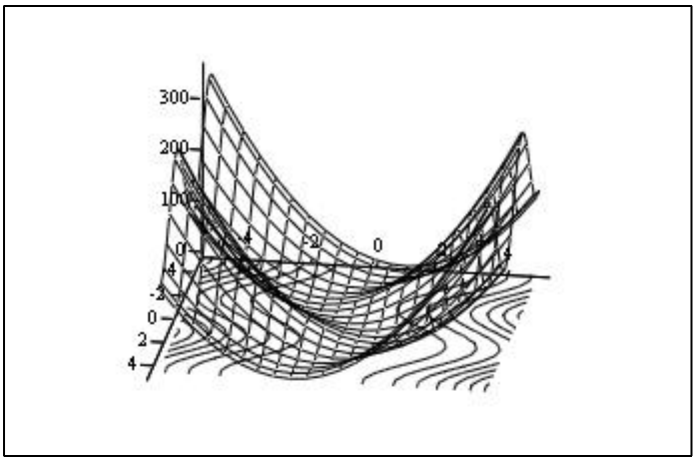
№	К(гаус.кр)	Н(сер.кр)	k_1	k_2
1	-0,0037	-0,2349	0,007757	-0,47759
2	-0,0162	-0,1884	0,038977	-0,41583
3	-0,0302	0,1021	0,303513	-0,09938
4	-0,0102	0,4097	0,83169	-0,01228
5	0,0529	0,6047	1,163932	0,045438
6	0,3112	0,8839	1,569588	0,198282
7	1,8794	1,5594	2,302691	0,816178

З таблиці 2 видно, що на поверхні є точки гіперболічного типу (№ 1-4), а в їх околі поверхня набуває сідлового вигляду, також є точки в околі яких маємо еліптичні западини (№ 5-7).

Метод Хука-Дживса (Додаток 2) дає такі результати: $z(0{,}9875;1) = -3{,}999533$. Відповіді практично співпадають.

Приклад 3

$$z(x,y)=10(y-\sin x)^2+0,2x^2$$



p , p

Рис.7

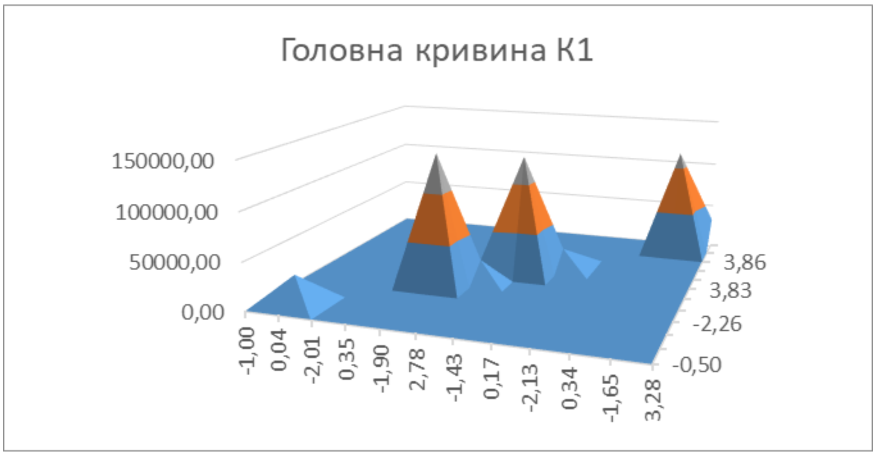


Рис.8

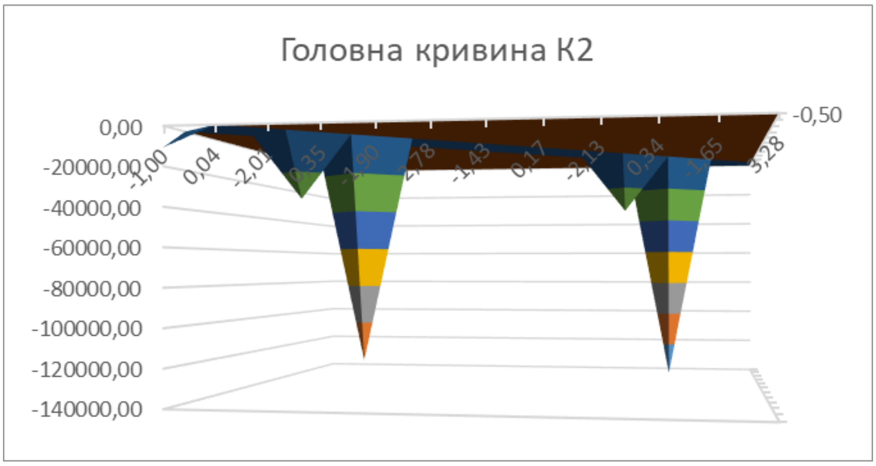


Рис.9

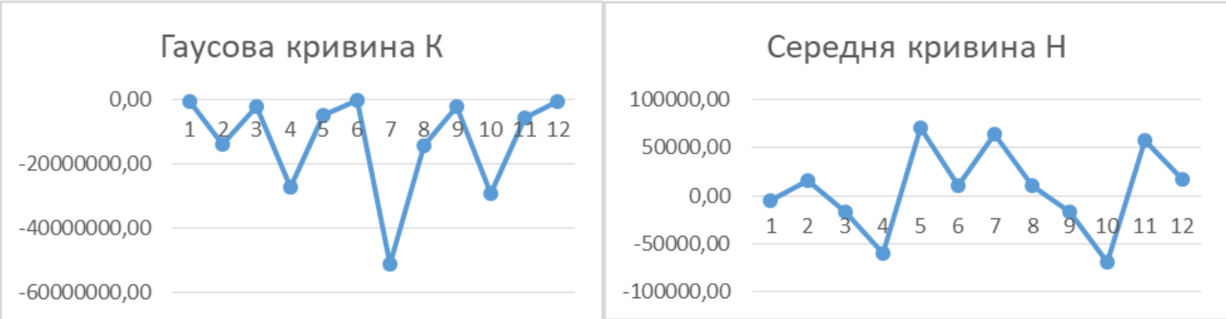


Рис.10

Таблиця 3

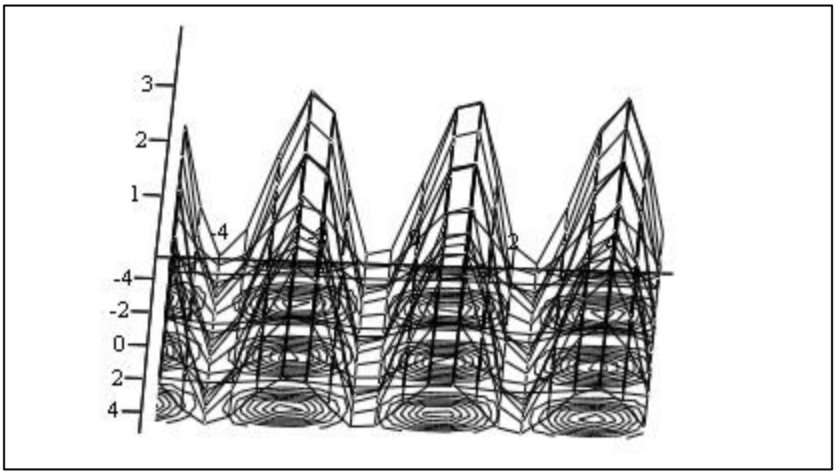
№	K(гаус.кр)	H(сер.кр)	k_1	k_2
1	-607040,03	-5268,80	57,29557	-10594,9
2	-13848890,98	15192,11	30833,37	-449,153
3	-2221664,09	-16798,48	65,99728	-33663
4	-27353787,90	-60152,33	226,9429	-120532
5	-4905568,99	70435,03	140904,9	-34,8148
6	-170098,84	10274,84	20557,96	-8,27411
7	-51072568,06	64281,30	128958,6	-396,038
8	-14363613,50	10035,05	20761,93	-691,825
9	-2110593,35	-16975,54	62,05229	-34013,1
10	-29228709,88	-69108,76	211,1464	-138429
11	-5777115,75	58194,45	116438,5	-49,6152
12	-738353,32	16417,02	32856,51	-22,4721

З таблиці 3 бачимо, що поточні точки є гіперболічними, а поверхня має сідлову форму. На цій поверхні відсутні точки мінімуму. Частинні похідні в поточних точках суттєво відрізняються від нуля.

За методом Хука-Дживса, маємо: $z(-0,8125;-0,75)=0,1377870962$ - найменше значення функції. Але якщо обчислити значення функції в достатньо близькій точці $z(-0,71;-0,68)=0,10875$, то отримане значення функції вже буде меншим за визначений мінімум. Отже метод Хука-Дживса для даної функції в околі заданої точки не дає вірного результату.

Приклад 4

$$z(x, y) = (2 + \sin 2x)(2 + \sin 2y)$$



1,1

Рис.11

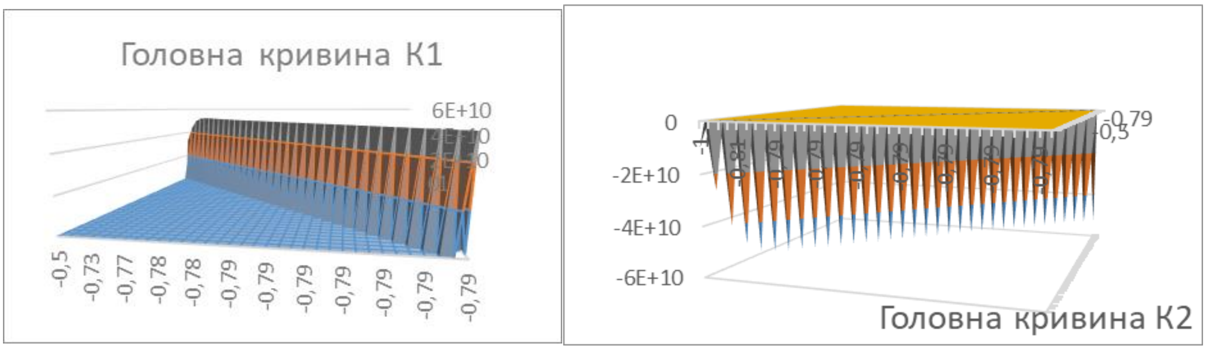


Рис.12

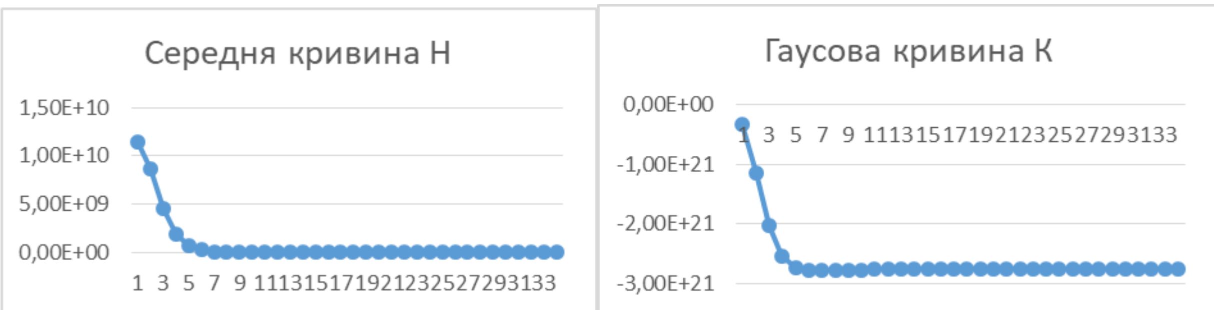


Рис.13

Ця функція досягає свого локального найменшого значення $z(-0,7848, -0,7858) = 1$. $K < 0, k_1 > 0, k_2 < 0$, $H > 0$. При цьому є точки на поверхні, в яких $k_1 = k_2$. Отже поточні точки мають гіперболічний характер, а функція має сідлову форму.

Методом Хука-Дживса отримано такі результати: найменше значення функції: - $z(-0,8125;-0,8125) = 1,012093$. Різниця у розв'язках є невеликою.

Висновки

З розглянутих прикладів зрозуміло, що обчислені значення кривин поверхонь в поточних точках дозволяють більш точно, ніж у стандартних методах оптимізації, визначити наявність екстремуму функції в певній точці. Так у прикладах 3 та 4 маємо сідлову форму поверхні, але у прикладі 3 не існує мінімальної точки, а у прикладі 4 така точка знайдена (локальний мінімум).

В роботі проведена геометрична інтерпретація квадратичної апроксимації поверхні стичним параболоїдом, і перехід від параболоїду до індикатриси Дюпена. Це зроблено у зв'язку з тим, що в загальному випадку задача відшукання головних кривин поверхні та головних напрямків у кожній поточній точці на сьогоднішній день в загальному випадку не вирішена. Як правило, її вирішують у внутрішніх координатах поверхні і перехід до зовнішньої системи координат є досить складним, а інколи і неможливим. Маючи рівняння індикатриси Дюпена - кривої другого порядку, можна відомими методами знайти головні напрямки відносно цієї кривої і далі рухатись по поверхні в одному із цих напрямків, адже, як відомо, в цих напрямках головні кривини поверхні набувають своїх екстремальних значень.

В роботі на прикладах показано, що обчислення кривин поверхні в поточних точках суттєво допомагає розв'язанню оптимізаційної задачі, адже, по суті, аналіз кривин в поточних точках методу оптимізації є достатньою умовою екстремуму і також допомагає більш чітко проаналізувати рельєф поверхні.

Список використаних джерел:

1. *Жалдак М.І.* Основи теорії і методів оптимізації/ *М.І. Жалдак.* - Черкаси: Брама-Україна, 2005. - 608 с.
2. *Виноградов И.М.* Математическая энциклопедия, Том 2/ *И.М. Виноградов.* – Москва: Советская энциклопедия, 1979. – 552 с.
3. *Кованцов Н.И.* Дифференциальная геометрия, топология, тензорный анализ: Сб. задач/ *Н.И. Кованцов.* - Киев: Выща шк., 1989.-398с.
4. *Веселов А.П.* Лекции по аналитической геометрии. Учеб. пособие/ *А.П. Веселов.* – Москва: Изд-во Центра прикладных исследований при механико-математическом ф-те МГУ, 2002. – 160 с.
5. *Погорелов А.В.* Геометрия/ А.В. *Погорелов.* - Москва: Наука, 1984.– 288 с.
6. *Нефьодов Ю.М.* Методи оптимізації в прикладах і задачах / *Ю.М. Нефьодов, Т.Ю Балицька* - *Київ: Кондор, 2011.-324с.*
7. *Зайченко Ю.П.* Исследование операций: Учебник.-6 изд., перераб. и доп./ *Ю.П. Зайченко.* - Киев: Издательский Дом «Слово», 2003. – 688 с.
8. *Шарый С.П.* Курс вычислительных методов/ *С.П. Шарый.* - Новосибирск: Институт вычислительных технологий СО РАН, 2012. – 315 с.
9. *Буката Л.Н.* Чисельні методи та моделювання на ЕОМ: навчальний посібник. – Ч. 1. – Модуль 2 / *Л.Н. Буката.* - Одеса: ОНАЗ ім. О. С. Попова, 2013. –84 с.
10. *Рашевский П.К.* Курс дифференциальной геометрии. Гос. издат. технико-теор. литературы, М.,Л. -1950 -428с.
11. http://study.sfu-kras.ru/file.php/168/10.html
12. https://is.gd/J86RXM

Додаток 1
Текст програми методу Хука-Дживса

```
PascalABC
type
  vector = array of real;
  Tfunction = function(x: vector): real;

var
  X, Delta: vector;
  Epsilon, Alpha, Lambda: real;
  HookResult, test: vector;
  f: Tfunction;

// Досліджувані функції
function f0(x: vector): real;
begin
  Result := 4 * Sqr(x[0] - 5) + Sqr(x[1] - 6);
end;

function f1(x: vector): real;
begin
  Result := -4 * x[0] - 6 * x[1] + 2 * Sqr(x[0]) + 2 * x[0] * x[1] + 2 *
Sqr(x[1]);
end;

// Зменшує кожен елемент Delta > Epsilon у Alpha разів.
// Істинна, якщо кожен елемент Delta став <= Epsilon
function CuttingDelta(var Delta:vector; Alpha, Epsilon: real):boolean;
var
  LittleAll: boolean;
begin
  LittleAll := true;
  write('  Старе значення Delta: ', Delta);
  for var i:=0 to Length(Delta)-1 do
    begin
      LittleAll := LittleAll AND (Delta[i] <= Epsilon);
```

```
      if Delta[i] > Epsilon then
        Delta[i] := Delta[i] / Alpha;
    end;
  writeln(', нове значення Delta: ', Delta);
  writeln;
  Result:=LittleAll;
end;


// Пошук локального мінімуму функції методом Хука-Дживса
function Hook(x, Delta: vector; Epsilon, Alpha, Lambda: real; f:
Tfunction):vector;
var
  x_old, x_tmp: vector;
  N: integer;
  i: integer;
  Etap: integer;// номер ітерації
  Cont:boolean;
begin
  // Визначаємо кількість аргументів
  N := Length(x);

  // Запам'ятовуємо вектор аргументів
  x_old := new real[N];
  x_old := Copy(x);

  // Створюємо допоміжний вектор аргументів
  x_tmp := new real[N];

  Etap:=1;
  repeat
    repeat
      writeln('Етап ', Etap);
      writeln('  Базис: ',x);
      Inc(Etap);
      x_old := Copy(x);
      for i := 0 to N - 1 do
```

```
        begin
          // Робимо крок вперед
          writeln('  Координата x[',i,']:');


          x_tmp := Copy(x);
          x_tmp[i] := x[i] + Delta[i];
          write('    базові: ', x, ', f(базові) = ', f(x), ', пробні: ',
x_tmp, ', f(пробні) = ', f(x_tmp), ' - крок ');
          if f(x_tmp) < f(x) then
            // Крок вперед виявився вдалим
            begin
              writeln('вдалий');
              x[i] := x[i] + Delta[i];
            end
          else
            // Крок вперед виявився невдалим, спробуємо крок назад
            begin
              writeln('невдалий');
              x_tmp := Copy(x);
              x_tmp[i] := x[i] - Delta[i];
              write('    базові: ', x, ', f(базові) = ', f(x), ', пробні: ',
x_tmp, ', f(пробні) = ', f(x_tmp), ' - крок ');
              if f(x_tmp) < f(x) then
                // Крок назад виявився вдалим
                begin
                  writeln('вдалий');
                  x[i] := x[i] - Delta[i];
                end
              else
                writeln('невдалий');
            end;
        end;
        writeln;
        writeln('  Отримали новий базис ',x);
        Cont:= f(x) < f(x_old);
        if Cont then
```

```
          begin
            write('  Від старого базису ', x_old, ' та нового ', x);
            x_tmp := Copy(x);
            for i:=0 to N-1 do
              x[i] := x[i] + Lambda*(x[i]-x_old[i]);
            writeln(' намагаємось прискорено перейти до базису ',x);
            // Перевіряємо, чи вдалим був перехід
            Cont := f(x) < f(x_tmp);
            if NOT Cont then
              begin
                writeln('  Оскільки f(', x,') = ', f(x), ' >= f(', x_tmp, ')
= ',f(x_tmp), ', прискорений перехід виявився невдалим.');
                x := Copy(x_tmp);
             end
            else
              begin
                writeln('  Оскільки f(', x,') = ', f(x), ' < f(', x_tmp, ') =
',f(x_tmp), ', прискорений перехід виявився вдалим.');
              end;
            writeln('  Новий базис: x = ', x);
            writeln;
            writeln;
          end;
    until NOT Cont;
    writeln('Зменшення кроку:');
  until CuttingDelta(Delta, Alpha, Epsilon); // Зменшуємо величину Delta
  Result := Copy(x);
end;

begin
  // Початкові дані
  x := new real[](8, 9);
  Delta := new real[](1, 2);
  Epsilon := 0.1;
  Alpha := 2;
  Lambda := 1;
```

```
  f:=f0;

  // Пошук мінімуму функції
  HookResult := Copy(Hook(X, Delta, Epsilon, Alpha, Lambda, f));
  writeln;
  writeln('ВІДПОВІДЬ: ', HookResult);
  writeln;

  writeln('Перевірка:');
  writeln('f(', HookResult, ') = ', f(HookResult));
  for var i:=0 to Length(HookResult)-1 do
    begin
      test := Copy(HookResult);
      test[i] := test[i] + Epsilon;
      write('    f(', test, ') = ', f(test));
      test := Copy(HookResult);
      test[i] := test[i] - Epsilon;
      writeln('; f(', test, ') = ', f(test));
    end;
end.
```

Додаток 2

Приклади знаходження мінімуму функцій методом Хука-Дживса в PascalABC

Приклад 1: $z(x,y)=3x^2+xy+2y^2-x-4y$

```
Досліджувана функція:
             2            2
f(x ,x ) = 3x  +x x  + 2x - x  -4x
   0  1      0   0 1     1   0    1

Пошук мінімуму методом Хука-Дживса:
Етап 1
  Базис: [0,0.75]
  Координата x[0]:
    базові: [0,0.75], f(базові) = -1.875, пробні: [1,0.75], f(пробні) = 0.875
- крок невдалий
    базові: [0,0.75], f(базові) = -1.875, пробні: [-1,0.75], f(пробні) =
1.375 - крок невдалий
  Координата x[1]:
    базові: [0,0.75], f(базові) = -1.875, пробні: [0,2.75], f(пробні) = 4.125
- крок невдалий
    базові: [0,0.75], f(базові) = -1.875, пробні: [0,-1.25], f(пробні) =
8.125 - крок невдалий
```

```
  Отримали новий базис [0,0.75]
Зменшення кроку:
  Старе значення Delta: [1,2], нове значення Delta: [0.5,1]

Етап 2
  Базис: [0,0.75]
  Координата x[0]:
    базові: [0,0.75], f(базові) = -1.875, пробні: [0.5,0.75], f(пробні) = -
1.25 - крок невдалий
    базові: [0,0.75], f(базові) = -1.875, пробні: [-0.5,0.75], f(пробні) = -1
- крок невдалий
  Координата x[1]:
    базові: [0,0.75], f(базові) = -1.875, пробні: [0,1.75], f(пробні) = -
0.875 - крок невдалий
    базові: [0,0.75], f(базові) = -1.875, пробні: [0,-0.25], f(пробні) =
1.125 - крок невдалий

  Отримали новий базис [0,0.75]
Зменшення кроку:
  Старе значення Delta: [0.5,1], нове значення Delta: [0.25,0.5]

Етап 3
  Базис: [0,0.75]
  Координата x[0]:
    базові: [0,0.75], f(базові) = -1.875, пробні: [0.25,0.75], f(пробні) = -
1.75 - крок невдалий
    базові: [0,0.75], f(базові) = -1.875, пробні: [-0.25,0.75], f(пробні) = -
1.625 - крок невдалий
  Координата x[1]:
    базові: [0,0.75], f(базові) = -1.875, пробні: [0,1.25], f(пробні) = -
1.875 - крок невдалий
    базові: [0,0.75], f(базові) = -1.875, пробні: [0,0.25], f(пробні) = -
0.875 - крок невдалий

  Отримали новий базис [0,0.75]
Зменшення кроку:
  Старе значення Delta: [0.25,0.5], нове значення Delta: [0.125,0.25]

Етап 4
  Базис: [0,0.75]
  Координата x[0]:
    базові: [0,0.75], f(базові) = -1.875, пробні: [0.125,0.75], f(пробні) = -
1.859375 - крок невдалий
    базові: [0,0.75], f(базові) = -1.875, пробні: [-0.125,0.75], f(пробні) =
-1.796875 - крок невдалий
  Координата x[1]:
    базові: [0,0.75], f(базові) = -1.875, пробні: [0,1], f(пробні) = -2 -
крок вдалий

  Отримали новий базис [0,1]
  Від старого базису [0,0.75] та нового [0,1] намагаємось прискорено перейти
до базису [0,1.25]
  Оскільки f([0,1.25]) = -1.875 >= f([0,1]) = -2, прискорений перехід
виявився невдалим.
  Новий базис: x = [0,1]


Зменшення кроку:
  Старе значення Delta: [0.125,0.25], нове значення Delta: [0.0625,0.125]

Етап 5
  Базис: [0,1]
  Координата x[0]:
```

```
    базові: [0,1], f(базові) = -2, пробні: [0.0625,1], f(пробні) = -
1.98828125 - крок невдалий
    базові: [0,1], f(базові) = -2, пробні: [-0.0625,1], f(пробні) = -
1.98828125 - крок невдалий
  Координата x[1]:
    базові: [0,1], f(базові) = -2, пробні: [0,1.125], f(пробні) = -1.96875 -
крок невдалий
    базові: [0,1], f(базові) = -2, пробні: [0,0.875], f(пробні) = -1.96875 -
крок невдалий

  Отримали новий базис [0,1]
Зменшення кроку:
  Старе значення Delta: [0.0625,0.125], нове значення Delta: [0.0625,0.0625]

Етап 6
  Базис: [0,1]
  Координата x[0]:
    базові: [0,1], f(базові) = -2, пробні: [0.0625,1], f(пробні) = -
1.98828125 - крок невдалий
    базові: [0,1], f(базові) = -2, пробні: [-0.0625,1], f(пробні) = -
1.98828125 - крок невдалий
  Координата x[1]:
    базові: [0,1], f(базові) = -2, пробні: [0,1.0625], f(пробні) = -1.9921875
- крок невдалий
    базові: [0,1], f(базові) = -2, пробні: [0,0.9375], f(пробні) = -1.9921875
- крок невдалий

  Отримали новий базис [0,1]
Зменшення кроку:
  Старе значення Delta: [0.0625,0.0625], нове значення Delta: [0.0625,0.0625]
```

ВІДПОВІДЬ: [0,1]

```
Перевірка:
f([0,1]) = -2
    f([0.1,1]) = -1.97; f([-0.1,1]) = -1.97
    f([0,1.1]) = -1.98; f([0,0.9]) = -1.98
```

Приклад 2. $z(x,y) = x^3 + 2y^2 - 3x - 4y$

Досліджувана функція:

```
            3    2
f(x ,x ) = x + 2x - 3x - 4x
   0  1     0    1    0    1
```

```
Пошук мінімуму методом Хука-Дживса:
Етап 1
  Базис: [-0.7,-1]
  Координата x[0]:
    базові: [-0.7,-1], f(базові) = 7.757, пробні: [0.3,-1], f(пробні) = 5.127
- крок вдалий
  Координата x[1]:
    базові: [0.3,-1], f(базові) = 5.127, пробні: [0.3,1], f(пробні) = -2.873
- крок вдалий

  Отримали новий базис [0.3,1]
  Від старого базису [-0.7,-1] та нового [0.3,1] намагаємось прискорено
перейти до базису [1.3,3]
```

Оскільки f([1.3,3]) = 4.297 >= f([0.3,1]) = -2.873, прискорений перехід виявився невдалим.
Новий базис: x = [0.3,1]

Зменшення кроку:
Старе значення Delta: [1,2], нове значення Delta: [0.5,1]

Етап 2
Базис: [0.3,1]
Координата x[0]:
базові: [0.3,1], f(базові) = -2.873, пробні: [0.8,1], f(пробні) = -3.888 - крок вдалий
Координата x[1]:
базові: [0.8,1], f(базові) = -3.888, пробні: [0.8,2], f(пробні) = -1.888 - крок невдалий
базові: [0.8,1], f(базові) = -3.888, пробні: [0.8,0], f(пробні) = -1.888 - крок невдалий

Отримали новий базис [0.8,1]
Від старого базису [0.3,1] та нового [0.8,1] намагаємось прискорено перейти до базису [1.3,1]
Оскільки f([1.3,1]) = -3.703 >= f([0.8,1]) = -3.888, прискорений перехід виявився невдалим.
Новий базис: x = [0.8,1]

Зменшення кроку:
Старе значення Delta: [0.5,1], нове значення Delta: [0.25,0.5]

Етап 3
Базис: [0.8,1]
Координата x[0]:
базові: [0.8,1], f(базові) = -3.888, пробні: [1.05,1], f(пробні) = -3.992375 - крок вдалий
Координата x[1]:
базові: [1.05,1], f(базові) = -3.992375, пробні: [1.05,1.5], f(пробні) = -3.492375 - крок невдалий
базові: [1.05,1], f(базові) = -3.992375, пробні: [1.05,0.5], f(пробні) = -3.492375 - крок невдалий

Отримали новий базис [1.05,1]
Від старого базису [0.8,1] та нового [1.05,1] намагаємось прискорено перейти до базису [1.3,1]
Оскільки f([1.3,1]) = -3.703 >= f([1.05,1]) = -3.992375, прискорений перехід виявився невдалим.
Новий базис: x = [1.05,1]

Зменшення кроку:
Старе значення Delta: [0.25,0.5], нове значення Delta: [0.125,0.25]

Етап 4
Базис: [1.05,1]
Координата x[0]:
базові: [1.05,1], f(базові) = -3.992375, пробні: [1.175,1], f(пробні) = -3.902765625 - крок невдалий
базові: [1.05,1], f(базові) = -3.992375, пробні: [0.925,1], f(пробні) = -3.983546875 - крок невдалий
Координата x[1]:
базові: [1.05,1], f(базові) = -3.992375, пробні: [1.05,1.25], f(пробні) = -3.867375 - крок невдалий

```
    базові: [1.05,1], f(базові) = -3.992375, пробні: [1.05,0.75], f(пробні) =
-3.867375 - крок невдалий

  Отримали новий базис [1.05,1]
Зменшення кроку:
  Старе значення Delta: [0.125,0.25], нове значення Delta: [0.0625,0.125]

Етап 5
  Базис: [1.05,1]
  Координата x[0]:
    базові: [1.05,1], f(базові) = -3.992375, пробні: [1.1125,1], f(пробні) =
-3.960607421875 - крок невдалий
    базові: [1.05,1], f(базові) = -3.992375, пробні: [0.9875,1], f(пробні) =
-3.999533203125 - крок вдалий
  Координата x[1]:
    базові: [0.9875,1], f(базові) = -3.999533203125, пробні: [0.9875,1.125],
f(пробні) = -3.968283203125 - крок невдалий
    базові: [0.9875,1], f(базові) = -3.999533203125, пробні: [0.9875,0.875],
f(пробні) = -3.968283203125 - крок невдалий

  Отримали новий базис [0.9875,1]
  Від старого базису [1.05,1] та нового [0.9875,1] намагаємось прискорено
перейти до базису [0.925,1]
  Оскільки f([0.925,1]) = -3.983546875 >= f([0.9875,1]) = -3.999533203125,
прискорений перехід виявився невдалим.
  Новий базис: x = [0.9875,1]


Зменшення кроку:
  Старе значення Delta: [0.0625,0.125], нове значення Delta: [0.0625,0.0625]

Етап 6
  Базис: [0.9875,1]
  Координата x[0]:
    базові: [0.9875,1], f(базові) = -3.999533203125, пробні: [1.05,1],
f(пробні) = -3.992375 - крок невдалий
    базові: [0.9875,1], f(базові) = -3.999533203125, пробні: [0.925,1],
f(пробні) = -3.983546875 - крок невдалий
  Координата x[1]:
    базові: [0.9875,1], f(базові) = -3.999533203125, пробні: [0.9875,1.0625],
f(пробні) = -3.991720703125 - крок невдалий
    базові: [0.9875,1], f(базові) = -3.999533203125, пробні: [0.9875,0.9375],
f(пробні) = -3.991720703125 - крок невдалий

  Отримали новий базис [0.9875,1]
Зменшення кроку:
  Старе значення Delta: [0.0625,0.0625], нове значення Delta: [0.0625,0.0625]
```

ВІДПОВІДЬ: [0.9875,1]

```
Перевірка:
f([0.9875,1]) = -3.999533203125
    f([1.0875,1]) = -3.976361328125; f([0.8875,1]) = -3.963455078125
    f([0.9875,1.1]) = -3.979533203125; f([0.9875,0.9]) = -3.979533203125
```

Приклад 3. $z(x,y) = 10(y - \sin x)^2 + 0,2x^2$

Досліджувана функція:

$$f(x_0, x_1) = 10(x_1 - \sin(x_0))^2 + 0.2x_0^2$$

```
Пошук мінімуму методом Хука-Дживса:
Етап 1
  Базис: [-0.5,-1]
  Координата x[0]:
    базові: [-0.5,-1], f(базові) = 2.75997769857524, пробні: [0.5,-1],
f(пробні) = 21.9369992427434 - крок невдалий
    базові: [-0.5,-1], f(базові) = 2.75997769857524, пробні: [-1.5,-1],
f(пробні) = 0.450062750921139 - крок вдалий
  Координата x[1]:
    базові: [-1.5,-1], f(базові) = 0.450062750921139, пробні: [-1.5,1],
f(пробні) = 40.3498622150833 - крок невдалий
    базові: [-1.5,-1], f(базові) = 0.450062750921139, пробні: [-1.5,-3],
f(пробні) = 40.550263286759 - крок невдалий

  Отримали новий базис [-1.5,-1]
  Від старого базису [-0.5,-1] та нового [-1.5,-1] намагаємось прискорено
перейти до базису [-2.5,-1]
  Оскільки f([-2.5,-1]) = 2.86224619060474 >= f([-1.5,-1]) =
0.450062750921139, прискорений перехід виявився невдалим.
  Новий базис: x = [-1.5,-1]


Зменшення кроку:
  Старе значення Delta: [1,2], нове значення Delta: [0.5,1]

Етап 2
  Базис: [-1.5,-1]
  Координата x[0]:
    базові: [-1.5,-1], f(базові) = 0.450062750921139, пробні: [-1,-1],
f(пробні) = 0.451314486577782 - крок невдалий
    базові: [-1.5,-1], f(базові) = 0.450062750921139, пробні: [-2,-1],
f(пробні) = 0.882269567804426 - крок невдалий
  Координата x[1]:
    базові: [-1.5,-1], f(базові) = 0.450062750921139, пробні: [-1.5,0],
f(пробні) = 10.3999624830022 - крок невдалий
    базові: [-1.5,-1], f(базові) = 0.450062750921139, пробні: [-1.5,-2],
f(пробні) = 10.50016301884 - крок невдалий

  Отримали новий базис [-1.5,-1]
Зменшення кроку:
  Старе значення Delta: [0.5,1], нове значення Delta: [0.25,0.5]

Етап 3
  Базис: [-1.5,-1]
  Координата x[0]:
    базові: [-1.5,-1], f(базові) = 0.450062750921139, пробні: [-1.25,-1],
f(пробні) = 0.338525690622944 - крок вдалий
  Координата x[1]:
    базові: [-1.25,-1], f(базові) = 0.338525690622944, пробні: [-1.25,-0.5],
f(пробні) = 2.32837188417881 - крок невдалий
    базові: [-1.25,-1], f(базові) = 0.338525690622944, пробні: [-1.25,-1.5],
f(пробні) = 3.34867949706708 - крок невдалий

  Отримали новий базис [-1.25,-1]
  Від старого базису [-1.5,-1] та нового [-1.25,-1] намагаємось прискорено
перейти до базису [-1,-1]
  Оскільки f([-1,-1]) = 0.451314486577782 >= f([-1.25,-1]) =
0.338525690622944, прискорений перехід виявився невдалим.
  Новий базис: x = [-1.25,-1]


Зменшення кроку:
```

```
  Старе значення Delta: [0.25,0.5], нове значення Delta: [0.125,0.25]

Етап 4
  Базис: [-1.25,-1]
  Координата x[0]:
    базові: [-1.25,-1], f(базові) = 0.338525690622944, пробні: [-1.125,-1], 
f(пробні) = 0.348641231631792 - крок невдалий
    базові: [-1.25,-1], f(базові) = 0.338525690622944, пробні: [-1.375,-1], 
f(пробні) = 0.381775752699204 - крок невдалий
  Координата x[1]:
    базові: [-1.25,-1], f(базові) = 0.338525690622944, пробні: [-1.25,-0.75], 
f(пробні) = 0.708448787400875 - крок невдалий
    базові: [-1.25,-1], f(базові) = 0.338525690622944, пробні: [-1.25,-1.25], 
f(пробні) = 1.21860259384501 - крок невдалий

  Отримали новий базис [-1.25,-1]
Зменшення кроку:
  Старе значення Delta: [0.125,0.25], нове значення Delta: [0.0625,0.125]

Етап 5
  Базис: [-1.25,-1]
  Координата x[0]:
    базові: [-1.25,-1], f(базові) = 0.338525690622944, пробні: [-1.1875,-1], 
f(пробні) = 0.334685259586105 - крок вдалий
  Координата x[1]:
    базові: [-1.1875,-1], f(базові) = 0.334685259586105, пробні: [-1.1875,-
0.875], f(пробні) = 0.309527553048274 - крок вдалий

  Отримали новий базис [-1.1875,-0.875]
  Від старого базису [-1.25,-1] та нового [-1.1875,-0.875] намагаємось 
прискорено перейти до базису [-1.125,-0.75]
  Оскільки f([-1.125,-0.75]) = 0.484979202127268 >= f([-1.1875,-0.875]) = 
0.309527553048274, прискорений перехід виявився невдалим.
  Новий базис: x = [-1.1875,-0.875]


Зменшення кроку:
  Старе значення Delta: [0.0625,0.125], нове значення Delta: [0.0625,0.0625]

Етап 6
  Базис: [-1.1875,-0.875]
  Координата x[0]:
    базові: [-1.1875,-0.875], f(базові) = 0.309527553048274, пробні: [-
1.125,-0.875], f(пробні) = 0.26056021687953 - крок вдалий
  Координата x[1]:
    базові: [-1.125,-0.875], f(базові) = 0.26056021687953, пробні: [-1.125,-
0.8125], f(пробні) = 0.333707209503399 - крок невдалий
    базові: [-1.125,-0.875], f(базові) = 0.26056021687953, пробні: [-1.125,-
0.9375], f(пробні) = 0.265538224255661 - крок невдалий

  Отримали новий базис [-1.125,-0.875]
  Від старого базису [-1.1875,-0.875] та нового [-1.125,-0.875] намагаємось 
прискорено перейти до базису [-1.0625,-0.875]
  Оскільки f([-1.0625,-0.875]) = 0.225801558097781 < f([-1.125,-0.875]) = 
0.26056021687953, прискорений перехід виявився вдалим.
  Новий базис: x = [-1.0625,-0.875]


Етап 7
  Базис: [-1.0625,-0.875]
  Координата x[0]:
    базові: [-1.0625,-0.875], f(базові) = 0.225801558097781, пробні: [-1,-
0.875], f(пробні) = 0.211241948597523 - крок вдалий
```

```
  Координата x[1]:
    базові: [-1,-0.875], f(базові) = 0.211241948597523, пробні: [-1,-0.8125],
f(пробні) = 0.208393179607394 - крок вдалий

  Отримали новий базис [-1,-0.8125]
  Від старого базису [-1.0625,-0.875] та нового [-1,-0.8125] намагаємось
прискорено перейти до базису [-0.9375,-0.75]
  Оскільки f([-0.9375,-0.75]) = 0.207232157037476 < f([-1,-0.8125]) =
0.208393179607394, прискорений перехід виявився вдалим.
  Новий базис: x = [-0.9375,-0.75]


Етап 8
  Базис: [-0.9375,-0.75]
  Координата x[0]:
    базові: [-0.9375,-0.75], f(базові) = 0.207232157037476, пробні: [-0.875,-
0.75], f(пробні) = 0.156202744707055 - крок вдалий
  Координата x[1]:
    базові: [-0.875,-0.75], f(базові) = 0.156202744707055, пробні: [-0.875,-
0.6875], f(пробні) = 0.217194622502089 - крок невдалий
    базові: [-0.875,-0.75], f(базові) = 0.156202744707055, пробні: [-0.875,-
0.8125], f(пробні) = 0.173335866912021 - крок невдалий

  Отримали новий базис [-0.875,-0.75]
  Від старого базису [-0.9375,-0.75] та нового [-0.875,-0.75] намагаємось
прискорено перейти до базису [-0.8125,-0.75]
  Оскільки f([-0.8125,-0.75]) = 0.137787096223993 < f([-0.875,-0.75]) =
0.156202744707055, прискорений перехід виявився вдалим.
  Новий базис: x = [-0.8125,-0.75]


Етап 9
  Базис: [-0.8125,-0.75]
  Координата x[0]:
    базові: [-0.8125,-0.75], f(базові) = 0.137787096223993, пробні: [-0.75,-
0.75], f(пробні) = 0.159232591311473 - крок невдалий
    базові: [-0.8125,-0.75], f(базові) = 0.137787096223993, пробні: [-0.875,-
0.75], f(пробні) = 0.156202744707055 - крок невдалий
  Координата x[1]:
    базові: [-0.8125,-0.75], f(базові) = 0.137787096223993, пробні: [-
0.8125,-0.6875], f(пробні) = 0.146860415299884 - крок невдалий
    базові: [-0.8125,-0.75], f(базові) = 0.137787096223993, пробні: [-
0.8125,-0.8125], f(пробні) = 0.206838777148103 - крок невдалий

  Отримали новий базис [-0.8125,-0.75]
Зменшення кроку:
  Старе значення Delta: [0.0625,0.0625], нове значення Delta: [0.0625,0.0625]
```

ВІДПОВІДЬ: [-0.8125,-0.75]

```
Перевірка:
f([-0.8125,-0.75]) = 0.137787096223993
    f([-0.7125,-0.75]) = 0.194214929243653; f([-0.9125,-0.75]) =
0.183370483618053
    f([-0.8125,-0.65]) = 0.189804406745418; f([-0.8125,-0.85]) =
0.285769785702568
```

Приклад 4. $z(x,y)=(2+\sin 2x)(2+\sin 2y)$

Досліджувана функція:

$$f(x_0,x_1) = (2+\sin(2x_0))(2+\sin(2x_1))$$

```
Пошук мінімуму методом Хука-Дживса:
Етап 1
  Базис: [-0.5,-1]
  Координата x[0]:
    базові: [-0.5,-1], f(базові) = 1.26361057796714, пробні: [0.5,-1],
f(пробні) = 3.09919971473014 - крок невдалий
    базові: [-0.5,-1], f(базові) = 1.26361057796714, пробні: [-1.5,-1],
f(пробні) = 2.02748519043136 - крок невдалий
  Координата x[1]:
    базові: [-0.5,-1], f(базові) = 1.26361057796714, пробні: [-0.5,1],
f(пробні) = 3.37050548280128 - крок невдалий
    базові: [-0.5,-1], f(базові) = 1.26361057796714, пробні: [-0.5,-3],
f(пробні) = 2.64076899234202 - крок невдалий

  Отримали новий базис [-0.5,-1]
Зменшення кроку:
  Старе значення Delta: [1,2], нове значення Delta: [0.5,1]

Етап 2
  Базис: [-0.5,-1]
  Координата x[0]:
    базові: [-0.5,-1], f(базові) = 1.26361057796714, пробні: [0,-1],
f(пробні) = 2.18140514634864 - крок невдалий
    базові: [-0.5,-1], f(базові) = 1.26361057796714, пробні: [-1,-1],
f(пробні) = 1.18963210312908 - крок вдалий
  Координата x[1]:
    базові: [-1,-1], f(базові) = 1.18963210312908, пробні: [-1,0], f(пробні)
= 2.18140514634864 - крок невдалий
    базові: [-1,-1], f(базові) = 1.18963210312908, пробні: [-1,-2], f(пробні)
= 3.00685157536574 - крок невдалий

  Отримали новий базис [-1,-1]
  Від старого базису [-0.5,-1] та нового [-1,-1] намагаємось прискорено
перейти до базису [-1.5,-1]
  Оскільки f([-1.5,-1]) = 2.02748519043136 >= f([-1,-1]) = 1.18963210312908,
прискорений перехід виявився невдалим.
  Новий базис: x = [-1,-1]


Зменшення кроку:
  Старе значення Delta: [0.5,1], нове значення Delta: [0.25,0.5]

Етап 3
  Базис: [-1,-1]
  Координата x[0]:
    базові: [-1,-1], f(базові) = 1.18963210312908, пробні: [-0.75,-1],
f(пробні) = 1.09343479773111 - крок вдалий
  Координата x[1]:
    базові: [-0.75,-1], f(базові) = 1.09343479773111, пробні: [-0.75,-0.5],
f(пробні) = 1.16143114589475 - крок невдалий
    базові: [-0.75,-1], f(базові) = 1.09343479773111, пробні: [-0.75,-1.5],
f(пробні) = 1.8635365112214 - крок невдалий

  Отримали новий базис [-0.75,-1]
  Від старого базису [-1,-1] та нового [-0.75,-1] намагаємось прискорено
перейти до базису [-0.5,-1]
  Оскільки f([-0.5,-1]) = 1.26361057796714 >= f([-0.75,-1]) =
1.09343479773111, прискорений перехід виявився невдалим.
  Новий базис: x = [-0.75,-1]
```

```
Зменшення кроку:
  Старе значення Delta: [0.25,0.5], нове значення Delta: [0.125,0.25]

Етап 4
  Базис: [-0.75,-1]
  Координата x[0]:
    базові: [-0.75,-1], f(базові) = 1.09343479773111, пробні: [-0.625,-1],
f(пробні) = 1.14634518011465 - крок невдалий
    базові: [-0.75,-1], f(базові) = 1.09343479773111, пробні: [-0.875,-1],
f(пробні) = 1.10816914212587 - крок невдалий
  Координата x[1]:
    базові: [-0.75,-1], f(базові) = 1.09343479773111, пробні: [-0.75,-0.75],
f(пробні) = 1.005016301884 - крок вдалий

  Отримали новий базис [-0.75,-0.75]
  Від старого базису [-0.75,-1] та нового [-0.75,-0.75] намагаємось
прискорено перейти до базису [-0.75,-0.5]
  Оскільки f([-0.75,-0.5]) = 1.16143114589475 >= f([-0.75,-0.75]) =
1.005016301884, прискорений перехід виявився невдалим.
  Новий базис: x = [-0.75,-0.75]


Зменшення кроку:
  Старе значення Delta: [0.125,0.25], нове значення Delta: [0.0625,0.125]

Етап 5
  Базис: [-0.75,-0.75]
  Координата x[0]:
    базові: [-0.75,-0.75], f(базові) = 1.005016301884, пробні: [-0.6875,-
0.75], f(пробні) = 1.0216598195209 - крок невдалий
    базові: [-0.75,-0.75], f(базові) = 1.005016301884, пробні: [-0.8125,-
0.75], f(пробні) = 1.00397735186774 - крок вдалий
  Координата x[1]:
    базові: [-0.8125,-0.75], f(базові) = 1.00397735186774, пробні: [-0.8125,-
0.625], f(пробні) = 1.05255896432598 - крок невдалий
    базові: [-0.8125,-0.75], f(базові) = 1.00397735186774, пробні: [-0.8125,-
0.875], f(пробні) = 1.01750623177685 - крок невдалий

  Отримали новий базис [-0.8125,-0.75]
  Від старого базису [-0.75,-0.75] та нового [-0.8125,-0.75] намагаємось
прискорено перейти до базису [-0.875,-0.75]
  Оскільки f([-0.875,-0.75]) = 1.01855918193961 >= f([-0.8125,-0.75]) =
1.00397735186774, прискорений перехід виявився невдалим.
  Новий базис: x = [-0.8125,-0.75]


Зменшення кроку:
  Старе значення Delta: [0.0625,0.125], нове значення Delta: [0.0625,0.0625]

Етап 6
  Базис: [-0.8125,-0.75]
  Координата x[0]:
    базові: [-0.8125,-0.75], f(базові) = 1.00397735186774, пробні: [-0.75,-
0.75], f(пробні) = 1.005016301884 - крок невдалий
    базові: [-0.8125,-0.75], f(базові) = 1.00397735186774, пробні: [-0.875,-
0.75], f(пробні) = 1.01855918193961 - крок невдалий
  Координата x[1]:
    базові: [-0.8125,-0.75], f(базові) = 1.00397735186774, пробні: [-0.8125,-
0.6875], f(пробні) = 1.02060366402957 - крок невдалий
    базові: [-0.8125,-0.75], f(базові) = 1.00397735186774, пробні: [-0.8125,-
0.8125], f(пробні) = 1.00293947588095 - крок вдалий
```

Отримали новий базис [-0.8125,-0.8125]
Від старого базису [-0.8125,-0.75] та нового [-0.8125,-0.8125] намагаємось прискорено перейти до базису [-0.8125,-0.875]
Оскільки f([-0.8125,-0.875]) = 1.01750623177685 >= f([-0.8125,-0.8125]) = 1.00293947588095, прискорений перехід виявився невдалим.
Новий базис: x = [-0.8125,-0.8125]

Зменшення кроку:
Старе значення Delta: [0.0625,0.0625], нове значення Delta: [0.0625,0.0625]

ВІДПОВІДЬ: [-0.8125,-0.8125]

Перевірка:
f([-0.8125,-0.8125]) = 1.00293947588095
f([-0.7125,-0.8125]) = 1.0120937122099; f([-0.9125,-0.8125]) = 1.0336519978783
f([-0.8125,-0.7125]) = 1.0120937122099; f([-0.8125,-0.9125]) = 1.0336519978783

Приклад 5. $z(x,y) = x^4+2y^4+x^2y^2+2x+y$

Досліджувана функція:

$$f(x_0,x_1) = x_0^4+2x_1^4+x_0^2x_1^2+2x_0+x_1$$

Пошук мінімуму методом Хука-Дживса:
Етап 1
Базис: [1,1]
Координата x[0]:
базові: [1,1], f(базові) = 7, пробні: [2,1], f(пробні) = 27 - крок невдалий
базові: [1,1], f(базові) = 7, пробні: [0,1], f(пробні) = 3 - крок вдалий
Координата x[1]:
базові: [0,1], f(базові) = 3, пробні: [0,3], f(пробні) = 165 - крок невдалий
базові: [0,1], f(базові) = 3, пробні: [0,-1], f(пробні) = 1 - крок вдалий

Отримали новий базис [0,-1]
Від старого базису [1,1] та нового [0,-1] намагаємось прискорено перейти до базису [-1,-3]
Оскільки f([-1,-3]) = 167 >= f([0,-1]) = 1, прискорений перехід виявився невдалим.
Новий базис: x = [0,-1]

Зменшення кроку:
Старе значення Delta: [1,2], нове значення Delta: [0.5,1]

Етап 2
Базис: [0,-1]
Координата x[0]:
базові: [0,-1], f(базові) = 1, пробні: [0.5,-1], f(пробні) = 2.3125 - крок невдалий
базові: [0,-1], f(базові) = 1, пробні: [-0.5,-1], f(пробні) = 0.3125 - крок вдалий
Координата x[1]:

базові: [-0.5,-1], f(базові) = 0.3125, пробні: [-0.5,0], f(пробні) = -0.9375 - крок вдалий

Отримали новий базис [-0.5,0]
Від старого базису [0,-1] та нового [-0.5,0] намагаємось прискорено перейти до базису [-1,1]
Оскільки f([-1,1]) = 3 >= f([-0.5,0]) = -0.9375, прискорений перехід виявився невдалим.
Новий базис: x = [-0.5,0]

Зменшення кроку:
Старе значення Delta: [0.5,1], нове значення Delta: [0.25,0.5]

Етап 3
Базис: [-0.5,0]
Координата x[0]:
базові: [-0.5,0], f(базові) = -0.9375, пробні: [-0.25,0], f(пробні) = -0.49609375 - крок невдалий
базові: [-0.5,0], f(базові) = -0.9375, пробні: [-0.75,0], f(пробні) = -1.18359375 - крок вдалий
Координата x[1]:
базові: [-0.75,0], f(базові) = -1.18359375, пробні: [-0.75,0.5], f(пробні) = -0.41796875 - крок невдалий
базові: [-0.75,0], f(базові) = -1.18359375, пробні: [-0.75,-0.5], f(пробні) = -1.41796875 - крок вдалий

Отримали новий базис [-0.75,-0.5]
Від старого базису [-0.5,0] та нового [-0.75,-0.5] намагаємось прискорено перейти до базису [-1,-1]
Оскільки f([-1,-1]) = 1 >= f([-0.75,-0.5]) = -1.41796875, прискорений перехід виявився невдалим.
Новий базис: x = [-0.75,-0.5]

Зменшення кроку:
Старе значення Delta: [0.25,0.5], нове значення Delta: [0.125,0.25]

Етап 4
Базис: [-0.75,-0.5]
Координата x[0]:
базові: [-0.75,-0.5], f(базові) = -1.41796875, пробні: [-0.625,-0.5], f(пробні) = -1.374755859375 - крок невдалий
базові: [-0.75,-0.5], f(базові) = -1.41796875, пробні: [-0.875,-0.5], f(пробні) = -1.347412109375 - крок невдалий
Координата x[1]:
базові: [-0.75,-0.5], f(базові) = -1.41796875, пробні: [-0.75,-0.25], f(пробні) = -1.390625 - крок невдалий
базові: [-0.75,-0.5], f(базові) = -1.41796875, пробні: [-0.75,-0.75], f(пробні) = -0.984375 - крок невдалий

Отримали новий базис [-0.75,-0.5]
Зменшення кроку:
Старе значення Delta: [0.125,0.25], нове значення Delta: [0.0625,0.125]

Етап 5
Базис: [-0.75,-0.5]
Координата x[0]:
базові: [-0.75,-0.5], f(базові) = -1.41796875, пробні: [-0.6875,-0.5], f(пробні) = -1.40843200683594 - крок невдалий
базові: [-0.75,-0.5], f(базові) = -1.41796875, пробні: [-0.8125,-0.5], f(пробні) = -1.39915466308594 - крок невдалий
Координата x[1]:

базові: [-0.75,-0.5], f(базові) = -1.41796875, пробні: [-0.75,-0.375], f(пробні) = -1.43994140625 - крок вдалий

Отримали новий базис [-0.75,-0.375]
Від старого базису [-0.75,-0.5] та нового [-0.75,-0.375] намагаємось прискорено перейти до базису [-0.75,-0.25]
Оскільки f([-0.75,-0.25]) = -1.390625 >= f([-0.75,-0.375]) = -1.43994140625, прискорений перехід виявився невдалим.
Новий базис: x = [-0.75,-0.375]

Зменшення кроку:
Старе значення Delta: [0.0625,0.125], нове значення Delta: [0.0625,0.0625]

Етап 6
Базис: [-0.75,-0.375]
Координата x[0]:
базові: [-0.75,-0.375], f(базові) = -1.43994140625, пробні: [-0.6875,-0.375], f(пробні) = -1.42057800292969 - крок невдалий
базові: [-0.75,-0.375], f(базові) = -1.43994140625, пробні: [-0.8125,-0.375], f(пробні) = -1.43180847167969 - крок невдалий
Координата x[1]:
базові: [-0.75,-0.375], f(базові) = -1.43994140625, пробні: [-0.75,-0.3125], f(пробні) = -1.42208862304688 - крок невдалий
базові: [-0.75,-0.375], f(базові) = -1.43994140625, пробні: [-0.75,-0.4375], f(пробні) = -1.44015502929688 - крок вдалий

Отримали новий базис [-0.75,-0.4375]
Від старого базису [-0.75,-0.375] та нового [-0.75,-0.4375] намагаємось прискорено перейти до базису [-0.75,-0.5]
Оскільки f([-0.75,-0.5]) = -1.41796875 >= f([-0.75,-0.4375]) = -1.44015502929688, прискорений перехід виявився невдалим.
Новий базис: x = [-0.75,-0.4375]

Зменшення кроку:
Старе значення Delta: [0.0625,0.0625], нове значення Delta: [0.0625,0.0625]

ВІДПОВІДЬ: [-0.75,-0.4375]

Перевірка:
f([-0.75,-0.4375]) = -1.44015502929688
f([-0.65,-0.4375]) = -1.40485190429688; f([-0.85,-0.4375]) = -1.40393002929688
f([-0.75,-0.3375]) = -1.43107221679688; f([-0.75,-0.5375]) = -1.39165034179688