МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Методи нелінійного програмування

МЕТОДИЧНІ ВКАЗІВКИ
ДО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ
З КУРСУ «ОПТИМІЗАЦІЯ ТЕХНОЛОГІЧНИХ ПРОЦЕСІВ ТА СИСТЕМ КЕРУВАННЯ»

Київ 2007

УДК 519.6:681.3

Методи нелінійного програмування.
Методичні вказівки до виконання лабораторних робіт з курсу "Оптимізація технологічних процесів і систем керування" для студ. спец. "Автоматизоване управління технологічними процесами" напряму "Автоматизація та комп’ютерно-інтегровані технологічні комплекси" Уклад.: А.І. Жученко, Л. Р. Ладієва, О. В. Снігур. – К.: НТУУ «КПІ», 2007. - 99 с.

Укладачі:	А.І. Жученко, д.т.н., проф.
	Л. Р. Ладієва, к. т. н., доц.
	О.В. Снігур
Відповідальний редактор:	А.І. Жученко, д.т.н., проф.
Рецензент:	В.В. Миленький, к. т. н., доц.

Вступ

В даних методичних вказівках викладені питання практичного застосування методів нелінійного програмування (НП). Увага приділяється методам і алгоритмам НП, що використовуються при проектуванні і керуванні технологічними об'єктами. Розглядаються методи оптимізації орієнтовані на розв'язок з неперервними змінними, обмеженнями, що містять функції.

Вивчаються важливі типи методів нелінійного програмування, починаючи з методів мінімізації функції однієї змінної і закінчуючи методами, що застосовуються для розв'язку нелінійних задач умовної оптимізації.

Приведений порівняльний аналіз і програми методів пошуку оптимального розв'язку з використанням математичного пакету MathCAD 2001i.

МЕТОДИ НЕЛІНІЙНОГО ПРОГРАМУВАННЯ, ЯКІ НЕ ВИКОРИСТОВУЮТЬ ПОХІДНІ

При реалізації методів прямого пошуку потрібно тільки значення цільової функції. При розв'язку задач безумовної оптимізації методами нелінійного програмування градієнтні методи та методи, які використовують другі похідні, сходяться швидше, ніж методи прямого пошуку.

Але, застосовуючи на практиці методи, які використовують похідні, доводиться стикатися з двома головними перешкодами. По-перше, в задачах з досить великим числом змінних важко або не можливо отримати похідні у вигляді аналітичних функцій, необхідних для градієнтного алгоритму або для алгоритму, який використовує похідні другого порядку. Хоч обчислення аналітичних похідних можна замінити обчисленням похідних за допомогою різницевих схем, помилка яка при цьому виникає, особливо в околиці екстремуму, може обмежити застосування подібної апроксимації. По-друге, при використанні методів оптимізації, основаних на обчисленні перших і при необхідності – других похідних, потрібно порівняно з методами прямого пошуку багато часу на підготовку задачі для розв'язку.

Методи прямого пошуку не потребують регулярності і неперервності цільової функції і існування похідних; хоча повільніше реалізуються у випадку простих задач але з точки зору користувача, на практиці можуть опинитися більш задовільними, ніж градієнтні методи.

Лабораторна робота №1

Методи одновимірного пошуку

Методи виключення інтервалів

За допомогою методів виключення інтервалів можна реалізувати процедуру пошуку, яка дозволяє знайти точку оптимуму шляхом послідовного виключення частин вихідного обмеженого інтервалу.

Логічна структура пошуку за допомогою методів виключення інтервалів основана на простому порівнянні значень функції в деяких пробних точках. Крім того, при такому порівнянні в розрахунок приймається тільки відношення порядку на множині значень функції і не враховується різність між значеннями функції. Пошук завершується, коли підінтервал, що залишився зменшується до достатньо малих розмірів.

Застосування методів виключення інтервалів накладається єдина вимога на досліджувану функцію: вона повинна бути унімодальною. Отже, вказані методи можна використовувати для аналізу як неперервних, так і розривних функцій, а також у випадках, коли змінні приймають значення із дискретної множини.

Метод дихотомії

Метод дихотомії (МД) застосовується для пошуку мінімуму унімодальної функції однієї змінної $Y = F(x)$, що задана на проміжку $[A, B]$. Алгоритми методу реалізуються у вигляді послідовності кроків, на кожному з яких здійснюється звуження інтервалу, що містить точку мінімуму. На початку обчислень $A_0 = A$, $B_0 = B$. На s-му кроці визначають величини

$$A_{s+1} = A_s,\ B_{s+1} = R_s, \text{ якщо } F(L_s) \le F(R_s),$$
$$A_{s+1} = L_s,\ B_{s+1} = B_s, \text{ якщо } F(L_s) > F(R_s),$$

Ітерації продовжують доти, поки не буде виконуватись нерівність

$$B_s - A_s \le EPS$$

де EPS > 0 – задане число, яке визначає похибку розв'язку задачі.

За наближений розв'язок задачі приймають

$$X^* = (A_s + B_s)/2,\ Y = F(X^*).$$

При розв'язуванні задачі максимізації функції $F(x)$ необхідно замінити її на функцію $-F(x)$.

Метод ділення інтервалу на чотири частини.

Цей метод інколи називають п'ятиточковим пошуком на рівних інтервалах, оскільки його реалізація основана на виборі п'яти пробних точок, рівномірно розподілених у інтервалі пошуку. Нижче наведений опис основних кроків пошукової процедури, орієнтованої на знаходження точки мінімуму функції $f(X)$ в інтервалі $[a, b]$.

Крок 1. Припустимо $L = b - a$, $x_1 = a$, $x_3 = a + L/2$, $x_5 = b$. Обчислити значення $f(x_1)$, $f(x_3)$ та $f(x_5)$.

Крок 2. Припустимо $x_2 = a + L/4$, $x_4 = b - L/4$. Замітити, що точки x_2, x_3 і x_4 ділять інтервал $[a, b]$ на чотири рівні частини. Обчислити значення $f(x_2)$ та $f(x_2)$.

Крок 3. Зрівняти значення $f(x_k)$, де $k = 1, ..., 5$, знайти найменше значення $f(x_m)$ серед обчислених $f(x_k)$. Якщо $x_m \le x_2$, виключити інтервали $[x_3, b]$, поклавши $b = x_3$. Якщо $x_m = x_3$, виключити інтервали $[a, x_2]$ та $[x_4, b]$. Припустимо $a = x_2$ і $b = x_4$. Якщо $x_m = x_4$, виключити інтервал $[a, x_3]$. Припустимо $a = x_3$ і $a = x_5$. Перейти до кроку 4.

Крок 4. Обчислити $L = b - a$. Якщо значення $L/4$ мале, закінчити пошук. В протилежному випадку повернутися до кроку 2.

Зауваження.

1.На кожній ітерації потребується не більше двох обчислень значення функції.

2.Показано [13], що з усіх методів пошуку на рівних інтервалах (чотириточкових, п′ятиточкових, шеститочкових і т.д.) п′ятиточковий відрізняється найбільшою ефективністю.

Абсолютна помилка в знаходженні положення екстремуму з збільшенням числа точок, в яких обчислюється значення цільової функції, визначається виразом

$$\Delta = (b - a)\, 2^{-\frac{N-1}{2}} \tag{1.1}$$

Завдання.

Реалізувати процедуру одновимірного пошуку точки оптимуму методом ділення інтервалу на чотири частини функції:

а) $f(X) = 3x^2 + \left(\frac{12}{x^3}\right) - 5$ у інтервалі $\frac{1}{2} \le x \le \frac{5}{2}$;

б) $f(X) = (10x^3 + 3x^2 + x + 5)$, $-2 \le x \le 1$;

в) $f(X) = 3x^4 + (x - 1)^2$, у інтервалі $0 \le x \le 4$;

г) $f(X) = 4x\sin x, \quad -1 \le x \le 3{,}14$;

д) $f(X) = 2(x-3)^2 + e^{0{,}5x^2}, \quad 0 \le x \le 100$.

Завдання для самостійної роботи.

Задані наступні функції однієї змінної:

а) $f(X) = x^5 + x^4 - \left(\dfrac{x^3}{3}\right) + 2$;

б) $f(X) = (2x+1)^2 (x-4)$.

Для кожної із заданих функцій знайти:

1) інтервал(и) зростання, убування;
2) точки перегину (якщо такі є);
3) інтервали, в яких функція вгнута, опукла;
4) локальні і глобальні максимум (мінімум) (якщо є).

Контрольні запитання.

1. Як перевірити, являється функція випуклою або вгнутою? Визначити, які з наступних функцій випуклі опуклі або вгнуті:

а) $f\left(X\right) = e^{x}$;

б) $f\left(X\right) = e^{-x}$;

в) $f\left(X\right) = \dfrac{1}{x^2}$;

г) $f\left(X\right) = x + \log x,\quad x > 0$.

2. У чому полягає властивість унімодальності функції і в чому його важливе значення при рішенні задач оптимізації з однією змінною?

Метод золотого перерізу

Величина підінтервалу, що виключається на кожному кроці пошуку оптимуму, залежить від знаходження пробних точок всередині інтервалу пошуку. Оскільки місцезнаходження точки оптимуму невідомо, доречно припустити, що розміщення пробних точок повинно забезпечувати зменшення інтервалу в одному і тому ж відношенні. Крім того, в цілях збільшення ефективності алгоритму необхідно вимагати, щоб вказане відношення було максимальним. Подібну стратегію називають стратегією пошуку.

На основі методів виключення інтервалів і мінімаксних стратегій пошуку можна зробити наступні висновки.

1. Якщо кількість пробних точок приймається рівним двом, то їх слід розміщувати на однакових відстанях від середини інтервалу.

2. У відповідності з загальною максимальною стратегією пробні точки повинні розміщувати в інтервалі по симетричній схемі таким чином, щоб відношення довжини під інтервалу, що виключається до довжини інтервалу пошуку залишалося постійним.

Одним з найбільш ефективних способів мінімаксної стратегії пошуку являється пошук за допомогою золотого перерізу, оснований на розбитті відрізку прямої на дві частини у відношенні, відомому як золотий переріз. При цьому відношенні довжини всього відрізку до більшої частини дорівнює відношенню більшої частини до меншої. Розглянемо симетричне розташування двох пробних точок x_2 та x_3 на вихідному інтервалі $[a, b]$, де $L = b - a$, $x_1 = a$ і $x_4 = b$, яке показано на рис. 1.1.

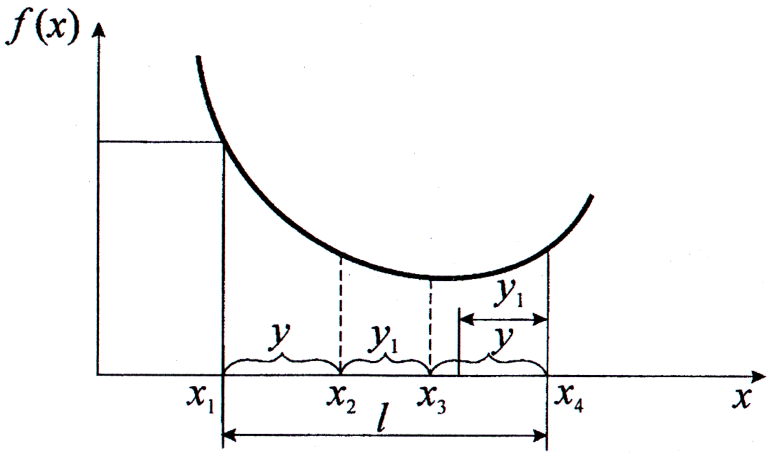


Рис. 1.1. Пошук за допомогою золотого перерізу

Довжини інтервалів [x_1, x_3] та [x_2, x_4] однакові, тобто $x_3 - x_1 = x_4 - x_2$ інтервали [x_1, x_2] і [x_3, x_4] також рівні. Позначимо $y = x_2 - x_1 = x_4 - x_3$. Значення y вибирається таким чином, щоб

$$\frac{y}{l} = \frac{l - 2y}{l - y} \tag{1.2}$$

Значення y визначається з (4.2) як рішення рівняння $y^2 - 3ly + l^2 = 0$.

$$\frac{y}{l} = \frac{3 \pm \sqrt{5}}{2} \tag{1.3}$$

З двох коренів рівняння залишаємо

$$\frac{y}{l} = \frac{3 - \sqrt{5}}{2} \approx 0{,}38 \tag{1.4}$$

так як другий корінь $y/l>1$, що не відповідає постановці задачі. Знайдене значення відношення y/l носить назву золотого перерізу.

Після відкидання у вихідному відрізку однієї з крайніх його частин відрізок, що залишився при збереженні на місці його вибраної точки повинен ділитися в тому ж відношенні, що і вихідний.

Процедура пошуку мінімуму методом золотого перерізу складається з наступних кроків.

Крок 1. Припустимо $L = b - a$, $x_1 = a$, $x_4 = b$. Обчислити значення $f(x_1)$ та $f(x_4)$.

Крок 2. Припустимо $z = \frac{3 - \sqrt{5}}{2}$, $x_2 = x_1 + zL$, $x_3 = x_4 + zL$. Обчислити значення $f(x_2)$ і $f(x_3)$.

Крок 3. Вибрати скорочений підінтервал в інтервалі L, в якому локалізований мінімум.

Якщо $f(x_2) < f(x_3)$ то $x_4 = x_3$, $x_3 = x_2$, $x_2 = x_1 + (x_4 - x_3)$.

Якщо $f(x_2) > f(x_3)$ то $x_1 = x_2$, $x_2 = x_3$, $x_3 = x_1 + (x_4 - x_2)$.

Крок 4. Обчислити $x_2 - x_1$. Якщо величина $x_2 - x_1$ мала, закінчити пошук. В протилежному випадку повернутися до кроку 3.

Оцінку точності визначення екстремуму методом золотого перерізу при заданому числі розрахунків значень функції $f(x)$ можна визначити як

$$\Delta = \frac{b-a}{2}\left(\frac{\sqrt{5}-1}{2}\right)^{s-3} \tag{1.5}$$

де Δ – абсолютна помилка у визначенні положення екстремуму після S обчислень $f(x)$.

Завдання.

Реалізувати процедуру одновимірного пошуку оптимуму методом золотого перерізу функції:

а) $f(X) = 3x^2 + \left(\frac{12}{x^3}\right) - 5$ у інтервалі $\frac{1}{2} \le x \le \frac{5}{2}$;

б) $f(X) = (10x^3 + 3x^2 + x + 5)^2$, $-2 \le x \le 1$;

в) $f(X) = 3x^4 + (x-1)^2$, $0 \le x \le 4$;

г) $f(X) = 4x\sin x$, $-1 \le x \le 3{,}14$;

д) $f(X) = 2(x-3)^2 + e^{0{,}5x^2}$, $0 \le x \le 100$.

Порівняти результуючі інтервали пошуку, отримані за допомогою:

а) методу ділення інтервалу на чотири частини;

б) методу золотого перерізу.

Який з методів виявився більш ефективним? Чому?

Завдання для самостійної роботи.

Мінімізувати $f(x) = x^2 - x$ за допомогою методу золотого перерізу в інтервалі $0 \le x \le 1$. Виконати число обчислень функції, достатнє, щоб отримати $|\Delta x_k| < 10^{-3}$. Побудувати графік залежності $[f(x_{k+1}) - f(x_k)]$ від послідовності номерів k.

Контрольні питання.

1. Вказати необхідні та достатні умови мінімуму функції однієї змінної.

Метод пошуку з використанням чисел Фібоначчі

Для організації оптимального пошуку екстремуму можна використовувати ряд чисел Фібоначчі, властивості якого описуються рекурентним співвідношенням

$$F_k = F_{k-1} + F_{k-2} \tag{1.6}$$

де $F_0 = F_1 = 1$.

Для оцінки точності визначення екстремуму при заданому числі розрахунків s значень функції $f(x)$, які визначаються на інтервалі $[a, b]$, використовують формулу

$$\Delta = \frac{b-a}{F_s} \tag{1.7}$$

де F_s – s-е число Фібоначчі.

Порядок виконання алгоритму пошуку мінімуму, який використовує числа Фібоначчі, складається з наступних етапів.

Крок 1. По заданій точності Δ, з якою необхідно знайти положення екстремуму функції $f(x)$ в інтервалі $[a, b]$, розрахувати допоміжне число

$$N = \frac{b-a}{\Delta} \tag{1.8}$$

Для отримання значення N знайти таке число Фібоначчі F_s, щоб виконувалась нерівність $F_{s-1} < N < F_s$.

Крок 2. Визначити крок пошуку за формулою

$$\Delta_m = \frac{b-a}{F_s} \tag{1.9}$$

Крок 3. Розрахувати значення функції $f(x)$ на початку інтервалу, тобто $f(a)$.

Крок 4. Знайти наступну точку, в якій обчислюється значення $f(x)$, за формулою

$$x^1 = a + \Delta_m F_{s-2} \tag{1.10}$$

Якщо цей крок вийшов вдалим, тобто $f(\mathrm{x}^{(1)}) < f(a)$, то наступна точка визначається за формулою

$$x^2 = x^1 + \Delta_m F_{s-3} \tag{1.11}$$

При $f(x^{(1)}) > f(a)$

$$x^2 = x^1 - \Delta_m F_{s-3} \tag{1.12}$$

Крок 5. Наступні кроки виконуються з зменшуваною величиною кроку, яка для i-го кроку

$$\Delta x^{(i)} = \pm\Delta_m F_{s-i-2}.$$

Процес продовжується до тих пір, доки не будуть вичерпані всі числа Фібоначчі у зменшуваній послідовності:

$$F_{s-i-2} = F_{s-i} - F_{s-i-1},\ i = 0, 1, 2.$$

Показано, що алгоритм пошуку з використанням чисел Фібоначчі в границі при $s \to \infty$, тобто при пошуку з високою точністю, співпадає з методом золотого перерізу [6], так як відношення F_{s-1}/F_s прямує до значення $1 - z$, де $z = \dfrac{3-\sqrt{5}}{2} \approx 0.38$.

Завдання.

Знайти екстремальні точки наступних функцій методом пошуку, який використовує числа Фібоначчі:

а) $f(X) = x^3 + x,\ \ -2 \le x \le 2$;

б) $f(X) = x^4 + x^2,\ \ -1 \le x \le 1$;

в) $f(X) = 4x^4 - x^2 + 5,\ \ 0 \le x \le 2$;

г) $f(X) = (3x-2)^2(2x-3)^2,\ \ -1 \le x \le 1$;

д) $f(X) = 6x^5 - 4x^3 + 10,\ \ 0 \le x \le 1$.

Завдання для самостійної роботи.

У структурі капітальних вкладів на розвиток хімічного заводу важливе місце займають витрати на придбання і монтаж труб, а також витрати на встановлення насосного обладнання. Розглянемо проект трубопроводу довжиною L(м), який повинен забезпечувати подачу рідини зі швидкістю Q (м^3/хв). Вибір найбільш економного діаметра труби D (м) здійснюється за рахунок мінімізації функції затрат на придбання труб, насосів та прокачування рідини. Відомо, що функція витрат в одиницю часу у випадку, коли трубопровід складається з труб, виготовлених з вуглецевої сталі, і центрострімкого насосу з електродвигуном, може бути описана виразом

$$f(x) = 0{,}45L + 0{,}245LD^{1,5} + 325(hp)^{0,5} + 61{,}6(hp)^{0,925}102,$$

де $hp = 4{,}4\cdot 10^{-8}\dfrac{LQ^3}{D^5} + 1{,}92\cdot 10^{-9}\dfrac{LQ^{2,68}}{D^{4,68}}$.

Сформулювати відповідну задачу оптимізації з однією змінною для проектування трубопроводу довжиною 300 м, який повинен забезпечити подання рідини зі швидкістю 0,1 (м^3/хв). Діаметр труби має бути в межах

$0{,}6 \cdot 10^{-2}...0{,}2$ м. Вирішити цю задачу за допомогою методу пошуку з використанням чисел Фібоначчі.

Контрольні запитання

1. В чому суть алгоритму пошуку мінімуму, що використовує числа Фібоначчі?
2. Як обчислюється точність визначення екстремуму за допомогою методу пошуку, що використовує числа Фібоначчі.

Поліноміальна апроксимація

В ряді випадків виходить більш ефективним, ніж методи виключення інтервалів, метод пошуку, який дозволяє враховувати відносні вимірювання значень функції. Основна його ідея пов'язана з можливістю апроксимації гладкої функції полінома і послідовного використання апроксимуючого поліному для оцінки координати точки оптимуму. Необхідними умовами ефективної реалізації такого підходу є унімодальність та неперервність досліджуваної функції. Відповідно теоремі Вєйєрштрасса про апроксимації, кількість оцінок координати точки оптимуму, що отримуються за допомогою апроксимуючого поліному, можна підвищити двома способами: використанням поліному більш високого порядку і зменшенням інтервалу апроксимації. Другий спосіб являється переважним, оскільки побудова апроксимуючого поліному порядку вище третього стає складною процедурою, тоді як зменшення інтервалу в умовах, коли виконується припущення про унімодальність функції, особливої складності не являє.

Методи оцінювання з використанням квадратичної апроксимації

Простіший варіант поліноміальної інтерполяції – квадратична апроксимація, основана на тому, що функція, яка приймає мінімальне значення у внутрішній точці інтервалу, повинна бути по крайній мірі квадратичною.

Коггінс запропонував використовувати алгоритм Девіса, Свенна та Кемпі (ДСК) [15] для визначення інтервалу, який складає точку мінімуму, щоб усі подальші обчислення проводились за алгоритмом Пауелла [15].

При одновимірному пошуку за алгоритмом ДСК проводяться зростаючі за величиною кроки до тих пір доки не буде пройдений мінімум, а потім за алгоритмом Пауелла проводиться квадратична апроксимація і визначається точка x, відповідна мінімуму квадратичної функції. На рис. 1.2 зображена така процедура. Квадратична апроксимація продовжується до тих пір, доки з потрібною точністю не знаходиться мінімум $f(x)$.

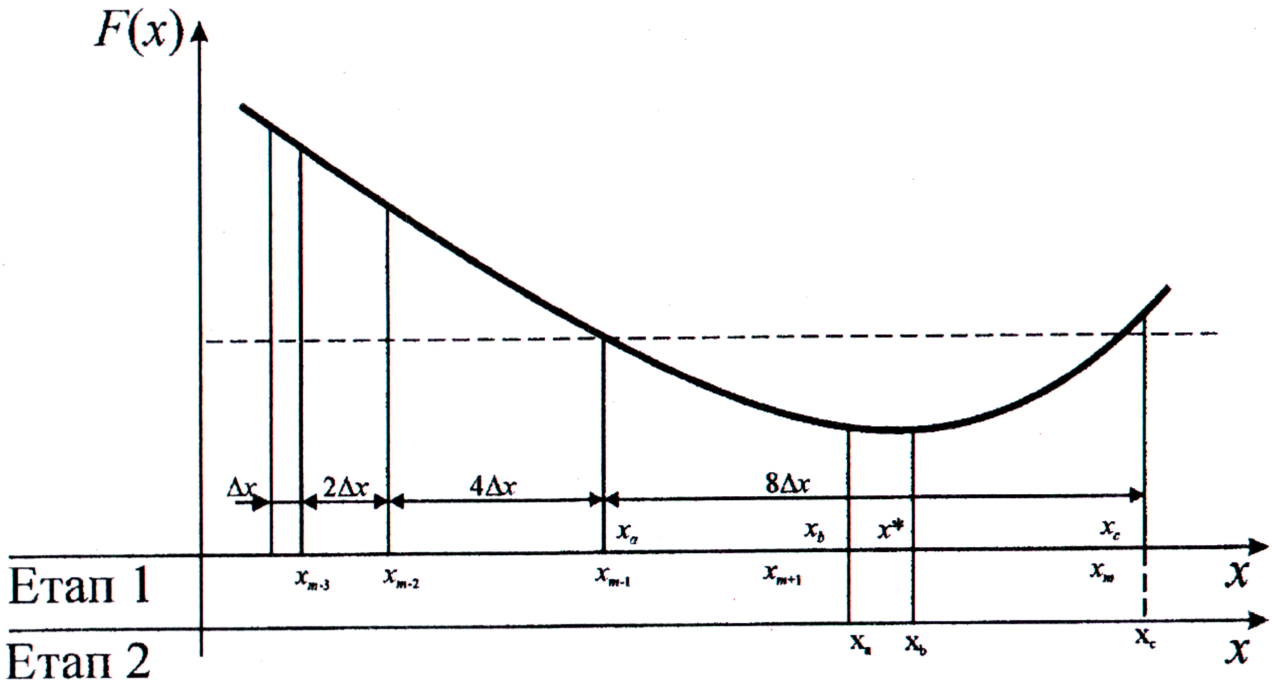


Рис. 1.2. Одновимірна мінімізація методом Коггінса

Крок 1. Обчислити $f(x)$ у початковій точці x_0. Якщо $f(x_0 + \Delta x) \leq f(x_0)$, то перейти до кроку 2. Якщо $f(x_0 + \Delta x) > f(x_0)$, то припустити $\Delta x = -\Delta x$ і перейти до кроку 2.

Крок 2. Обчислити $x_{k+1} = x_k + \Delta x$.

Крок 3. Обчислити $f(x_{k+1})$.

Крок 4. Якщо $f(x_{k+1}) \leq f(x_k)$, тоді подвоїти Δx та повернутися до кроку 2 при $k = k + 1$. Якщо $f(x_{k+1}) > f(x_k)$, то x_{k+1} позначити x_m, x_k – x_{m-1} і т.д., зменшити наполовину і повернутися до кроків 2 та 3.

Крок 5. З чотирьох рівновіддалених значень $x\{x_{m+1}, x_m, x_{m-1}, x_{m-2}\}$ вилучити або x_m, або x_{m-2} в залежності від того, яке з них знаходиться далі від x, відповідного найменшому значенню $f(x)$. Нехай x_a, x_b та x_b – залишені три значення x, де x_b – центральна точка, а $x_a = x_b - \Delta x$ і $x_c = x_b - \Delta x$.

Крок 6. Обчислити наближене значення x у точці мінімуму $f(x)$ за формумою

$$\tilde{X}^* = -\frac{1}{2}\frac{(x_b^2 - x_c^2) f(x_a) + (x_c^2 - x_a^2) f(x_b) + (x_a^2 - x_b^2) f(x_c)}{(x_b - x_c) f(x_a) + (x_c - x_a) f(x_b) + (x_a - x_b) f(x_c)} \qquad (1.13)$$

Крок 7. Якщо $\tilde{X}^*$ і будь-яке із значень $x\{x_a, x_b, x_c\}$, відповідних мінімуму $f(x)$, відрізняється менш, ніж наперед заданому точність x або на точність відповідних значень функції $f(x)$, тоді закінчити пошук. У протилежному випадку обчислити $f(\tilde{X}^*)$ та вилучити з множини $\{x_a, x_b, x_c\}$ те значення x, яке відповідає найбільшому значенню $f(x)$, якщо, однак, при цьому не буде загублений інтервал, в якому знаходиться мінімум $f(x)$. У цьому випадку слід так вилучити x, щоб зберегти цей інтервал. Перейти до кроку 6.

Робота алгоритму продовжується до тих пір, доки не буде досягнута бажана точність, вказана у кроці 7.

Завдання.

Знайти точку мінімуму функції $f(x) = (10x^3 + 3x^2 + x + 5)^2$. Задані початкова точка $x = 2$ та довжина кроку $\Delta = 0,5$.

Порівняти отриманий результуючий інтервал пошуку з отриманим за допомогою методу золотого перерізу.

Завдання для самостійної роботи. Мінімізувати $f(x) = x^2 - x$ починаючи з точки $x = 3$, за допомогою методу Коггінса. Нехай $|\Delta x_0| = 0,1$. Виконати кількість обчислень функції, достатнє, щоб отримати $|\Delta x_k| < 10^{-3}$. Побудувати графік залежності $[f(x_{k+1}) - f(x_k)]$ від послідовності номерів k. Визначити інтервал, в якому знаходиться мінімум, методом золотого перерізу.

Контрольні запитання.

1. Сформулювати умову, при виконанні якої метод пошуку, оснований на поліноміальній інтерполяції, може не привести до отримання вірного рішення.

2.Чи являються методи виключення інтервалів в цілому більш ефективними, ніж методи точкового оцінювання? Чому?

Порівняння методів одновимірного пошуку.

За допомогою теоретичних висновків показано, що такі методи точкового оцінювання, як метод Коггінса або метод пошуку з використанням кубічної апроксимації, суттєво ефективніше методів виключення інтервалів, серед яких виділяється метод золотого перерізу. У випадку, коли інтервали збігу порівняні між собою, метод оснований на квадратичній апроксимації, збігається швидше, ніж будь-який з методу виключення інтервалів.

Результати експериментальних досліджень довели [13], що якщо необхідно отримати рішення з високим ступенем швидкості, тоді кращими виявляються методи пошуку на основі поліноміальної апроксимації. З другого боку, доведено, що при дослідженні мультимодальних (швидко - змінних) функцій метод Коггінса збігається повільніше, ніж метод виключення інтервалів.

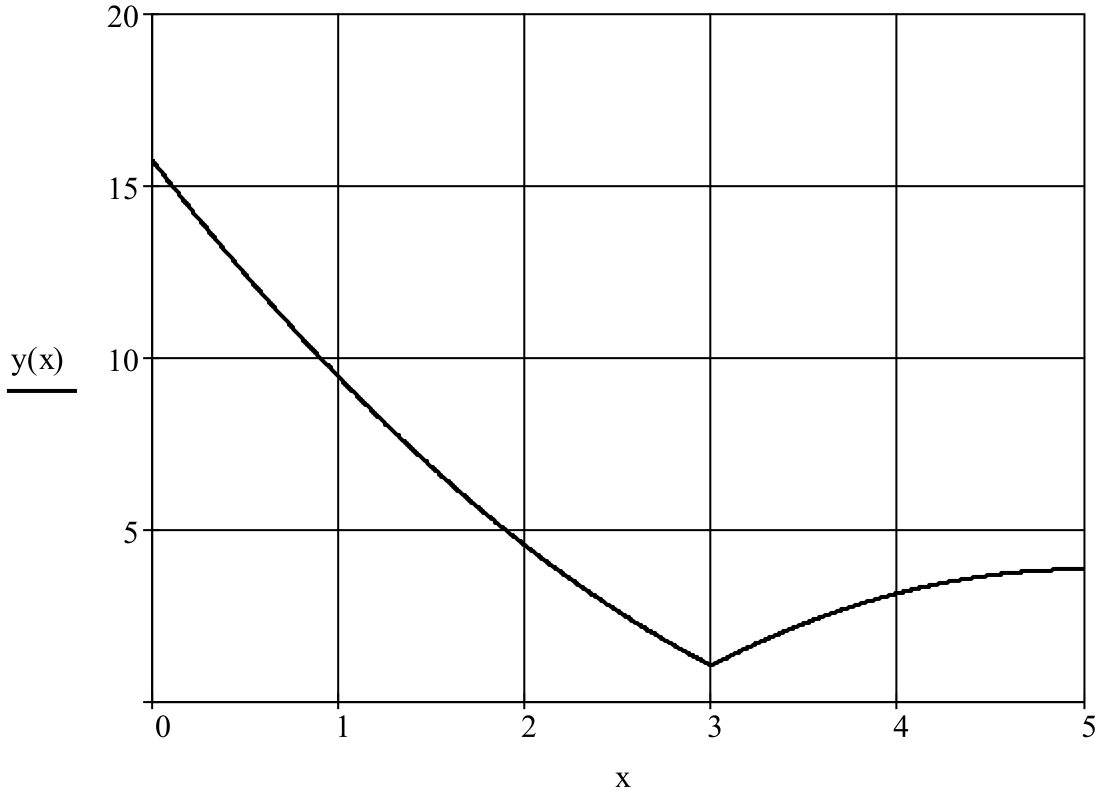
Для оцінки ефективності методів використовувались три характеристики: час, витрачений на отримання рішення, точність рішення та чутливість до зміни параметру збігу.

Методи детермінованого пошуку для функції однієї змінної

Завдання:

Знайти мінімум функції на інтервалі (A, B)

$$y(x) := 0.7 \cdot \left| x^2 - 10 \cdot x + 21 \right| + 1$$



$$\text{metod\_dihotomii}(A, B, \varepsilon, \Delta) := \left| \begin{array}{l} i \leftarrow 0 \\ \text{while } \; B - A > \varepsilon \\ \quad \left| \begin{array}{l} i \leftarrow i + 1 \\ L \leftarrow \dfrac{A + B - \Delta}{2} \\ R \leftarrow \dfrac{A + B + \Delta}{2} \\ A \leftarrow L \;\; \text{if } \; y(L) > y(R) \\ B \leftarrow R \;\; \text{if } \; y(L) < y(R) \\ c_i \leftarrow A \\ d_i \leftarrow B \end{array} \right. \\ a \leftarrow A \\ b \leftarrow B \\ x \leftarrow \dfrac{a + b}{2} \\ \begin{pmatrix} i \\ x \\ \text{augment}(c, d) \end{pmatrix} \end{array} \right.$$

$$\text{metod\_dihotomii}(0, 5, 0.01, 0.001) = \begin{pmatrix} 10 \\ 3 \\ \{11,2\} \end{pmatrix}$$

$$\begin{pmatrix} i \\ x \\ \text{Iteracii} \end{pmatrix} := \text{metod\_dihotomii}(0, 5, 0.01, 0.001)$$

Iteracii =

	0	1
0	0	0
1	2.499	5
2	2.499	3.75
3	2.499	3.125
4	2.812	3.125
5	2.968	3.125
6	2.968	3.047
7	2.968	3.008
8	2.988	3.008
9	2.997	3.008
10	2.997	3.003

$i = 10 \quad y(x) = 1.001$

Метод золотого перерізу

$$
\text{metod\_zp}(A,B,eps) :=
\left|
\begin{array}{l}
z \leftarrow \dfrac{3-\sqrt{5}}{2} \\
i \leftarrow 0 \\
\text{while } \; B - A > eps \\
\quad \left|
\begin{array}{l}
x1 \leftarrow A \\
x4 \leftarrow B \\
x2 \leftarrow x1 + z\cdot(B-A) \\
x3 \leftarrow x4 - z\cdot(B-A) \\
i \leftarrow i+1 \\
\text{if } \; y(x2) < y(x3) \\
\quad \left|
\begin{array}{l}
x4 \leftarrow x3 \\
x3 \leftarrow x2 \\
x2 \leftarrow x1 + (x4 - x3)
\end{array}
\right. \\
\text{if } \; y(x2) > y(x3) \\
\quad \left|
\begin{array}{l}
x1 \leftarrow x2 \\
x2 \leftarrow x3 \\
x3 \leftarrow x1 + (x4 - x2)
\end{array}
\right. \\
A \leftarrow x1 \\
B \leftarrow x4 \\
c_i \leftarrow A \\
d_i \leftarrow B
\end{array}
\right. \\
x \leftarrow \dfrac{A+B}{2} \\
\begin{bmatrix}
x \\
i \\
\dfrac{B-A}{2}\cdot\left(\dfrac{\sqrt{5}-1}{2}\right)^{i-3} \\
\text{augment}(c,d)
\end{bmatrix}
\end{array}
\right.
$$

$$\begin{pmatrix} x \\ i \\ delta \\ Iteracii \end{pmatrix} := metod\_zp(0, 5, 0.01)$$

Iteracii =

	0	1
0	0	0
1	1.91	5
2	2.639	3.82
3	2.639	3.369
4	2.812	3.09
5	2.918	3.09
6	2.959	3.024
7	2.984	3.024
8	2.993	3.009
9	2.997	3.003

$x = 3 \quad y(x) = 1 \quad i = 9 \quad delta = 1.653 \times 10^{-4}$

Метод пошуку з використанням чисел Фібоначчі

```
metod_fibonachi(A,B,n) := | Fi_0 ← 1
                          | Fi_1 ← 1
                          | for k ∈ 2..n
                          |   Fi_k ← Fi_(k−1) + Fi_(k−2)
                          | A_0 ← A
                          | B_0 ← B
                          | for s ∈ 0..n − 3
                          |   | G_s ← Fi_(n−s−1) / Fi_(n−s)
                          |   | L_s ← B_s − G_s·(B_s − A_s)
                          |   | R_s ← A_s + G_s·(B_s − A_s)
                          |   | if y(L_s) ≤ y(R_s)
                          |   |   | A_(s+1) ← A_s
                          |   |   | B_(s+1) ← R_s
                          |   | if y(L_s) > y(R_s)
                          |   |   | A_(s+1) ← L_s
                          |   |   | B_(s+1) ← B_s
                          |   | c_s ← A_s
                          |   | d_s ← B_s
                          | x ← (A_s + B_s) / 2
                          | ( x ; augment(A,B) )
```

$$\begin{pmatrix} x \\ Iteracii \end{pmatrix} := metod\_fibonachi(0,5,22)$$

Iteracii =

	0	1
0	0	5
1	1.91	5
2	1.91	3.82
3	2.639	3.82
4	2.639	3.369
5	2.639	3.09
6	2.812	3.09
7	2.918	3.09
8	2.918	3.024
9	2.959	3.024
10	2.984	3.024
11	2.984	3.009
12	2.993	3.009
13	2.993	3.003
14	2.997	3.003
15	2.999	3.003
16	2.999	3.002
17	2.999	3.001
18	3	3.001
19	3	3
20	3	3

$x = 3 \quad y(x) = 1$

Лабораторна робота № 2

Функції декількох змінних

Метод сканування

Метод сканування полягає у послідовному перегляді значень функції мети у ряді точок, які належать області зміни незалежних змінних і знаходження серед них мінімального (максимального) значення функції мети [6]. Точність методу визначається тим, наскільки щільно розташовуються обрані точки у припустимій області зміни незалежних змінних.

Переваги методу – при використанні щільного розташування досліджуваних точок гарантується відшукування оптимуму, а також незалежність пошуку від вигляду функції, що оптимізується.

Недолік методу – обчислення значень цільової функції для великої кількості точок.

Найбільш простий алгоритм пошуку оптимуму методом сканування, який називається пошуком на сітці змінних, полягає у приростах, які дають у відповідному порядку по кожній з незалежних змінних і забезпечують заповнення всієї області зміни рівномірною і достатньо щільною сіткою (рис. 2.1).

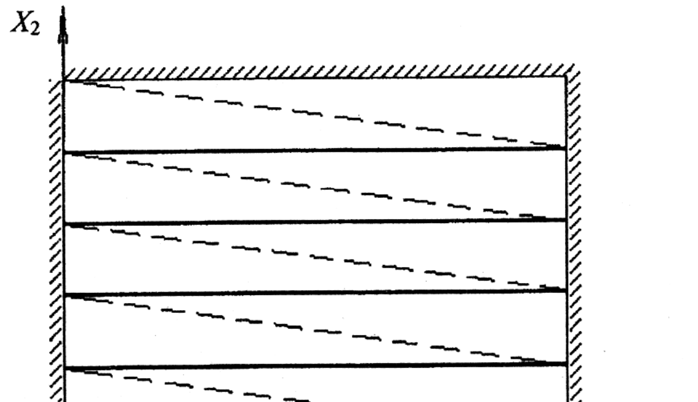


Рис. 2.1. Пошук оптимуму для випадку двох змінних.

Додаткові обмеження на незалежні змінні не ускладнюють процедуру використання методу, так як у цьому випадку точки, які не задовольняють заданим умовам, просто вилучаються з розгляду і значення функції мети не обчислюються.

Для оцінки обчислювальних затрат методом сканування визначають кількість обчислень функції мети при умові, що точність визначення положення оптимуму дорівнює Δ:

$$S = \left(\frac{1}{\Delta}\right)^n, \qquad (2.1)$$

де n – кількість незалежних змінних.

Для скорочення об’єму обчислень застосовують різні модифікації методу сканування, наприклад метод пошуку з змінним кроком сканування. Початкова величина кроку обирається достатньо великою, по можливості такою, що перевищує точність визначення положення оптимуму. Виконується грубий пошук, який локалізує область знаходження глобального оптимуму. Коли область визначена, проводиться пошук з меншим кроком. Необхідний об’єм обчислень значень цільової функції скорочується:

$$S = k^{-m}\left(\frac{1}{\Delta}\right)^n + r\;\;2k\;\;^{n}, \qquad (2.2)$$

де r – число етапів уточнення пошуку, на яких крок пошуку зменшується у к разів.

На рис. 2.2 показаний пошук з змінним кроком для функції двох змінних.

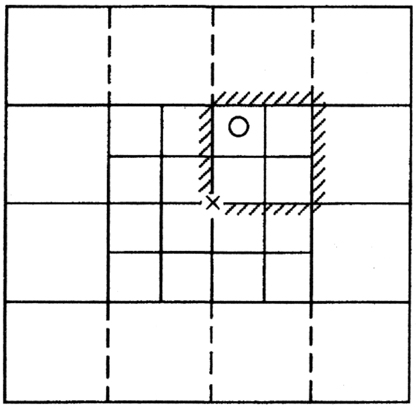


Рис. 2.2. Пошук методом сканування з змінним кроком: хрестиком позначене значення оптимуму, знайдене у наслідок грубого пошуку, кружком - справжнє положення оптимуму

Порядок виконання пошуку процедури мінімуму функції $f(x)$ методом сканування зі змінним кроком складається з наступних етапів.

Крок 1. Обчислити крок сканування по кожній змінній

$$H\;\;z\;\; = \frac{B\;\;z\;\; - A\;\;z\;\;}{2K\;\;z\;\;} \qquad (2.3)$$

і початковий крок сканування по першій змінній

$$HO = \frac{B\ 1\ - A\ 1}{L} \tag{2.4}$$

де A, B – ліва та права межі вихідної області; k – кратність дроблення по кожній змінній; L – кратність дроблення по першій змінній.

Крок 2. Для ряду значень першої змінної, які відстають один від одного на величину кроку H,

$$X\ 1\ = A\ 1\ + H\ 1\ Z\ 1 \tag{2.5}$$

і при фіксованих значеннях інших змінних обчислити значення функції мети f. Якщо $\left[f\ \ x_{k+1}^{i}\ \ < f\ \ x_{k}^{i}\ \ \right]$, то переприсвоїти $y(z) = x(z), f_1 = f$. Знайти мінімальне значення функції мети f_1.

Крок 3. При першому значенні другої змінної, яке відстає на величину кроку від попереднього значення,

$$X\ 2\ = A\ 2\ + H\ 2\ Z\ 2 \tag{2.6}$$

та фіксованих значеннях інших змінних перейти до кроку 2 і т.д.

Крок 4. Якщо весь діапазон зміни змінних досліджений, перевірити умову збігу за величиною кроку по першій змінній. Якщо $\frac{B\ 1\ - A\ 1}{K\ 1} \le HO$, то пошук закінчується; у протилежному випадку локалізується область знаходження оптимуму

$$A\ z\ = Y\ z\ + H\ z\ ,\ \ B\ z\ = Y\ z\ + H\ z\ , \tag{2.7}$$

і крок пошуку зменшується кратно к $H(z) = H(z)/K$, здійснюється повернення до кроку 1.

Завдання.

Використовуючи програми 1 та 2 досліджувати функцію

$f(x) = x^3 - 12x + 3$ на інтервалі $-4 \le x \le 4$

Знайти локальний мінімум і глобальний мінімум в заданому інтервалі. Зрівняти об'єми обчислень методів сканування з постійним та змінним кроком.

Завдання для самостійної роботи.

Використовуючи програму 3, знайти екстремальні точки наступних функцій:

$$f(x) = x_1^3 + x_2^3 - 3x_1x_2,\ \ 0 \le x_1 \le 5,\ \ 0 \le x_2 \le 5.$$

Контрольні запитання:

1. Нехай дана точка задовольняє достатнім умовам існування локального мінімуму. Як встановити, чи являється цей мінімум глобальним?

2. Модифікації методу сканування.

Метод Гаусса-Зейделя

Метод почергової зміни змінних, який називається також методом Гаусса-Зейделя полягає в тому, що почергово змінюються всі незалежні змінні так, щоб кожна з них досягала найменше (найбільше) значення цільової функції. Черговість варіювання незалежних змінних встановлюється довільно і звичайно не змінюється в процесі пошуку. Кожна уточнююча змінна варіюється до тих пір поки в даному осьовому напрямку не буде знайдено мінімум, після чого починається процес крокового пошуку по наступному осьовому напрямку. Стратегія пошуку мінімуму по кожній змінній при цьому може бути будь-якою. Зокрема можна використовувати один із описаних вище методів пошуку оптимуму функції однієї змінної.

Відзначимо, що для двох незалежних змінних метод почергового змінення змінних та метод релаксації (градієнтний метод пошуку оптимуму) співпадають. Метод Гаусса-Зейделя приводить до оптимуму більш довгим шляхом. Однак загальний об'єм обчислень порівняно з методом релаксації в даному випадку може стати меншим, так як при переході до уточнення наступної змінної похідні цільової функції не обчислюються. До недоліків методу Гаусса-Зейделя відносяться труднощі пошуку при наявності обмежень або особливостей цільової функції.

Порядок виконання пошукової процедури мінімуму функції $f(x)$ методом Гаусса-Зейделя складається із наступних етапів:

Крок 1. Обчислити f(x) при початкових координатах змінних.

Крок 2. При зміні однієї змінної, якщо $f\left(X_0^i+\Delta X^i\right)\le f\left(X_0^i\right)$, то перейти до кроку 3; якщо $f\left(X_0^i+\Delta X^i\right)>f\left(X_0^i\right)$, то прийняти $\Delta X^i=-\Delta X^i$ і перейти до кроку 3.

Крок 3. Обчислити $X_{k+1}^i=X_k^{(i)}+\Delta X^i$.

Крок 4. Обчислити $f\left(X_{k+1}^i\right)$.

Крок 5. Якщо $f\left(X_{k+1}^{i}\right) < f\left(X_{0}^{i}\right)$, то переприсвоїти і повернутися до кроку 2. Якщо перевіряється умова збігу метода по величині кроку.

При виконанні, коли величина кроку H_k стане менше заданої точності, пошук закінчується; в протилежному випадку величина кроку пошуку зменшується кратно K, проходить повернення до кроку 2.

Завдання.

Використовуючи метод Гаусса-Зейделя знайти мінімум функції

а) $f(x_1, x_2) = 10x_1^2 + 2(x_2 - 5)^2$, $X^0 = (1 \quad 2)$;

б) $f(x_1, x_2) = 3(x_1 + 1)^2 + 2(x_2 - 5)^2$, $X^0 = (0 \quad 0)$;

в) $f(x_1, x_2) = 2(x_1 - 1)^2 + 3(x_2 - 2)^2$, $X^0 = (3 \quad 4)$;

Переробити програму для пошук максимуму функції по методу Гаусса-Зейделя, який знайти для функції

а) $f(x_1, x_2) = 2 + 2x_1 + x_2 + x_1^2 + 2x_2^2 + x_1x_2$, $X^0 = (3 \quad 3)$;

б) $f(x_1, x_2) = x_1^2 + x_2^2 - x_1$, $X^0 = (1 \quad 2)$;

в) $f(x_1, x_2) = 2 + 2x_1 + 4x_2 - x_1^2 - x_2^2$, $X^0 = (1 \quad 2)$;

Завдання для самостійної роботи. Знайти мінімум функції

$f(x_1, x_2) = (x_1 - 5)^2 + 3(x_2 - 3)^2$

за допомогою метода Гаусса-Зейделя, використовуючи пошук мінімуму по кожній змінній методом золотого перерізу. В якості початкової точки вибрана точка $X^0 = (2 \quad 1)$.

Контрольні запитання:

1. Область застосування методу Гаусса-Зейделя.
2. Зрівняти методи Гаусса-Зейделя та релаксації.

Симплексний метод

Спробою уникнути необхідності обчислення похідних для визначення напрямку найшвидшого посування до оптимуму і в той же час зберегти можливість достатньо швидкого руху до нього являється алгоритм симплексного методу [13].

Процедура симплексного пошуку Спендлі, Хекста та Хімсворта базується на тому, що експериментальним зразком, який складається з найменшої кількості точок являється регулярним симплексом. Регулярний симплекс в N-мірному просторі являє собою багатокутник, створений N+1 рівновіддаленими одна від одної точками-вершинами. Наприклад, у випадку двох змінних симплексом являється рівносторонній трикутник, в тривимірному просторі – тетраедр (рис.2.3)

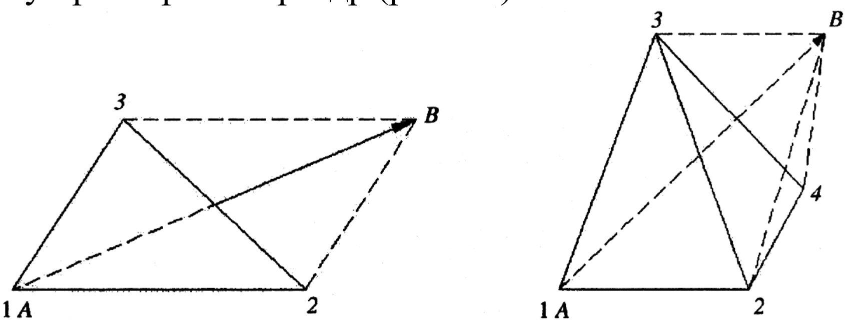


Рис.2.3 Регуляторні симплекси для випадку двох (а) та трьох (б) незалежних змінних: вершина 1 означає найбільше значення f(x), стрілка вказує найшвидше покращення

Із аналітичної геометрії відомо, що координати вершин регулярного симплексу визначаються матрицею D, в якій стовпці являють собою вершини, пронумеровані від 1 до N+1, а рядки – координати (і приймає значення від 1 до N):

$$D=\begin{bmatrix} 0 & d_1 & d_2 & ... & d_2 \\ 0 & d_2 & d_1 & ... & d_2 \\ 0 & d_2 & d_2 & ... & d_2 \\ ... & ... & ... & ... & ... \\ 0 & d_2 & d_2 & ... & d_1 \end{bmatrix} \tag{2.8}$$

де $d_2=\dfrac{t}{N\sqrt{2}}(\sqrt{N+1}+N-1)$; $d_2=\dfrac{t}{N\sqrt{2}}(\sqrt{N+1}-1)$

t – відстань між вершинами.

Наприклад, для N = 2 та t = 1 трикутник на рис.4.5 має слідуючи координати:

Таблиця 2.1

Вершина	$X_{1,i}$	$X_{1,i}$
1	0	0
2	0,965	0,259
3	0,259	0,965

Цільову функцію можна обчислити в кожній з вершин симплексу; з вершини, де цільова функція максимальна (точка A на рис.4.5), проводять проектуючу пряму через центр тяжіння симплексу. Потім точку A виключають і будують новий симплекс, що називається відбитим, з попередніх точок, що залишилися та однієї нової точки B, розташованої на проектуючій прямій на належній відстані від центру тяжіння. Якщо функція зменшується достатньо плавно, ітерації продовжують до тих пір, поки або не буде накрита точка мінімуму (рис.2.4), або почнеться циклічний рух по двох та більше симплексах.

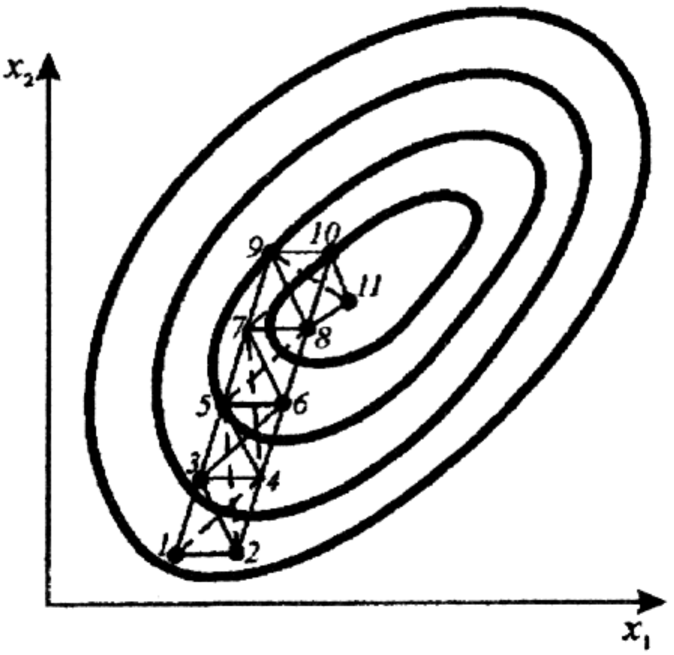


Рис.2.4. Послідовність регулярних симплексів отриманих при мінімізації цільової функції двовимірному просторі

В таких ситуаціях можна скористатися наступними правилами:

Правило 1. Накриття точки мінімуму.

Якщо вершина, якій відповідає найбільше значення цільової функції, побудована на попередній ітерації, то замість неї береться вершина, якій відповідає наступне за величиною значення цільової функції.

Правило 2. Циклічний рух.

Якщо деяка вершина симплексу не виключається протягом більш ніж M ітерацій, то необхідно зменшити розміри симплексу за допомогою коефіцієнту редукції і побудувати новий симплекс, вибравши в якості базової точку, якій відповідає мінімальне значення цільової функції. Спендлі, Хекст і Хімсворт запропонували обчислити значення М по формулі:

$$M = 1{,}65\,N + 0{,}05\,N^2$$

де N – розмірність задачі, а M округлюється до найближчого цілого числа.

Правило 3. Критерій закінчення пошуку.

Пошук завершується, коли або розмір симплексу, або різниці між значеннями функцій у вершинах стають достатньо малими.

Відображення відносно центру тяжіння являє неважку процедуру. Нехай $X^{(j)}$ точка, підлягаюча відображенню. Центр тяжіння N точок розташованій в точці

$$X_C = \frac{1}{N} \sum_{\substack{i=0 \\ j=0}}^{N} X^{\ i} \qquad (2.9)$$

Всі точки прямої, що проходить через точки $X^{(j)}$ та, X_C задаються формулою

$$X = X^{\ j} + \lambda \ X_C - X^{\ j} \qquad (2.10)$$

При $\lambda = 0$ отримуємо вихідну точку $X^{(j)}$, тоді як значення λ=1 відповідає центру тяжіння X_C. Для того, щоб побудований симплекс володів властивістю регулярності, відображення повинно бути симетричним. Відповідно, нова вершина отримується при $\lambda = 2$. Таким чином

$$X^{\ j}_{\text{нов}} = 2X_C - X^{\ \text{j}}_{\text{поп}} \qquad (2.11)$$

Алгоритм володіє рядом суттєвих недоліків:

1) не виключено виникнення труднощів, зв'язаних з масштабуванням, оскільки всі координати вершин симплексу залежать від одного і того ж масштабного множника *t*;

2) алгоритм працює дуже повільно, так як отримана на попередніх ітераціях інформація не використовується для прискорення пошуку;

3) не існує простого способу розширення симплексу, не потребуючого перерахунку значень цільової функції у всіх точках зразка.

Завдання.

Визначити регулярну симплексну фігуру в тримірному просторі таку, що відстань між вершинами дорівнює 0.2 і одна вершина знаходиться в точці (–1.2,–2).

Використовуючи симплексну фігуру, побудовану в п.1, провести вісім циклів відкидання вершин та отримати нові при пошуку мінімуму цільової функції

$$f(X) = x_1^2 + 3x_2^2 + 5x_3^2.$$

Тривимірний оптимальний симплекс пошук дав слідуючи проміжні результати

Таблиця 2.2

Вектор	Значення цільової функції
$[0\ 0\ 0]^T$	4
[–1/3 –1/3 –1/3]	7
[–1/3 –4/3 –1/3]	10
[–1/3 –1/3 –4/3]	5

Яка наступна точка підлягає обчисленню в процесі пошуку?

Яка точка пропущена?

Що буде центром тяжіння нового симплексу?

Завдання для самостійної роботи: число змінних дорівнює 2. Обмеження відсутні. Потрібно мінімізувати $f(X) = 100(X_i - X_1^2)^2 + (1 - X_1)^2$ за допомогою симплекс методу. В якості почергової точки вибирається точка $X^{(0)} = [-1{,}2\ 1]^T$ в якій $f(X^{(0)}) = 24{,}20$.

Контрольні запитання

1. В чому полягає властивість спаду цільової функції при переході від ітерації до ітерації? Чому виконання цієї властивості необхідне для побудови ефективного алгоритму?
2. Чому квадратичні функції використовуються як основа для побудови алгоритмів нелінійної оптимізації?
3. Пояснити поняття квадратичного збігу.
4. Переваги та недоліки симплексного методу.

Пошук по деформованому багатограннику

Практичні труднощі , які зустрічаються при використанні регулярних симплексів, а саме відсутність прискорення пошуку на скривлених ярах і хребтах, привели до необхідності деяких покращених методів. Нелдер і Мід [15] запропонували метод пошуку більш складний, порівняно з методом пошуку Спендлі, Хекста і Хімсворта , але який є ефективнішим і легко здійснюваним на ЕОМ. В зв’язку з тим, що симплекс може змінювати свою форму і таким чином вже не буде залишатись симплексом, метод пошуку названо по деформованому багатограннику.

Алгоритм пошуку мінімуму за допомогою методу Нелдера і Міда може бути описаним наступним чином.

Нехай вектор $X_i^{(k)} = [X_{i1}^{(k)}, ..., X_{ij}^{(k)}, ..., X_{in}^{(k)}]$, $i = 1, ... n + 1$ є i-ою вершиною в n- мірному просторі на k-ому етапі пошуку ($k = 0, 1, ...$) та хай значення цільової функції в $X_i^{(k)}$ дорівнює $f(X_i^{(k)})$. Крім того , відмітимо ті вектори X багатокутника, котрі дають максимальне та мінімальне значення $f(X)$. Визначимо

$$f\left(X_h^{k}\right) = \max\left\{f\left(X_l^{k}\right), \ldots, f\left(X_{n+1}^{k}\right)\right\} \tag{2.12}$$

де $X_l^{(k)} = X_i^{(k)}$, та

$$f\left(X_l^{k}\right) = \max\left\{f\left(X_l^{k}\right), \ldots, f\left(X_{n+1}^{k}\right)\right\} \tag{2.13}$$

Оскільки багатокутник в n-мірному просторі складається з n+1 вершин $X_1, ..., X_{n+1}$ буде центром тяжіння всіх вершин , виключаючи X_h.

Координати центру тяжіння визначаються формулою

$$X_{n+2}^{k}=\frac{1}{n}\left[\sum_{i=1}^{n+1}X_{ij}^{k}-X_{hj}^{k}\right],\quad j=1,...,n, \tag{2.14}$$

де j- координатний напрямок.

Початковий багатогранник звичайно вибирається у вигляді регулярного симплексу з точкою 1 в якості початку координат , початок координат можливо помістити в центр тяжіння. Процедура відшукання вершин в n-мірному просторі , в якій $f(X)$ має краще значення , складається з наступних операцій.

1. Відображення.

Проектування $X_h^{(k)}$ через центр тяжіння в відповідності співвідношенню

$$X_{n+3}^{k}=X_{n+1}^{k}+\alpha\ X_{n+2}^{k}-X_{h}^{k} \tag{2.15}$$

де α>0 – коефіцієнт відображення; $X_{n+1}^{(k)}$ – центр тяжіння, обчислений за формулою (2.14); $X_n^{(k)}$ – вершина, в якій функція $f(X)$ приймає найбільше з n+1 її значень на k-ому етапі.

2. Розтяг.

Полягає в наступному: якщо $f(X_{n+3}^{(k)})\leq f(X_i^{(k)})$ то вектор $X_{n+3}^{(k)}-X_{n+2}^{(k)}$ розтягнеться у відповідності з співвідношенням

$$X_{n+4}^{k}=X_{n+2}^{k}+\gamma\ X_{n+2}^{k}-X_{h}^{k} \tag{2.16}$$

де $j>1$ – коефіцієнт розтягу, якщо $f(X_{n+4}^{(k)})<f(X_i^{(k)})$ то $X_h^{(k)}$ заміняється на $X_{n+4}^{(k)}$ і процедура продовжується знову з операції 1 при $k=k+1$. В протилежному випадку $X_h^{(k)}$ замінюється на $X_{n+3}^{(k)}$ і також здійснюється перехід до операції 1 при $k=k+1$.

3.Стиск.

Якщо $f(X_{n+3}^{(k)})>f(X_i^{(k)})$ для всіх $i\neq h$ то вектор $X_h^{(k)}-X_{n+1}^{(k)}$ стискається у відповідності з формулою

$$X_{n+5}^{k}=X_{n+1}^{k}+\beta\ X_{h}^{k}-X_{n+2}^{k} \tag{2.17}$$

де $0<\beta<1$ – коефіцієнт стиску. Потім $X_h^{(k)}$ замінюємо на $X_{n+5}^{(k)}$ і повертаємося до операції 1 для продовження пошуку (k + 1)-му кроці.

4. Редукція.

Якщо $f(X_{n+3}^{(k)})<f(X_h^{(k)})$ то всі вектори $X_i^{(k)}-X_l^{(k)}$ $i=1,...,n+1$, зменшуються в 2 рази з відліком від $X_i^{(k)}$ у відповідності з формулою

$$X_{i}^{k}=X_{l}^{k}+0.5\ X_{i}^{k}-X_{l}^{k}\ ,\quad i=1,\ldots,n+1 \tag{2.18}$$

Потім повертаються до операції 1 для продовження пошуку на (n+1)-му кроці.

Критерій закінчення пошуку, використаний Нелдером і Мідом , складається з перевірки умови

$$\frac{1}{n+1}\left\{\sum_{i=1}^{n+1}\left[f\;X_i^{\;k}\;-f\;X_{n+2}^{\;k}\;\right]\right\}\le\varepsilon \tag{2.19}$$

де ε – довільне мале число; $f(X_{n+2}^{(k)})$ – значення цільової функції в центрі тяжіння $X_{n+2}^{(k)}$.

Деформований багатогранник в протилежність жорсткому симплексу адаптується до топографії цільової функції, витягуючись вздовж довгих нахилених площин, міняючи напрямок в зігнутих впадинах і стискаючись в околиці мінімуму. Коефіцієнт відображення α використовується для проектування вершин з найбільшим значенням $f(x)$ через центр тяжіння деформованого багатокутника. Коефіцієнт γ вводиться для розтягу вектору пошуку у випадку, коли відображення дає вершину із значенням $f(x)$, меншим ніж найменше значення $f(x)$, отримане до відображення. Коефіцієнт стиску β використовується для зменшення вектора, якщо операція відображення не привела до вершини із значенням $f(x)$, меншим ніж друге по величині (після найбільшого) значення $f(x)$, отримане до відображення. Таким чином, за допомогою операції розтягу або стиску розміри і форма деформованого багатокутника масштабується так, щоб вони задовольняли топологію розв’язаної задачі.

При оптимізації без обмежень Нелдер і Мід рекомендували в якості задовільних значень значення параметрів α=1 , β=0.5 та γ=2.

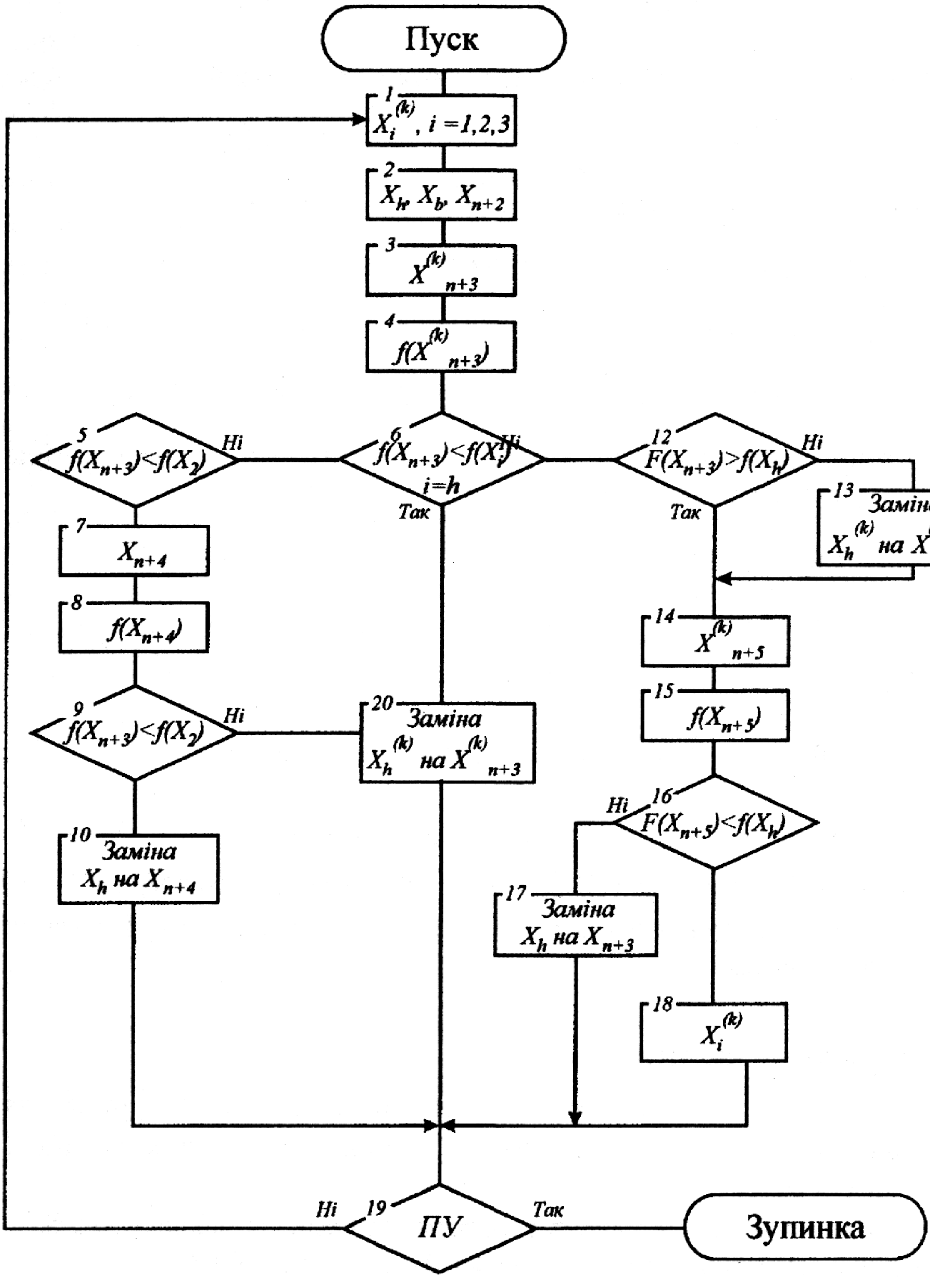


Рис. 2.5 Схема пошуку мінімуму методом Нелдера і Міда

Завдання.

1. Методом Нелдера і Міда знайти оптимум для функції:

а) $f(x) = 3x_1^2 + 5x_2^2$ починаючи з базисної точки $X^{(0)} = \begin{bmatrix} 2 & 1 \end{bmatrix}^T$

б) $f(x) = 4x_1^2 + x_2^2 - 40x_1 - 12x_2 + 136,\quad X^{(0)} = [4 \quad 8]^T$

в) $f(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4,\ X^{(0)} = [3 \quad -10 \quad 1]^T$

г) $f(x) = (x_1x_2)^2 (1 - x_1)^2 \left[1 - x_1 - x_2(1 - x_1)^5\right]^2,\quad X^{(0)} = [-1{,}2 \quad 1]^T$

2. Шляхом підбору значень параметрів β і γ в алгоритмі Нелдера і Міда знайти їх оптимальні значення для функції.

а) $f(X) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2,\quad X^{(0)} = [1 \quad 1]^T$

б) $f(X) = (x_1^2 + 12x_2 + 1)^2 + (49x_1^2 + 49x_2^2 + 48x_1 + 2324x_2 - 681)^2,\quad X^{(0)} = [1 \quad 1]^T$

в) $f(X) = 100\left[x_{33} - \left(\frac{x_1 + x_2}{2}\right)^2\right]^2 + (1 - x_1)^2 + (1 - x_2)^2,\quad X^{(0)} = [-1{,}2 \quad 2 \quad 0]^T$

3. Знайти мінімум функції Розенброка $f(X) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$, починаючи з базової точки $X^{(0)} = [-1{,}2\ \ 0]^T$ методом Нелдера і Міда. Побудувати траєкторію і вказати значення цільової функції $f(x)$ в залежності від числа ітерацій.

4. Знайти координати точок мінімуму функції Хіммельблау $f(X) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_1^2 - 7)$ з точністю до трьох десяткових знаків , при пошуку скористатися наступними початковими точками:

$X^{(1)} = [5\ 5]^T, X^{(2)} = [5\ -5]^T, X^{(3)} = [0\ 0]^T, X^{(4)} = [-5\ -5]^T, X^{(5)} = [5\ 0]^T,$

Завдання для самостійної роботи

Знайти максимум функції $f(X) = 4x_1 + 6x_2 - 2x_1^2 - 2x_1x_2 - 2x_2^2$ методом Нелдера і Міда переробивши програму пошуку для випадку максимізації цільової функції.

Контрольні запитання

Що буде вихідним симплексом у випадку пошуку Нелдера і Міда для функції $f(X) = 2x_1^2 + x_2^4$? Яка із вершин виключається першою? другою?

Як впливають на збіг алгоритму значення коефіцієнтів α, β, γ:

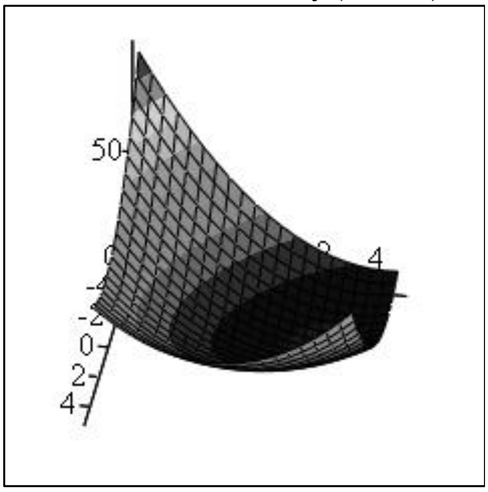
а) якщо α > 1 або α < 1;

б) якщо 0 < β < 0,4 або β > 0,6 ?

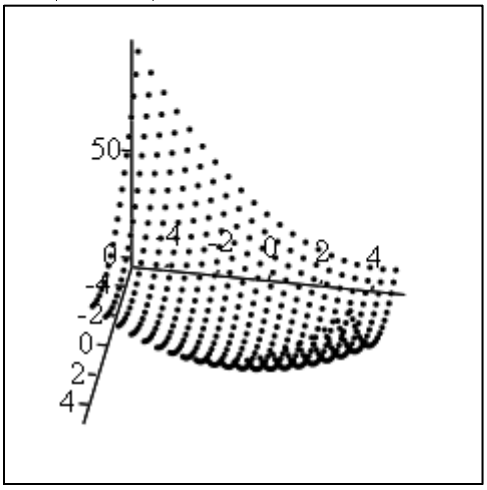
Метод Гаусса-Зейделя

Введіть $y(x_1,x_2)$ – функцію:

$$y(x1, x2) := x1^2 + x1 \cdot x2 + (x2 - 2)^2$$



y



y

Введіть *a*, *b* - ліву та праву межі зони пошуку:

$$a := -5 b := 5$$

Введіть точність обчислень *E*:

$$E := 0.0001$$

Введіть крок сканування *h*:

$$h := 0.1$$

Кількість змінних:

$$n := 2$$

$$
\mathrm{min\_gaus}(a,b,h,y) := \left|
\begin{array}{l}
x1 \leftarrow a \\
x2 \leftarrow a \\
M \leftarrow y(x1,x2) \\
x \leftarrow \begin{pmatrix} x1 \\ x2 \end{pmatrix} \\
i \leftarrow 1 \\
\text{while } x1 < b \\
\quad \left| \begin{array}{l}
x1 \leftarrow x1 + h \\
\text{if } M > y(x1,x2) \\
\quad \left| \begin{array}{l}
M \leftarrow y(x1,x2) \\
x_0 \leftarrow x1 \\
i \leftarrow i + 1
\end{array} \right. \\
\text{break otherwise}
\end{array} \right. \\
\text{while } x2 < b \\
\quad \left| \begin{array}{l}
x2 \leftarrow x2 + h \\
\text{if } M > y(x1,x2) \\
\quad \left| \begin{array}{l}
M \leftarrow y(x1,x2) \\
x_1 \leftarrow x2 \\
i \leftarrow i + 1
\end{array} \right. \\
\text{break otherwise}
\end{array} \right. \\
\begin{pmatrix} M \\ x \\ i \end{pmatrix}
\end{array} \right.
$$

$$\mathrm{Func} := \mathrm{min\_gaus}(a,b,h,y)_0$$

$$\mathrm{Arg} := \mathrm{min\_gaus}(a,b,h,y)_1$$

$$\mathrm{Iter} := \mathrm{min\_gaus}(a,b,h,y)_2$$

Розв'язок за допомогою методу Гаусса-Зейделя:

Мінімум функції на заданому інтервалі знаходиться у точці:

$$\mathrm{Arg} = \begin{pmatrix} 2.5 \\ 0.7 \end{pmatrix}$$

Значення мінімуму функції:

$$\mathrm{Func} = 10.27$$

Кількість ітерацій:

$$\mathrm{Iter} = 133$$

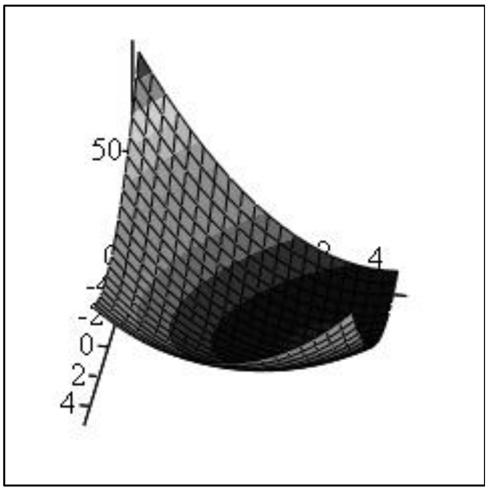
Метод Нелдера-Міда (пошук за деформованим багатогранником)

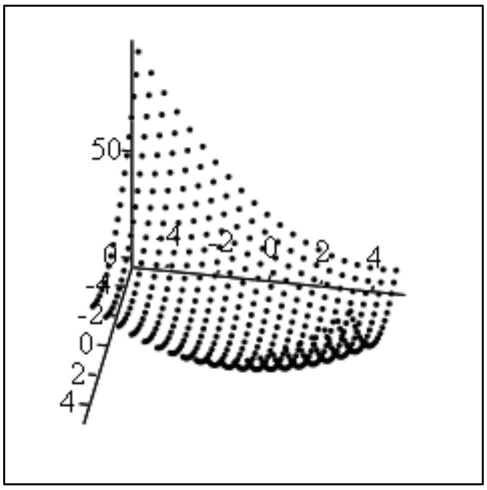
Введіть $F(x_1,x_2)$ – функцію:

$$F(x1,x2) := x1^2 + x1 \cdot x2 + (x2 - 2)^2$$

Кількість змінних:

$$n := 2$$





F F

$$\mathrm{Func}(x) := F(x_1, x_2)$$

Введіть значення початкової точки, коефіцієнт відображення, крок, точність обчислень, коефіцієнт розтягу, коефіцієнт стискання:

коефіцієнт відображення

$$\alpha := 1$$

крок

$$\lambda := 2$$

початкова точка

$$x := \begin{pmatrix} 2 \\ 3 \end{pmatrix}$$

точність обчислень

$$\varepsilon := 0.001$$

коефіцієнт розтягу

$$\gamma := 2$$

коефіцієнт стискання

$$\beta := 0.5$$

```
Nelder_Mid(x, α, λ, γ, β, ε, n) := | t ← 0
                                   | for i ∈ 1 .. n + 4
                                   |   XX_i ← x
                                   | x1 ← x
                                   | for i ∈ 1 .. n
                                   |   | for j ∈ 1 .. n
                                   |   |   | x1_j ← x_j + λ · 1  if i = j
                                   |   |   | x1_j ← x_j  otherwise
                                   |   | XX_{i+1} ← x1
                                   | σ ← 10 ε
                                   | while σ > ε
                                   |   | t ← t + 1
                                   |   | for i ∈ 1 .. n + 1
                                   |   |   F_i ← Func(XX_i)
                                   |   | Max ← (Min ← F_1)
                                   |   | h ← [g ← (el ← 1)]
                                   |   | for i ∈ 1 .. n + 1
                                   |   |   | if F_i > Max
                                   |   |   |   | h ← i
                                   |   |   |   | Max ← F_i
```

```
        otherwise
          if  F_i < Min
              el ← i
              Min ← F_i
          g ← i   otherwise
    for  i ∈ 1 .. n + 1
      g ← i  if  F_i > F_g  if  (i ≠ h) ∧ (i ≠ el)
    for  i ∈ 1 .. n
      x1_i ← 0
    for  i ∈ 1 .. n + 1
        x ← XX_i
        x ← x · 0   if  i = h
        for  j ∈ 1 .. n
          x1_j ← x1_j + x_j / n
    XX_(n+2) ← x1
    F_(n+2) ← Func(XX_(n+2))
    XX_(n+3) ← XX_(n+2) + α · (XX_(n+2) − XX_h)
    F_(n+3) ← Func(XX_(n+3))
    if  F_(n+3) < F_el
        XX_(n+4) ← XX_(n+2) + γ · (XX_(n+3) − XX_(n+2))
        F_(n+4) ← Func(XX_(n+4))
        XX_h ← XX_(n+4)   if  F_(n+4) < F_el
        XX_h ← XX_(n+3)   otherwise
    otherwise
        if  F_(n+3) > F_g
            XX_h ← XX_(n+3)   if  F_(n+3) ≤ F_h
            XX_(n+5) ← XX_(n+2) + β · (XX_h − XX_(n+2))
            F_(n+5) ← Func(XX_(n+5))
```

$$\left|\begin{array}{l} \left|\begin{array}{l} \left|\begin{array}{l} \left|\begin{array}{l} \text{for } i \in 1..n+1 \\ \quad \left|\begin{array}{l} XX_i \leftarrow \dfrac{XX_i + XX_{el}}{2} \\ F_i \leftarrow Func(XX_i) \end{array}\right. \end{array}\right. \quad \text{if } F_{n+5} > F_h \\ XX_h \leftarrow XX_{n+5} \quad \text{otherwise} \end{array}\right. \\ XX_h \leftarrow XX_{n+3} \quad \text{otherwise} \end{array}\right. \\ SumF \leftarrow \dfrac{\sum_{i=1}^{n+1} F_i}{n+1} \\ \sigma \leftarrow \sqrt{\dfrac{\sum_{i=1}^{n+1} (F_i - SumF)^2}{n+1}} \end{array}\right. \\ Fmin \leftarrow F_1 \\ k \leftarrow 1 \\ \text{for } i \in 1..n+1 \\ \quad k \leftarrow i \ \text{ if } F_i = Fmin \\ \begin{pmatrix} Fmin \\ XX_k \\ t \end{pmatrix} \end{array}\right.$$

$$Rez := Nelder\_Mid(x, \alpha, \lambda, \gamma, \beta, \varepsilon, n)$$

$$Funct := Rez_1$$

$$Arg := Rez_2$$

$$Iter := Rez_3$$

Розв'язок за допомогою методу Нелдера-Міда:

Мінімум функції на заданому інтервалі знаходиться у точці:

$$Arg = \begin{pmatrix} -1.337 \\ 2.706 \end{pmatrix}$$

Значення мінімуму функції:

$$Funct = -1.312$$

Кількість ітерацій:

$$Iter = 16$$

Оптимальний розв'язок:

$$M := Minimize(Func, x)$$

$$M = \begin{pmatrix} -1.333 \\ 2.667 \end{pmatrix}$$

$$Fmin := Func(M)\ Fmin = -1.333$$

Метод сканування

Введіть $y(x_1,x_2)$ - функцію:

$$y(x1, x2) := x1^2 + x1 \cdot x2 + (x2 - 2)^2$$

Введіть a, b – ліву та праву межі зони пошуку:

$$a := -5\ b := 5$$

Введіть точність обчислень E:

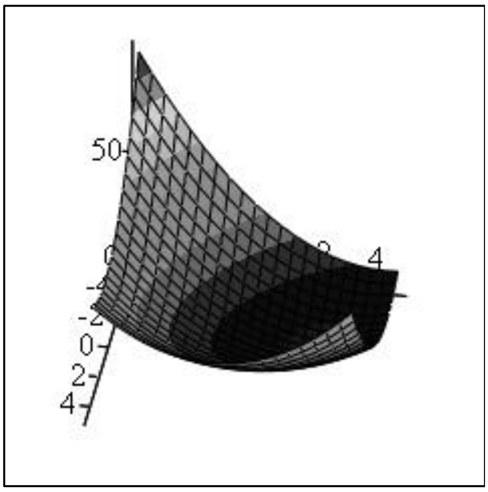
$$E := 0.01$$

Введіть крок сканування h:

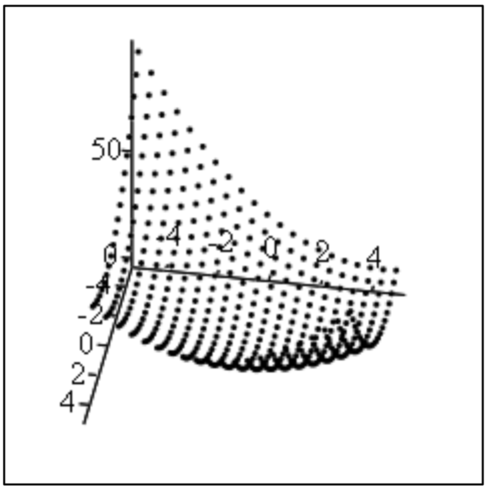
$$h := 0.1$$

Кількість змінних:

$$n := 2$$



y



y

$$
\mathrm{min\_scan}(a,b,h,y) := \left|
\begin{array}{l}
x1 \leftarrow a \\
x2 \leftarrow a \\
M \leftarrow y(x1,x2) \\
x \leftarrow \begin{pmatrix} x1 \\ x2 \end{pmatrix} \\
\text{while } x1 \le b \\
\quad \left| \begin{array}{l}
\text{while } x2 \le b \\
\quad \left| \begin{array}{l}
\text{if } M > y(x1,x2) \\
\quad \left| \begin{array}{l}
M \leftarrow y(x1,x2) \\
x \leftarrow \begin{pmatrix} x1 \\ x2 \end{pmatrix}
\end{array} \right. \\
x2 \leftarrow x2 + h
\end{array} \right. \\
x2 \leftarrow a \\
x1 \leftarrow x1 + h
\end{array} \right. \\
\begin{pmatrix} M \\ x \end{pmatrix}
\end{array} \right.
$$

$$\mathrm{Func} := \mathrm{min\_scan}(a,b,h,y)_0$$

$$\mathrm{Arg} := \mathrm{min\_scan}(a,b,h,y)_1$$

$$\mathrm{Iter} := \frac{(b-a)^3}{h^3}$$

Розв'язок за допомогою методу сканування:

Мінімум функції на заданому інтервалі знаходиться у точці:

$$\mathrm{Arg} = \begin{pmatrix} -1.4 \\ 2.7 \end{pmatrix}$$

Значення мінімуму функції:

$$\mathrm{Func} = -1.33$$

Кількість ітерацій:

$$\mathrm{Iter} = 1 \times 10^6$$

Метод сканування з постійним кроком

Введіть *F*(*x*) - функцію:

$$F(x) := (x-2)^2$$

Введіть *a*, *b* – ліву та праву межі зони пошуку:

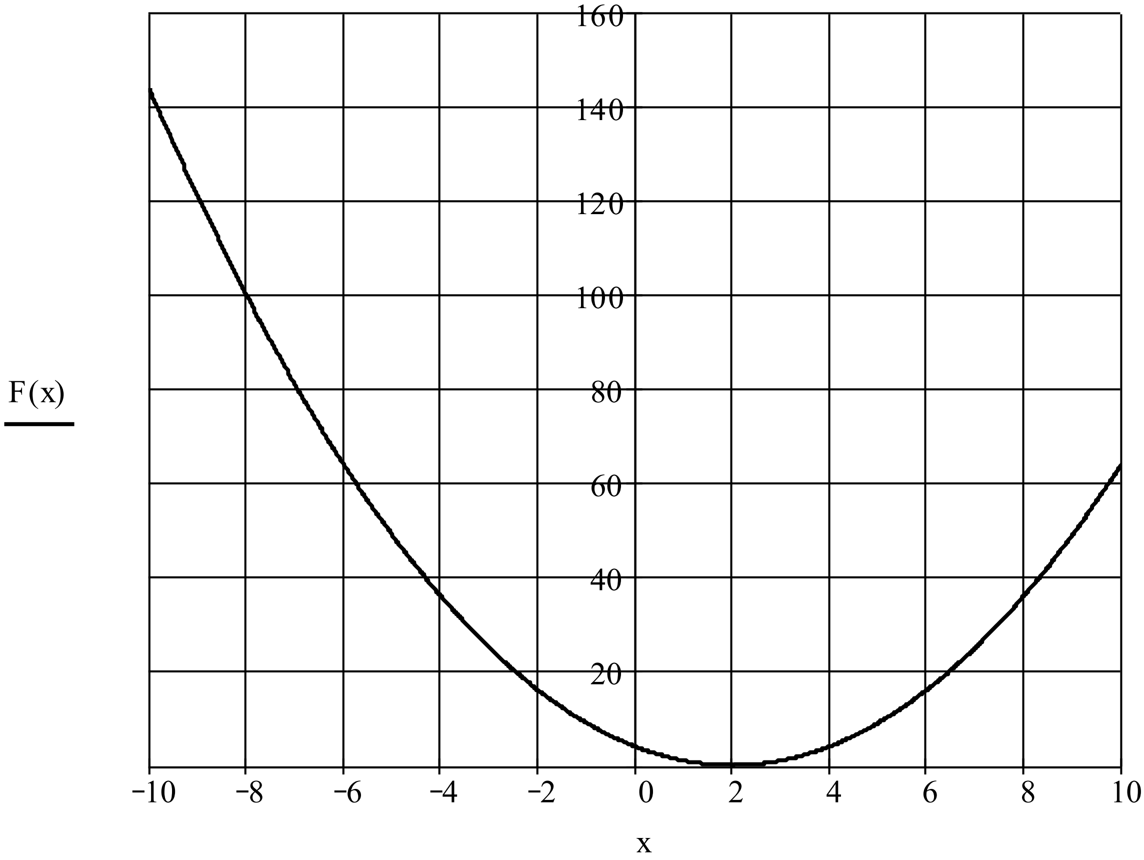
$$a := -10 \quad b := 10$$

Введіть точність обчислень *E*:

$$E := 0.01$$

Введіть крок сканування *h*:

$$h := 0.2$$



$$\mathrm{ScanConstStep}(a,b,E,h) := \left|\begin{array}{l} x1 \leftarrow a \\ F1 \leftarrow F(x1) \\ x \leftarrow a \\ \text{for } i \in 1 .. \dfrac{(b-a)}{h} \\ \quad \left|\begin{array}{l} x \leftarrow x + h \\ F \leftarrow F(x) \\ \text{if } F < F1 \\ \quad \left|\begin{array}{l} F1 \leftarrow F \\ x1 \leftarrow x \end{array}\right. \end{array}\right. \\ \begin{pmatrix} x1 \\ F1 \end{pmatrix} \end{array}\right.$$

$\mathrm{ORIGIN} := 1$

$\mathrm{Arg} := \mathrm{ScanConstStep}(a,b,E,h)_1$

$\mathrm{Func} := \mathrm{ScanConstStep}(a,b,E,h)_2$

$\mathrm{ScanConstStep}(a,b,E,h) = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$

$\mathrm{Iter} := \dfrac{(b-a)}{h}$

Розв'язок за допомогою методу сканування:

Мінімум функції на заданому інтервалі знаходиться у точці:

$$\mathrm{Arg} = 2$$

Значення мінімуму функції:

$$\mathrm{Func} = 0$$

Кількість ітерацій:

$$\mathrm{Iter} = 100$$

Оптимальний розв'язок:

$$\mathrm{Iksmin} := \mathrm{Minimize}(F, \mathrm{Arg})$$

$$\mathrm{Iksmin} = 2$$

$$\mathrm{Fmin} := F(\mathrm{Iksmin})$$

$$\mathrm{Fmin} = 0$$

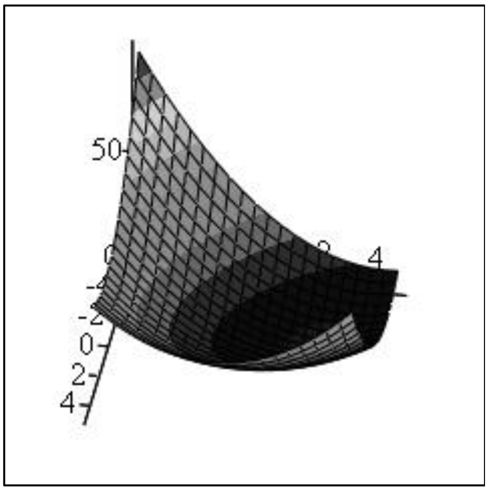
Симплексний метод

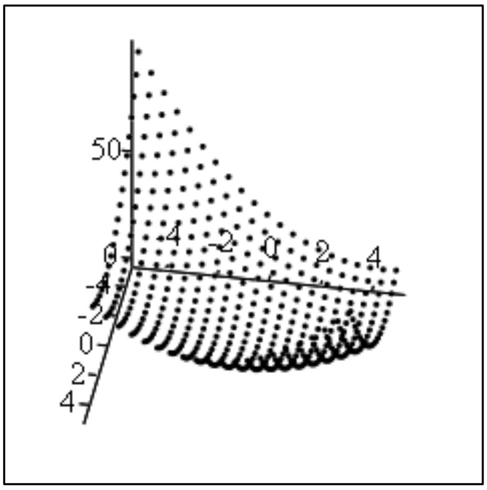
Введіть функцію F(x1,x2):

$$F(x1,x2) := x1^2 + x1 \cdot x2 + (x2-2)^2$$

Кількість змінних:

$$n := 2$$





F F

Введіть значення початкової точки, масштабний множник, крок та точність обчислень:

$$Func(x) := F\left(x_1, x_2\right)$$

початкова точка:

$$x := \begin{pmatrix} 2 \\ 3 \end{pmatrix}$$

масштабний множник:

$$\alpha := 1$$

крок:

$$\lambda := 2$$

точність обчислень:

$$\varepsilon := 0.001$$

```
Simpl(x, α, λ, ε, n) := | t ← 0
                        | x1 ← x
                        | for i ∈ 1 .. n + 1
                        |   XX_i ← x
                        | δ1 ← ((√(n + 1) + n − 1) / (n · √2)) · α
                        | δ2 ← ((√(n + 1) − 1) / (n · √2)) · α
                        | for i ∈ 1 .. n
                        |   | for j ∈ 1 .. n
                        |   |   | x1_j ← x_j + δ2  if i = j
                        |   |   | x1_j ← x_j + δ1  otherwise
                        |   | XX_(i+1) ← x1
                        | for i ∈ 1 .. n + 1
                        |   F_i ← Func(XX_i)
                        | σ ← 10 ε
                        | FM ← 2 · max(F)
                        | k ← 1
                        | s ← 0
                        | Imin ← 0
                        | while σ > ε
                        |   | t ← t + 1
                        |   | M ← max(F)
                        |   | if M = FM
                        |   |   | m ← 1  if k = n + 1
                        |   |   | m ← k + 1  otherwise
```

$$
\begin{array}{l}
\quad \left| \begin{array}{l} M \leftarrow F_m \\ \text{for } \ i \in 1 .. n+1 \\ \quad M \leftarrow F_i \ \text{ if } \ F_i > M \ \text{ if } \ i \neq k \end{array} \right. \\
\text{for } \ i \in 1 .. n+1 \\
\quad k \leftarrow i \ \text{ if } \ M = F_i \\
\text{for } \ i \in 1 .. n \\
\quad Xc_i \leftarrow 0 \\
\text{for } \ i \in 1 .. n+1 \\
\quad \text{if } \ i \neq k \\
\qquad \left| \begin{array}{l} x \leftarrow XX_i \\ \text{for } \ j \in 1 .. n \\ \quad Xc_j \leftarrow Xc_j + \dfrac{x_j}{n} \end{array} \right. \\
x1 \leftarrow XX_k \\
x \leftarrow x1 + \lambda \cdot (Xc - x1) \\
F_k \leftarrow \mathrm{Func}(x) \\
FM \leftarrow F_k \\
XX_k \leftarrow x \\
SumF \leftarrow \dfrac{\displaystyle\sum_{i=1}^{n+1} F_i}{n+1} \\
\sigma \leftarrow \sqrt{\dfrac{\displaystyle\sum_{i=1}^{n+1} \left(F_i - SumF\right)^2}{n+1}} \\
s \leftarrow s + k \\
\text{if } \ \mathrm{mod}[t,(n+1)] = 0
\end{array}
$$

$$\text{if } s \neq \sum_{j=1}^{n+1} j$$

$$\alpha \leftarrow \frac{\alpha}{2}$$

$$\delta 1 \leftarrow \left(\frac{\sqrt{n+1}+n-1}{n\cdot\sqrt{2}}\right)\cdot\alpha$$

$$\delta 2 \leftarrow \left(\frac{\sqrt{n+1}-1}{n\cdot\sqrt{2}}\right)\cdot\alpha$$

$Fmin \leftarrow \min(F)$

for $i \in 1..n+1$

  $Imin \leftarrow i$ if $F_i = Fmin$

$x \leftarrow XX_{Imin}$

$XX_1 \leftarrow XX_{Imin}$

for $i \in 1..n$

  for $j \in 1..n$

    $x1_j \leftarrow x_j + \delta 2$ if $i = j$

    $x1_j \leftarrow x_j + \delta 1$ otherwise

  $XX_{i+1} \leftarrow x1$

for $i \in 1..n+1$

  $F_i \leftarrow Func(XX_i)$

$s \leftarrow 0$

$Fmin \leftarrow \min(F)$

for $i \in 1..n+1$

  $k \leftarrow i$ if $F_i = Fmin$

$$\begin{pmatrix} Fmin \\ XX_k \\ t \end{pmatrix}$$

$Rez := Simpl(x, \alpha, \lambda, \varepsilon, n)$ $Funct := Rez_1$ $Arg := Rez_2$ $Iter := Rez_3$

Розв'язок за допомогою симплексного методу:
Мінімум функції на заданому інтервалі знаходиться у точці:

$$\mathrm{Arg} = \begin{pmatrix} -1.34 \\ 2.665 \end{pmatrix}$$

Значення мінімуму функції:

$$\mathrm{Funct} = -1.333$$

Кількість ітерацій:

$$\mathrm{Iter} = 31$$

Оптимальний розв'язок:

$$M := \mathrm{Minimize}(\mathrm{Func}, x)$$

$$M = \begin{pmatrix} -1.333 \\ 2.667 \end{pmatrix}$$

$$\mathrm{Fmin} := \mathrm{Func}(M)$$

$$\mathrm{Fmin} = -1.333$$

Лабораторна робота № 3

Методи нелінійного програмування, що використовують похідні

Градієнтні методи

В лабораторній роботі №2 розглядались методи , які дозволяють отримати рішення задачі на основі використання тільки значення цільової функції. Важливість прямих методів безперечна, оскільки в ряді практичних інженерних задач інформація про значення цільової функції являється єдиною надійною інформацією, яку має дослідник. З іншого боку, при використанні навіть найефективніших прямих методів для отримання рішення інколи потрібна надзвичайно велика кількість обчислень значень функції. Ця обставина призводить до необхідності розглядати методи, які основуються на використанні градієнту цільової функції, які ґрунтуються на ітераційній процедурі, яка реалізується в відповідності з формулою

$$X^{k+1} = X^{k} + \Delta X^{k} = X^{k} + \lambda^{k} S^{k},$$

де $X^{(k)}$ – поточне наближення до розв'язку X' $\Delta X^{(k)}$ – вектор переходу з точки $X^{(k)}$ в точку $X^{(k+1)}$, $\lambda^{(k)}$ – параметр, який характеризує довжину кроку; $S^{(k)}$ – напрямок пошуку в N – мірному просторі керованих змінних X.

Спосіб визначення значень S і λ на кожній ітерації пов'язаний з особливостями методу, який застосовується.

Звичайно значення $\lambda^{(k)}$ вибирається шляхом розв'язку задачі мінімізації $f(X)$ в напрямку $S^{(k)}$, тому при реалізації вивчених методів необхідно використовувати ефективні алгоритми одномірної мінімізації.

Метод релаксації

Алгоритм методу полягає в знаходженні осьового напрямку, уздовж якого функція цілі зменшується найбільш суттєво. Для цього в початковій точці пошуку визначаються похідні функції, що оптимізуємо за всіма незалежними змінними. Осьовому напрямку з найшвидшим убуванням цільової функції , очевидно, відповідає найбільша за модулем похідна. Якщо знак похідної від'ємний, то цільова функція убуватиме в напрямку осі, якщо – додатній, то в зворотному напрямку. В напрямі убування функції проводять кроки до тих пір, поки не буде отримано мінімальне значення за обраним напрямком в відповідності з формулою

$$X^{k+1} = X^{k} - \lambda^{k}\ \mathrm{sgn}\frac{\partial f \ \ X^{p}}{\partial X}, \qquad (3.1)$$

де $X^{(p)}$ – вектор точки , який останній раз проводились обчислення похідних цільової функції.

Тоді знов визначаються похідні по всім напрямкам за виключенням тієї, по якій здійснюється спуск і знов знаходиться осьовий напрямок найшвидшого убування функції цілі, проводяться подальші кроки. Графік руху від початкового стану до оптимуму показаний на рис. 3.1.

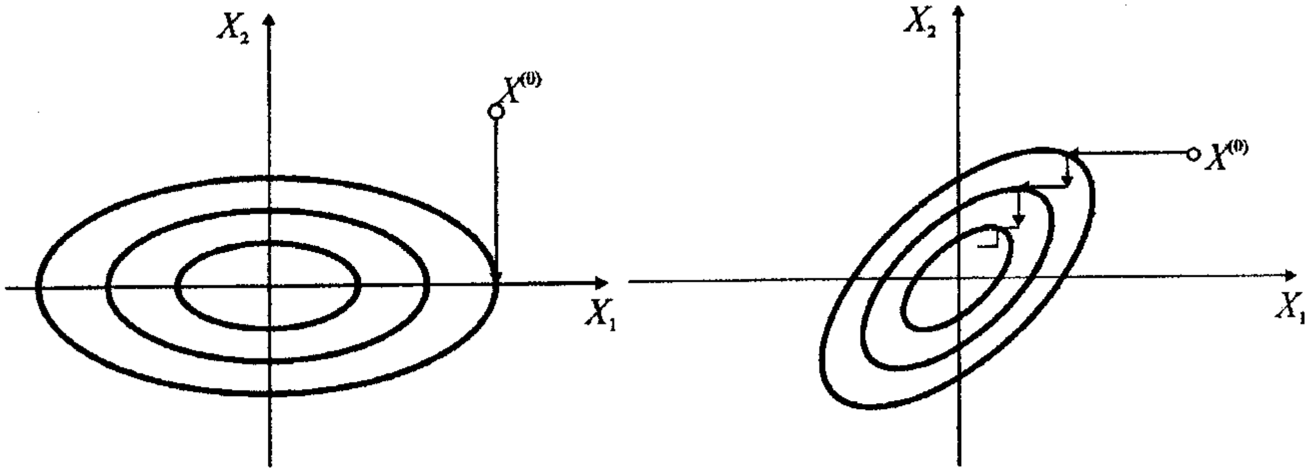


Рис. 3.1. Залежність часу пошуку оптимуму методом релаксації від орієнтації системи координат.

Суттєвою особливістю методу релаксації є залежність часу пошуку від орієнтації системи координат. Якщо в випадку (а) для досягнення мінімуму цільової функції необхідно тільки два циклу руху вздовж осьового напрямку, то для випадку (б) число таких циклів суттєво збільшується. Якщо у цільової функції є яри, напрямок яких не співпадає з осьовими, тоді пошук призупиняється в будь-якій точці дна яру.

Швидкість руху до мінімуму залежить від того, на скільки вдало підібрано крок $\lambda(k)$ змін незалежних змінних, який повинен зменшуватись при наближенні до точки мінімуму. Коли величина кроку $\lambda(k)$ стане меншою від заданої точності визначення мінімуму ε в осьовому напрямку, пошук мінімуму можна закінчити.

Пошукова процедура методу релаксації, яка реалізується в програмі, наступна.

Крок 1. Задати $X^{(0)}$ – початкове наближення до точки мінімуму X^{*}, H – початкове значення кроку, ε – параметр збіжності.

Крок 2. Визначити $f(X^{(0)})$.

Крок 3. Здійснити дослідницький пошук, знайти напрямок, в якому функція цілі найбільше зменшується.

Крок 4. Перевірка знаходження осьового напрямку. Так – перейти до кроку 5. Ні – перейти до кроку 7.

Крок 5. Провести пошук в осьовому напрямку.

Крок 6. Перейти до кроку 3, використовуючи якості базової точки, знайдену на кроці 5.

Крок 7. Перевірка на закінчення пошуку. Чи виконується нерівність $H < \varepsilon$? Так – закінчити пошук. Ні – зменшити розмір кроку по формулі $H = H/K$. Перейти до кроку 3.

Завдання.

1. Мінімізувати наступні функції двох незалежних змінних методом релаксації:

а) $f(X) = \frac{1}{2}\left\{1-\cos 360\left[(2x_1-1)^2+(2x_2-1)^2\right]^{1/2}\right\}\left[1-\frac{(x_2-3x_1)^2}{8}\right]$;

б) $f(X) = x_2 + \sin x_1$;

в) $f(X) = -(x_2-x_1)^4 + (1-x_1)^2$;

г) $f(X) = x_1^2 + x_1 - x_2^2 + x_2 - 4$;

д) $f(X) = \exp\left[-(x_1-1)^2\right] - \frac{(x_2^2-0.5)^2}{0.132}$.

2. Мінімізувати наступні функції методом релаксації:

а) $f(X) = 100(x_2-x_1^2)^2 + (1-x_1)^2$

б) $f(X) = 4(x_1-5)^2 + (x_2-6)^2$

в) $f(X) = 8x_1^2 + 4x_1x_2 + 5x_2^2$;

г) $f(X) = 2x_1^3 + 4x_1x_2^3 - 10x_1x_2 + x_2^2$.

Завдання для самостійної роботи.

В задачі, зв'язаній з проблемою прийняття рішення, бажано мінімізувати очікуваний ризик, який визначається наступним чином:

$$\varepsilon(\text{ризик}) = (1-p)c_1\left[1-F(b)\right] + pc_1^{\theta}\left[\frac{b}{2}+\frac{2\pi}{4}\right]F\left(\frac{b}{2}-\frac{\sqrt{2\pi}}{4}\right),$$

де $F(b) = \int_{-\infty}^{6} e^{-u^2/2\Theta^2}du$; $c_1 = 1.25\cdot 10^5$; $c_2 = 15$; $\Theta = 2000$; $p = 0.25$.

Знайти мінімальний очікуваний ризик і значення b.

Контрольні запитання:

У чому суть методу релаксації?

Чому квадратичні функції використовуються як основа для побудови алгоритмів нелінійного програмування?

Метод градієнта

На відміну від методу релаксації в простому градієнтному методі кроки здійснюються у напрямку найшвидшого зменшення цільової функції, що прискорює процес пошуку.

Градієнт цільової функції $f(X)$ в будь-якій точці X є вектор у напрямку найбільшого локального збільшення $f(X)$. Отже потрібно рухатись в напрямку, протилежному градієнту $f(X)$, оскільки від'ємний градієнт $f(X)$ в точці $X(k)$ направлений в бік найбільшого зменшення $f(X)$ по всім компонентам X і ортогональний лінії рівня $f(X)$ в точці $X(k)$. Напрям, протилежний нормованому (одиничному) градієнту $f(X)$, який визначається в точці $X(k)$ за формулою

$$S^{k} = \frac{\nabla f\left(X^{k}\right)}{\left\|\nabla f\left(X^{k}\right)\right\|}, \tag{3.2}$$

де $\nabla f(X(k))$ – вектор-стовпець градієнту,

$$\nabla f\left(X^{k}\right) = \begin{pmatrix} \dfrac{\partial f(X^{k})}{\partial x_1} \\ \vdots \\ \dfrac{\partial f\left(X^{k}\right)}{\partial X_n} \end{pmatrix}; \tag{3.3}$$

$\left\|\nabla f\left(X^{k}\right)\right\|$ – довжина вектору,

$$\left\|\nabla f\left(X^{k}\right)\right\| = \sqrt{\left(\frac{\partial f\left(X^{k}\right)}{\partial X_1}\right)^2 + \ldots + \left(\frac{\partial f\left(X^{k}\right)}{\partial X_n}\right)^2}.$$

В основі простого градієнтного методу лежить формула переходу:

$$X^{k+1} = X^{k} - \frac{\lambda^{k} \nabla f\left(X^{k}\right)}{\left\|\nabla f\left(X^{k}\right)\right\|} = X^{k} - \lambda^{*} \nabla f\left(X^{k}\right) \tag{3.4}$$

Від'ємний градієнт дає тільки напрям оптимізації, але не довжину кроку. Задача вибору стратегії зміни довжини кроку в градієнтному пошуку більш важлива, ніж в методі релаксації. Це пояснюється тим, що після кожного кроку тут визначаються похідні цільової функції, розрахунок яких зв'язаний з обчисленням значень цільової функції. Якщо, з одного боку, розмір кроку надто малий, то рух до оптимуму буде довгим через необхідність розрахунку цільової функції в багатьох точках. З іншого боку, якщо крок вибрано надто великим, в районі оптимуму можуть виникнути пошуки, які або не затихають, або затихають надто повільно. Характер пошуку оптимуму при малій і великій довжині кроку показано на рис. 3.2.

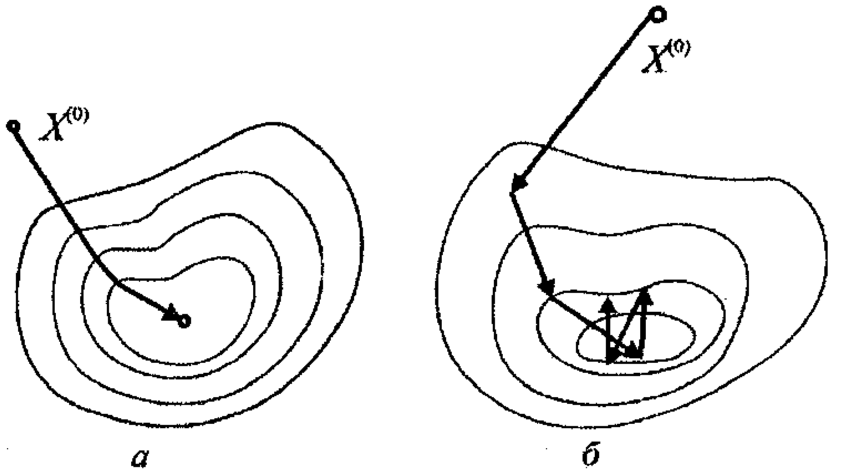


Рис. 3.2. Характер руху до оптимуму в методі градієнта з малою (а) і великою (б) довжиною кроку.

В точці мінімуму всі складові вектора градієнта дорівнюють нулю. Тип стаціонарної точки може бути перевірений шляхом дослідження матриці Гессе (якщо її можливо отримати) цільової функції, взятої в даній стаціонарній точці. Якщо ця матриця не являється додатною визначеною, тоді стаціонарна точка – сідлова.

Як критерій закінчення послідовної процедури під час руху в напрямку найшвидшого спуску застосовуються різні правила, які основуються на значенні $f(X)$ і значеннях $X, \lambda, \nabla f(X)$. Схема алгоритму закінчується, коли модуль градієнта або модуль вектора ΔX стає достатньо малим. В програмі, яка буде приведена далі, критерієм закінчення пошуку є значення λ.

Якщо нелінійно-цільова функція надто важка, щоб її можливо було диференціювати аналітично, тоді складові градієнту, які є поодинокими похідними за оптимізованими змінними, апроксимуються співвідношенням різниць:

$$\frac{\partial f}{\partial x_i} = \frac{f\left(x_1, x_2, \ldots, x_{i+\Delta}, \ldots, x_N\right) - f\left(x_1, x_2, \ldots, x_i, \ldots, x_N\right)}{\Delta}, \quad i = 1, \ldots, N,$$

де Δ – невелике зміщення напрямку X_i. Цю формулу часто називають наближенням січної. На деякій відстані від мінімуму апроксимація різниці похідних цілком прийнятна, однак поблизу мінімуму вона недостатньо задовільна. Наведемо алгоритми програми пошуку мінімуму методом градієнту.

Крок 1. Задати $X^{(0)}$ – початкове наближення до точки мінімуму X^*; H – початкове значення кроку, ε – параметр збіжності.

Крок 2. Визначити $f(X^{(0)})$.

Крок 3. Визначити компоненти $\nabla f(X^{(k)})$.

Крок 4. Покласти $X^{(k+1)} = X^{(k)} - \lambda^{(k)} \nabla f\left(X^{(k)}\right)$.

Крок 5. Чи виконуються нерівності $f\left(X^{(k+1)}\right) < f\left(X^{(k)}\right)$? Так – перейти до кроку 3. Ні – перейти до кроку 6.

Крок 6. Повернутись до попередньої вдалої точки *X(k)*.

Крок 7. Перевірка на відхилення пошуку. Чи виконується нерівність $H < \varepsilon$? Так – призупинити пошук. Ні – зменшити розмір кроку за формулою $H = H/K$. Перейти до кроку 3.

Метод має недоліки: по-перше, виникає необхідність вибору підходящого значення λ; по-друге , методу притаманна повільна збіжність в точці мінімуму внаслідок невеликої Δf навколо цієї точки; по-третє, при використанні методу можна знайти тільки локальний метод цільової функції.

Завдання.

1. Мінімізувати наступних функції двох незалежних змінних методом градієнту.

а) $f(X) = \frac{1}{2}\left\{1 - \cos 360\left[(2x_1 - 1)^2 + (2x_2 - 1)^2\right]^{1/2}\right\}\left[1 - \frac{(x_2 - 3x_1)^2}{8}\right]$;

б) $f(X) = x_2 + \sin x_1$;

в) $f(X) = -(x_2 - x_1)^4 + (1 - x_1)^2$;

г) $f(X) = x_1^2 + x_1 - x_2^2 + x_2 - 4$;

д) $f(X) = \exp\left[-(x_1 - 1)^2\right] - \frac{(x_2^2 - 0.5)^2}{0.132}$;

2. Мінімізувати наступні функції методом градієнту:

а) $f(X) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$;

б) $f(X) = 4(x_1 - 5)^2 + (x_2 - 6)^2$;

в) $f(X) = 8x_1^2 + 4x_1x_2 + 5x_2^2$;

г) $f(X) = 2x_1^3 + 4x_1x_2^3 - 10x_1x_2 + x_2^2$.

Завдання для самостійної роботи: Виконати один цикл пошуку максимуму функції $f(X) = x_1 + x_2 - (x_1 x_2)^2 - x_1^2 - 2x_2^2$ початкової точки $X^{(0)}$ =[1 0]T , використовуючи алгоритм градієнту з постійним кроком $\lambda = 1$. Знайти значення косинусу кута між напрямком градієнту в початковій точці, яку було знайдено після першого кроку. Оцінити, як потрібно змінювати крок λ в другому циклі алгоритму градієнту з уточненням кроку.

Контрольні запитання:

1. В який локальний оптимум приведе пошук методом градієнту, якщо функція мультимодальна?
2. Як вплине помилка в визначенні однієї із складових градієнту на його значення і напрямок?
3. Як впливає масштабування змінних на збіжність методу градієнту і при якій формі ліній рівного рівня можна отримати швидку збіжність?

Метод найшвидшого спуску

Застосування методу градієнту потребує на кожному кроці визначення значення всіх приватних похідних оптимізованої функції по всім незалежним змінним. Якщо розрахунок одного значення даної функції потребує значного об’єму обчислень, тоді час пошуку оптимуму, особливо при великій кількості незалежних змінних, може бути довгим.

Метод релаксації має переваги, тому-що під час спускання вздовж вибраного осьового напрямку не потребує обчислення похідних. Однак рух здійснюється не в оптимальному напрямку, оскільки градієнт в загальному випадку не співпадає з осьовим напрямком.

Поєднання методів релаксації і градієнту дає метод найшвидшого спуску, який полягає в наступному. Після того як в початковій точці знайдено градієнт функції, що оптимізуємо і тим самим визначено напрям її найшвидшого убування, в даному напрямку робиться крок спуску. Якщо значення функції в результаті цього кроку зменшилось, тоді робиться наступний крок в тому ж напрямку до тих пір, поки в цьому напрямку не буде знайдено мінімум, після чого обчислюються градієнт і визначається новий напрям найшвидшого убування цільової функції.

В порівнянні з методом градієнту метод найшвидшого спуску більш зручний, бо скорочується об’єм обчислень. Крім того, він вигідно відрізняється від методу релаксації тим, що перші кроки після визначення градієнту здійснюються в оптимальному напрямку.

При визначенні розміру кроку застосовують два загальних методи. В першому методі при переході з точки $X^{(k)}$ в точку $X^{(k+1)}$ цільова функція мінімізується за кроком λ, в другому методі довжина кроку λ вибирається або змінюється від кроку до кроку. Функцію $f(x)$ можна мінімізувати за допомогою одного з методів одномірного чисельного пошуку (наприклад, метод Фібоначчі або методу квадратичної апроксимації Коггінса). Якщо в методі обирається фіксоване значення скаляра λ або ця величина змінна, то вона повинна контролюватись, щоб уникнути як різкого росту $f(x)$, так і надмірної кількості кроків. Важливою рисою методу найшвидшого спуску є те, що новий напрямок руху до оптимуму ортогональний до попереднього пошуку $S^{(k)}$. Метод найшвидшого спуску дозволяє суттєво зменшити значення цільової функції, рухаючись з точок, які розташовані на великій відстані від точки мінімуму і тому часто використовується при реалізації градієнтних методів в якості процедури. Наведемо алгоритм пошуку мінімуму методом найшвидшого спуску.

Крок 1. Задати $X^{(0)}$ – початкове приближення до точки мінімуму X^*; H – початкове значення кроку; K – кратність зменшення кроку; ε – параметр збіжності.

Крок 2. Визначити $f(X^{(0)})$.

Крок 3. Визначити компоненти $f(X^{(k)})$.

Крок 4. Покласти $X^{(k+1)} = X^{(k)} - \lambda^{*(k)} \nabla f\left(X^{(k)}\right)$.

Крок 5. Чи виконується нерівність $f(X^{(k+1)}) < f(X^{(k)})$? Так - перейти до кроку 4. Ні – перейти до кроку 6.

Крок 6. Повернути до попередньої вдалої точки $X^{(k)}$.

Крок 7. Перевіряється нерівність $f(X^{(k)}) < f(X^{(0)})$? Так – перейти до кроку 3. Ні – перейти до кроку 8.

Крок 8. Перевірка закінчення пошуку. Чи виконується нерівність $H < \varepsilon$? Так – припинити пошук. H_i – зменшити розмір кроку за формулою $H = H/K$. Перейти до кроку 3.

Завдання:

1. Мінімізувати наступні функції двох незалежних змінних методом найшвидшого спуску:

а) $f(x) = \dfrac{1}{2}\left\{1 - \cos 360\left[(2X_1 - 1)^2 + (2X_2 - 1)^2\right]^{\frac{1}{2}}\left[1 - \dfrac{(X_2 - 3X_1)^2}{8}\right]\right\}$

б) $f(x) = X_2 + \sin X_1$;

в) $f(X) = -(X_2 - X_1)^4 + (1 - X_1)^2$; $f(X) = -X_1^2 + X_1 - X_2^2 + X_2 + 4$;

г) $f(X) = \exp\left[-(X_1 - 1)^2\right] - \dfrac{(X_2^2 - 0.5)^2}{0.132}$;

2. Мінімізувати наступні функції методом найшвидшого спуску:

а) $f(X) = 100(X_2 - X_1^2)^2 + (1 - X_1)^2$;

б) $f(X) = 4(X_1 - 5)^2 + (X_2 - 6)^2$;

в) $f(X) = 8X_1^2 + 4X_1X_2 + 5X_2^2$;

г) $f(X) = 2X_1^3 + 4X_1X_2^3 - 10X_1X_2 + X_2^3$;

Порівняти швидкість збіжності методу зі швидкістю збіжності методів релаксації і градієнту.

Завдання для самостійної роботи.

Максимізувати $f(X) = e^{-X_1 - X_2}(2X_1^2 + 3X_2^2)$ методом найшвидшого спуску. В якості початкової вибирається точка $X^{(0)} = [2,5; 2,5]$.

Контрольні запитання.

1. Якщо визначено напрямок градієнтного пошуку в точці $X^{(k)}$, як треба вибрати розмір кроку, щоб досягти наступної точки $X^{(k+1)}$?
2. Що являється напрямком найшвидшого спуску в точці $X = [1; 1]$ для цільової функції $f(X) = X_1^2 + 2X_2^2$? Провести два кроки оптимізації.
3. Довести, чи являється метод найшвидшого спуску методом, який використовує спряжені напрямки при мінімізації квадратичної функції з додатно визначеною матрицею Гессе.

Метод других похідних. Метод Ньютона

Напрямок пошуку, який відповідає найшвидшому спуску, інтерпретується як наслідок лінійної апроксимації цільової функції. Для визначення напрямку пошуку необхідно обчислити значення функції і її перших похідних. Однак рух в напрямку, протилежному градієнту, де в точку мінімуму лише у випадку, коли лінії рівня функції $f(X)$ являють собою кола. Для побудови більш узагальнюючої стратегії пошуку потрібно застосувати інформацію про другі похідні цільової функції. Квадратичну апроксимацію цільової функції можна отримати, відкидаючи в ряді Тейлора члени третього і більш високих порядків.

$$f(X) = f(X^k) + \nabla^T f(X^k)\Delta X^k + \frac{1}{2}(\Delta X^k)^T \nabla^2 f(X^k)\Delta X^k \qquad (3.5)$$

де $\nabla^T f(X^k)$ – вектор-рядок градієнту; $\nabla^2 f(X^k)$ - матриця Гессе $f(X)$, $H(X)$, яка представляє собою квадратичну матрицю других частинних похідних $f(X)$, взятих в точці $X^{(k)}$:

$$\nabla^2 f\left(X^{k}\right) = H\left(X^{k}\right) = \begin{pmatrix} \dfrac{\partial^2 f(X^{(k)})}{\partial X_1^2} \cdots \dfrac{\partial^2 f\left(X^{k}\right)}{\partial X_1 \partial X_n} \\ \dfrac{\partial^2 f\left(X^{k}\right)}{\partial X_n \partial X_1} \cdots \dfrac{\partial^2 f\left(X^{k}\right)}{\partial X_n^2} \end{pmatrix}.$$

Мінімум функції $f(X)$ в напрямку $\Delta X^{(k)}$ визначається диференціюванням $f(X)$ по кожному із компонентів ΔX і прирівнюванням до нуля отриманих виразів. Останнє призводить до

$$\Delta X^{k} = -\left[\nabla^2 f\left(X^{k}\right)\right]^{-1} \nabla f\left(X^{k}\right) \tag{3.6}$$

де $\left[\nabla^2 f\left(X^{k}\right)\right]^{-1}$ – матриця, обернена до матриці Гессе $H(X^{(k)})$.

Застосування схеми квадратичної апроксимації призводить до реалізації оптимального методу Ньютону за формулою

$$X^{k+1} = X^{k} - \left[\nabla^2 f\left(X^{k}\right)\right]^{-1} \nabla f\left(X^{k}\right) \tag{3.7}$$

Якщо $f(X)$ – квадратична функція, то для досягнення мінімуму $f(X)$ достатньо тільки одного кроку. Але у випадку загальної нелінійної цільової функції мінімуму $f(X)$ неможливо досягнути за один крок, тому рівняння (3.7) шляхом введення спеціального параметру величини кроку λ приводить до виду

$$X^{k+1} = X^{k} - \lambda^{k} \frac{\left[\nabla^3 f\left(X^{k}\right)\right]^{-1}\left[\nabla f\left(X^{k}\right)\right]}{\left\|\left[\nabla^2 f\left(X^{k}\right)\right]^{-1}\left[\nabla f\left(X^{k}\right)\right]\right\|} \tag{3.8}$$

де $\left\|\left[\nabla^2 f(X^{(k)})\right]^{-1}\left[\nabla f(X^{(k)})\right]\right\|$ – довжина вектору, яка дорівнює квадратному кореню добутку квадратів складових вектору.

Співвідношення $\dfrac{\lambda^{k}}{\left\|\left[\nabla^2 f\left(X^{k}\right)\right]^{-1}\left[\nabla f\left(X^{k}\right)\right]\right\|}$ – просто деякий скаляр $\lambda^{*(k)}$, тому рівняння (3.8) записують наступним чином:

$$X^{k+1} = X^{k} - \lambda^{*k} H^{-1}\left(X^{k}\right) \nabla f\left(X^{k}\right) \tag{3.9}$$

де $S^{k} = H^{-1}\left(X^{k}\right) \nabla f\left(X^{k}\right)$ – вектор напрямку пошуків.

Критерієм, який гарантує збіжність методу Ньютону при припущенні, що функція $f(X)$ двічі диференційована, є те, що матриця, обернена матриці Гессе цільової функції, повинна бути додатно визначеною:

$$\left[\nabla^2 f\left(X^{k}\right)\right]^{-1} = H^{-1}\left(X^{k}\right) > 0 \qquad (3.10)$$

Якщо в деякій точці $\nabla^2 f\left(X^{k}\right)$ від'ємно визначена, то вказаний напрямок є напрямком підйому. Метод Ньютону має переваги, які містяться в квадратичній збіжності навколо мінімуму цільової функції (якщо $H(X)>0$ і цільову функцію можна достатньо добре апроксимувати квадратичною функцією), тобто саме там, де особливо проявляються слабкі сторони методів найшвидшого спуску. Навпаки, в області, далекій від мінімуму, методи найшвидшого спуску можуть виявитись кращими. Обернена матриця Гессе $H^{-1}(X^{(k)})$ може апроксимуватись на кожних n кроках матрицею, що проектується $R^{(k)}$, так що в сутності здійснюється (приблизно) пошук Ньютона. Приведемо основні кроки алгоритму методу проекцій.

Крок 1. Почати в точці $X^{(0)}$ і встановити матрицю, що проектується $R^{(0)}$ = 1.

Крок 2. Для K-го кроку матриця, що проектується визначається за формулою:

$$R^{k+1} = R^{k} + \frac{\left(\Delta X^{k} - R^{k}\Delta g^{k}\right)^{T}\left(\eta^{k}\Delta g^{k}\right)^{T}}{\left(\Delta g^{k}\right)^{T}\eta^{k}\left(\Delta g^{k}\right)},$$

де $\Delta g^{k} = \nabla f\left(X^{k+1}\right) - \nabla f\left(X^{k}\right)$.

Крок 3. Якщо $R^{k}\Delta f\left(X^{k}\right) \neq 0$, то встановити $S^{k} = -R^{k}\nabla f\left(X^{k}\right)$ і мінімізувати $f(X)$ в напрямку $S^{(k)}$; точка мінімуму – $X^{(k+1)}$. Повторити крок 2. Після того як пройдено n напрямку пошуку, почати знову з кроку 1 при $X^{(0)} = X^{(n)}$. Якщо $R^{k}\nabla f\left(X^{k}\right)$ і $\nabla f\left(X^{k}\right) = 0$, то закінчити пошук. Якщо $R^{k}\nabla f\left(X^{k}\right) = 0$, а $\nabla f\left(X^{k}\right) = 0$, то почати з кроку 1, встановивши $X^{(0)} = X^{(k)}$.

Завдання:

1. Мінімізувати методом Ньютона наступні функції:

а) $f\left(X\right) = \left(X_1^2 + 12X_1 - 1\right)^2 + \left(49X_1^2 + 49X_2^2 + 84X_1 + 232X_2 - 681\right)^2, \quad X^{(0)} = [1;1]^T;$

б) $f\left(X\right) = 100\left[X_3 - \left(\frac{X_1 + X_2}{2}\right)^2\right]^2 + \left(1 - X_1\right)^2 + \left(1 - X_2\right)^2, \quad X^{(0)} = [1{,}2\ 2\ 0]^T;$

в) $f(X) = \left(X_1^2 + X_2 - 11\right)^2 + \left(1 - X_1\right)^2 + \left(1 - X_2\right)^2, \quad X^{(0)} = [1.2\ 2\ 0]^T$;

г) $f(X) = \left(X_1 + 10X_2\right)^2 + 5\left(X_3 - X_4\right)^2 + \left(X_2 - 2X_3\right)^4 + 10\left(X_1 - X_4\right)^4, \quad X^{(0)} = [3\ -1\ 0\ 1]^T$;

д) $f(X) = \dfrac{\left(X_1 - X_2\right)^2}{\left(1 - X_1\right)^2}\left[1 - \left(X_1 - X_2\right)\left(1 - X_1\right)^5\right]^2, \quad X^{(0)} = [1.2\ 1]^T$.

2. На прикладах порівняти методи, які основуються на використанні похідних першого і другого порядку.

Завдання для самостійної роботи. Мінімізувати цільову функцію $f(X) = U_1^2 + U_2^2 + U_3^2$ методом Ньютона, де $U_i = C_i - X_1(1 - X_2^i)$, $C_1 = 1{,}5$; $C_2 = 2{,}25$; $C_3 = 2{,}625$; $X^{(0)} = [2\ 0{,}2]$. Як початкову вибрати точку $X^{(0)} = [2\ 0{,}2]^T$.

Контрольні запитання

1. Розглянути наступні цільові функції:

а) $f(X) = 1 + X_1 + X_2 + \dfrac{4}{X_1} + \dfrac{9}{X_2}$;

б) $f(X) = \left(X_1 + 5\right)^2 + \left(X_2 + 8\right)^2 + \left(X_3 + 7\right)^2 + 2X_1^2 X_2^2 + 4X_1^2 X_3^2$.

Чи буде метод Ньютона сходитись для цих функцій?

2. Що можна сказати про збіжність методу Ньютона, виходячи з розгляду матриці Гессе цільової функції?

3. Скільки потрібно буде кроків для досягнення мінімуму цільової функції методом Ньютона.

Метод спряженого градієнту

Метод Ньютону не має високої надійності при визначенні точки мінімуму з віддаленої точки, однак є ефективним у випадках, коли точка $X^{(k)}$ знаходиться поблизу точки мінімуму. Серед методів, які характеризуються високою надійністю при визначенні точки мінімуму X^* з віддаленої точки і, з іншого боку, що швидко наближаються навкруги мінімуму, можна виділити клас алгоритмів, в основі яких лежить побудова спряжених напрямків. В загальному випадку система n лінійно незалежних напрямків пошуку $S^{(0)}$, $S^{(1)}$, ..., $S(n-1)$ зветься спряженою по відношенню до деякої додатно визначеної (квадратичної) матриці H, якщо

$$\left(S^{(j)}\right)^T H S^{(j)} = 0, \quad i \neq j \leq n-1 \qquad (3.11)$$

Спряженість – поняття, аналогічне ортогональності; дійсно, коли $H = I$, де I – одинична матриця, то у відповідності з рівнянням (3.11) $\left(S^{(j)}\right)^T H S^{(j)} = 0$. Якщо використовуються спряжені напрямки, то будь-яка квадратична функція n змінних, що має мінімум, може бути мінімізованою за n кроків, по одному в кожному серед спряжених напрямків. В методі спряженого градієнту Флетчера-Рівса будується послідовність напрямків пошуку S, які є лінійними комбінаціями $\nabla f\left(X^{k}\right)$ – поточного напрямку найшвидшого спуску і $S^{(0)}$, ..., $S^{(k-1)}$ - попередніх напрямків пошуку, причому вагові коефіцієнти підбирають так, щоб зробити напрямки пошуку спряженими. Згадані коефіцієнти такі, що для визначення нового напрямку пошуку в точці $X^{(k)}$ використовуються тільки поточний і передостанній градієнти.

Спочатку вибирають початкову точку простору проектування і шляхом обчислення компонент вектору градієнту визначається напрямок найшвидшого спуску. Потім в напрямку найшвидшого спуску ведеться одномірний пошук. Визначивши мінімум у цьому напрямку, визначають новий напрямок, який дещо відрізняється від напрямку вектору градієнту і визначається з відношення

$$S^{k+1} = -\nabla f\left(X^{k+1}\right) + S^{k+1}\frac{\nabla^{T} f\left(X^{k+1}\right)\nabla f\left(X^{k+1}\right)}{\nabla^{T} f\left(X^{k}\right)\nabla f\left(X^{k}\right)} \qquad (3.12)$$

Після $(n + 1)$-й ітерації $(k = n)$ процедура циклічно повторюється з заміною $X^{(0)}$ на $X^{(n+1)}$.

Перевага цього алгоритму полягає в тому, що він дозволяє використовувати переваги градієнтних методів, які дозволяють при вивченні цільової функції з розривними похідними. Так як $n + 1$ направлений пошук різниться від напрямків векторів градієнту, то пошук на зламі, а йде вздовж лінії, яка поєднує точки зламів лінії рівня, яка, як правило, проходить через точку оптимуму. Крім того, тут потрібна оберненість матриць, програма методу потребує обмеженої пам'яті ЕОМ. Наведений метод застосовується і до квадратичної функції, тому що якщо пошук здійснюється поблизу мінімуму, то можна чекати на досягнення квадратичної збіжності.

Завдання.

1. Починаючи з точки $X^{(k)}=[1\ -2,\ 3]^{T}$, визначити точку $X^{(k+1)}$ методом спряженого градієнту для наступних цільових функцій:

$f(X) = X_1^2 + X_2^2 + X_3^2$;

$f(X) = 2X_1^2 + 2X_1X_2 + 3X_2^3 + X_3$;

$f(X) = \exp\left(X_1^2 + X_2^2 - X_3 - X_1 + 4\right)$.

2. Мінімізувати за допомогою методу Флетчера-Рівса наступні функцій, починаючи з вектора $X^{(0)}=[2\ -2{,}5\ 2\ -2{,}5]^{T}$:

$f(X) = X_1^2 + X_2^2 + X_3^2 + X_4^2;$

$f(X) = (X_1 - X_2)^2 + (X_3 - X_4)^2;$

$f(X) = X_1^3 + X_2 + X_3^3 + X_4 + 16X_1^2X_2 + 8X_2^2X_3 + X_3^2X_4 + 2.$

3. Використовуючи метод Флетчера-Рівса з початковою точкою $X^{(0)}=[9\ -7\ 11]^T$ мінімізувати функцію $f(X)=3(X_1 - 1)^2 + 2(X_2 - 2)^2 + (X_3 - 2)^2$.

Завдання для самостійної роботи:

1. Максимізувати наступну цільову функцію

$$f(X) = X_1^3\exp[X_2 - X_1^2 - 10\,(X_1 - X_2)^2].$$

Порівняти траєкторії оптимізації в просторі X при використанні методів:

найшвидшого спуску,

а) Ньютона,

б) Флетчера-Рівса.

2. Щомісячно втрати, які пов’язані з експлуатацією компресору на газопроводі, визначаються формулою:

$$C = \frac{KQZ}{10^6 Z}\left(\ln\frac{P_1}{P_2} + b\right) + K_1D^2\left[\frac{P_1}{2(S - P_1)} + \frac{P_1^2}{4(S + P_1)^2}\right]$$

де C – експлуатаційні витрати, грн/рік; Q – кількість газу, що накачали, м^3/с; L – відстань між попередніми станціями, м; P_1, P_2 – тиск відповідно на виході і вході, Па; D – діаметр трубопроводу, м; K_1, K_2, Z, S, b – константи. Крім того: $Q = K_2\dfrac{D^{2,5}(P_1^2 - P_2^2)^{0,54}}{L^{0,54}Z^{0,54}}$ при значеннях $Z = 1$, $K = 1370$, $L = 20$, $b = 1,476$; $K_1 = 0,081$; $K_2 = 1,13$; $S = 100$ визначити P_1 і P_2, мінімізувати C за допомогою методу Флетчера-Рівса.

Контрольні запитання.

1. Умова спряженості незалежних напрямків.
2. З’ясувати, чи можливо, щоб ортогональний напрямок був
3. Чи є напрямки $\begin{bmatrix}0\\1\end{bmatrix}$ і $\begin{bmatrix}1\\0\end{bmatrix}$ ортогональними? Спряженими?

Методи змінної метрики

Існує клас методів, які називаються методами змінної метрики, або квазіньютонівськими, які апроксимують матрицю Гессе або обернену до неї. Квазіньютонівські методи мають лише перші похідні. в більшості цих методів використовують спряжені напрямки. Новий вектор X визначається за вектором попереднього кроку за допомогою рівняння:

$$X^{(k+1)} = X^{(k)} - \lambda^{(k)} S^{(k)} = X^{(k)} - \lambda^{*(k)} \eta X^{(k)} f\left(X^{(k)}\right), \tag{3.13}$$

де матриця $\eta X^{(k)}$ являє собою апроксимацію $H^{-1}(X)$.

При використанні $H^{-1}(X)$ необхідно точно визначити другі похідні $f(X)$ і обертати матрицю $H(X)$, тоді як в методах змінної метрики для обчислення $\eta X^{(k)}$ використовуються різні співвідношення, що не потребують ні того, ні іншого.

Співвідношення, що пов'язують точки $X^{(k+1)}$ і $X^{(k)}$ у випадку квадратичної цільової функції, наступні:

$$\nabla f\left(X^{(k+1)}\right) - \nabla f\left(X^{(k)}\right) = H\left(X^{(k)}\right)\left(X^{(k+1)} - X^{(k)}\right) \tag{3.14}$$

Помноживши обидві частини цього рівняння на $H^{-1}(X^{(k)})$, отримаємо:

$$X^{(k+1)} - X^{(k)} = H^{-1}\left(X^{(k)}\right)\left[\nabla f\left(X^{(k+1)}\right) - \nabla f\left(X^{(k)}\right)\right] \tag{3.15}$$

У достатньо великій групі методів $H^{-1}(X^{(k+1)})$ апроксимується за допомогою інформації, яка була отримана на K-му кроці:

$$H^{-1}\left(X^{(k+1)}\right) \approx \omega \cdot \eta^{(k+1)} = \omega\left(\eta^{(k)} + \Delta\eta^{(k)}\right), \tag{3.16}$$

де ω – масштабний множник, константа, яка звичайно дорівнює одиниці.

Вибір $\Delta\eta^{(k)}$ по суті визначає метод змінної метрики. Для забезпечення збіжності матриця $\Delta\eta^{(k)}$ повинна бути додатно визначеною і задовольняти рівняння (3.15) в тому випадку, коли вона замінює H^{-1}. На $(k + 1)$ кроці відомі значення $X^{(k)}$, $\nabla f\left(X^{(k)}\right)$, $\nabla f\left(X^{(k+1)}\right)$ і $\eta^{(k)}$. Потрібно визначити $\eta^{(k+1)}$ так, щоб задовольняти співвідношення

$$\eta^{(k+1)} \Delta g^{(k)} = \frac{1}{\omega} \Delta X^{(k)}. \tag{3.17}$$

Нехай $\Delta\eta^{(k)} = \eta^{(k+1)} - \eta^{(k)}$. Тоді рівняння

$$\eta^{(k+1)} \Delta g^{(k)} = \frac{1}{\omega} \Delta X^{(k)}. \tag{3.18}$$

Потрібно вирішити відносно $\Delta\eta^{(k)}$:

$$\eta^{(k)} \Delta g^{(k)} = \frac{1}{\omega} \Delta X^{(k)} - \eta^{(k)} \Delta g^{(k)}, \tag{3.19}$$

де y і Z - довільні вектори.

Алгоритм Бройдена

Якщо для ω = 1 в рівнянні (3.19) вибирається спеціальна лінійна комбінація двох напрямків $\Delta X^{(k)}$ і $\eta^{(k+1)}\Delta g^{(k)}$, а саме $y = Z = \Delta X^{(k)}\eta^{(k+1)}\Delta g^{(k)}$, то використовується алгоритм Бройдена, в якому $\Delta\eta^{(k)}$ являється

$$\Delta\eta^{(k)} = \frac{\left[\Delta X^{(k)} - \eta^{(k)}\Delta g^{(k)}\right]\left[\Delta X^{(k)} - \eta^{(k)}\Delta g^{(k)}\right]^T}{\left[\Delta X^{(k)} - \eta^{(k)}\Delta g^{(k)}\right]^T \Delta g^{(k)}}, \tag{3.20}$$

де $\Delta X^{(k)}$.

Коли для кожного напрямку пошуку $\lambda^{(k)}$ являє собою скаляр, що мінімізує $f(X)$ в цьому напрямку, то даний метод дає спряжені напрямки пошуку. У випадку, коли цільова функція не є квадратичною, використання рівняння (3.20) може привести до наступних негативних явищ:

1. Матриця η може перестати бути додатно визначеною;
2. Обчислюване значення $\eta^{(k)}$ може стати необмеженим (інколи навіть у випадку квадратичної функції) внаслідок помилок при заокругленні.
3. Якщо $\lambda X^{(k)} = -\lambda^{*(k)}\eta(X^{(k)})\nabla f(X^{(k)})$ $f(X^{(k)})$ випадково співпадає у напрямку попереднього етапу, тоді матриця $\eta(X^{(k)})$ стає невизначеною.

Завдання:

1. Мінімізувати алгоритмом Бройдена наступні цільові функції:

а) $f(X) = 100(X_2 - X_1^2)^2 + (1 - X_1)^2, \quad X^{(0)} = [-0{,}5 \quad 0{,}5]^T$;

б) ‡) $f(X) = 4(X_1 - 5)^2 + (X_2 - 6)^2, \quad X^{(0)} = [8 \quad 9]^T$;

в) $f(X) = X_1^2 + X_2^2 - 4, \quad X^{(0)} = [4 \quad 4]^T$.

2. Мінімізувати алгоритмом Бройдена наступні цільові функції:

г) $f(X) = \left\{12 + X_1^2 + \frac{1 + X_2^2}{X_1^2} + \frac{X_1^2 X_2^2 + 100}{(X_1 X_4)^4}\right\}\frac{1}{10}, \quad X^{(0)} = [0{,}5 \quad 0{,}5]^T$;

$$f(X) = 100(X_2 - X_1^2)^2 + (1 - X_1)^2 + 90(X_4 - X_3^2)^2 + (1 - X_3)^2 +$$
$$+10{,}1\left[(X_2 - 1)^2 + (X_4 - 1)\right] + 19{,}8(X_2 - 1)(X_4 - 1), \quad X^{(0)} = [-3 \quad -1 \quad -3 \quad -1]^T.$$

Завдання для самостійної роботи:

Мінімізувати алгоритмом Бройдена наступні цільові функції:

а) $f(X) = X_1^3 \exp\left[X_2 - X_1^2 - 10(X_1 - X_2)^2\right]$;

б) $$f(X) = (0{,}35 + 0{,}4X_1 + 0{,}31X_2)^4 (0{,}85 - 0{,}6X_1 + 0{,}85X_2)^4 \times$$
$$\times \exp\left[2 - (0{,}35 - 0{,}4X_1 + 0{,}35X_2)^4\right];$$

в) $f(X)=X_1^2\exp\left[1-X_1^2-20{,}25\left(X_1-X_2\right)^2\right].$

Контрольні запитання

1. Чому додатна визначеність матриці η$^{(k)}$ є необхідною умовою при рішенні задач мінімізації за допомогою методів змінної метрики?
2. Вивести рекурентне рівняння для визначення матриці напрямків за допомогою алгоритму Бройдена, якщо необхідно максимізувати, а не мінімізувати цільову функцію.
3. Як найлегшим засобом використовувати алгоритм Бройдена для максимізації, а не для мінімізації?

Метод Девідона-Флетчера-Пауелла

Якщо в рівнянні (3.19) береться $y = \Delta X^{(k)}$, $Z = \eta^{(k)}\Delta g^{(k)}$, то матрицю η$^{(k+1)}$ визначаємо за допомогою алгоритму Девідону-Флетчера-Пауелла (ДФП):

$$\eta^{(k+1)}=\eta^{(k)}+A^{(k)}-e^{(k)}=\eta^{(k)}+\frac{\Delta X^{(k)}\left(\Delta X^{(k)}\right)^T}{\left(\Delta X^{(k)}\right)^T\Delta g^{(k)}}-\frac{r^{(k)}\Delta g^{(k)}\left(\Delta g^{(k)}\right)^T\left(\eta^{(k)}\right)^T}{\left(\Delta g^{(k)}\right)^T\eta^{(k)}\Delta g^{(k)}}. \qquad (3.21)$$

Початкова матриця η звичайно вибирається у вигляді одиничної матриці η$^{(0)}$=1 (але може бути і будь-якою симетричною додатно визначеною матрицею), так що початковий напрямок мінімізації – напрям найшвидшого спуску. Під час оптимізації має місце поступовий перехід від градієнтного напрямку до ньютонівського; при цьому використовуються перевага кожного з них на відповідному етапі. Рекурентне співвідношення (3.21) на практиці цілком задовольняє, якщо помилка в розрахунку $f(X)$ невелика. Роль матриці $A^{(k)}$ полягає в забезпеченні того, щоб η→$H^{(-1)}$, тоді як матриця $B^{(k)}$ забезпечує додатну визначеність на всіх етапах і в межах виключає початкову матрицю η$^{(0)}$.

Використаємо формулу (5.21) на кількох етапах, починаючи з η$^{(0)}$:

$$\eta^{(1)}=I+A^{(0)}-B^{(0)},$$

$$\eta^{(2)}=\eta^{(1)}+A^{(1)}-B^{(1)}=I+\left(A^{(0)}+A^{(1)}\right)-\left(B^{(0)}+B^{(1)}\right),$$

…………………………………………………………

$$\eta^{(k+1)}=I+\sum_{i=0}^{k}A^{(i)}-\sum_{i=0}^{k}B^{(i)}.$$

У випадку квадратичної функції сума матриць $A^{(i)}$ повинна дорівнювати H^{-1} при $k = n - 1$, а сума матриць $B^{(i)}$ – будується так, щоб вона скоротилась з матрицею, яка була обрана в якості початкової матриці η$^{(0)}$ (тут - одиничної матриці). Потрібно визначити, що у випадку квадратичної цільової функції в алгоритмі ДФП використовується спряжені напрямки. Щоб кінцевий напрямок $S^{(n-1)}$ був спряжений по відношенню до всіх попередніх напрямків, повинна виконуватись рівність $\left(X^{n-1}\right)^T HS^{n-1} = 0$ або при $S^{n-1} = -\eta^{n-1}\nabla f\left(X^{n-1}\right)$:

$$\left(X^{n-1}\right)^T H\eta^{n-1}\nabla f\left(X^{n-1}\right) = 0 \qquad (3.22)$$

У випадку загальної цільової функції ефективність методу ДФП є скоріше наслідком використання спряжених напрямків, чим близькою до апроксимації H^{-1} матрицею η. В деяких задачах неможливо досягнути мінімуму цільової функції за допомогою методів змінної метрики, якщо ступінь точності одномірного пошуку недостатня, тому рекомендується, щоб точність одномірного пошуку була по крайній мірі еквівалентна точності, яка потрібна для закінчення основного алгоритму.

Метод ДФП протягом ряду років продовжує залишатись найбільш широко застосовним при рішенні різноманітних задач. Основним недоліком методів такого типу є необхідність зберігати в пам’яті ЕОМ матрицю η порядку $n\times n$.

Завдання:

1. Мінімізувати наступні цільові функції методом ДФП:

а) $f\left(X\right) = 100\left(X_2 - X_1^2\right)^2 + \left(1 - X_1\right)^2, \quad X^{(0)} = \left[-0{,}5 \quad 0{,}5\right]^T$;

б) $f\left(X\right) = 4\left(X_1 - 5\right)^2 + \left(X_2 - 6\right)^2, \quad X^{(0)} = \left[8 \quad 9\right]^T$;

в) $f(X) = X_1^2 + X_2^2 - 4, \quad X^{(0)} = \left[4 \quad 4\right]^T$.

2. Мінімізувати наступні цільові функції методом ДФП:

$$f\left(X\right) = \left(X_1 + 10X_2\right)^2 + 5\left(X_3 - X_4\right)^2 + \left(X_2 - 2X_3\right)^2 + 10\left(X_1 - X_4\right)^4, \quad X^{(0)} = \left[3 \quad -10\right]^T;$$

Завдання для самостійної роботи:

Вивчити вплив ε – параметру збіжності для пошуку вздовж прямої на процедуру розрахунків за методами Флетчера-Рівса, Бройдена, Девідона-Флетчера-Пауелла. Використовуючи кожен з перерахованих методів для рішення задачі мінімізації функції вигляду Вуда

$$f(X) = 100\left(X_2 - X_1^2\right)^2 + \left(1 - X_1\right)^2 + 90\left(X_4 - X_3^2\right)^2 + \left(1 - X_3\right)^2 + 10{,}1\left[\left(X_2 - 1\right)^2 + \left(X_4 - 1\right)^2\right] +$$
$$+ 19{,}8\left(X_2 - 1\right)\left(X_4 - 1\right), \quad X^{(0)} = \left[-3 \; -1 \; -3 \; -1\right]^T,$$

при $\varepsilon_2 = \alpha\varepsilon_1$; $\alpha = 0{,}01$; $0{,}1$; 1 і 10.

Контрольні запитання

1. Чи використовується в алгоритмі ДФП спряжені напрямки?
2. Роль матриць $A^{(k)}$ і $B^{(k)}$ в рекурентному відношенні для обчислення матриці $\eta^{(k+1)}$.
3. Що є початковим напрямком мінімізації цільових функцій методом ДФП?

Алгоритм Пірсона

Пірсон запропонував кілька методів обчислення матриці η, які використовують спряжені напрямки пошуку. Алгоритм Пірсона можна отримати, задаючи різним образом вектори y і Z в рівнянні (3.19).

1. Алгоритм Пірсона №2.

Покладемо в рівнянні (5.19) $y = Z = \Delta X^{(k)}$, а ω = 1. Тоді

$$\eta^{(k+1)} = \eta^{(k)} + \frac{\left[\Delta X^{(k)} - \eta^{(k)} \Delta g^{(k)}\right]\left[\Delta X^{(k)}\right]^T}{\left[\Delta X^{(k)}\right]^T \Delta g^{(k)}} \quad \eta^{(0)} = R^{(0)},$$

де $R^{(0)}$ – довільна додатно визначена симетрична матриця.

Якщо через кілька кроків матриця напрямку стає поганою, коливаючись між додатно і від'ємно визначеною, повтор початку алгоритму через кожні n кроків (тобто приведення $\eta^{(n+1)}$ до $R^{(0)}$ після кожних n кроків) допомагає уникнути труднощів такого роду.

2. Алгоритм Пірсона №3.

Покладемо в рівнянні (3.19) $y = Z = \eta^{(k)} \Delta g^{(k)}$, а ω = 1. Тоді

$$\eta^{(k+1)} = \eta^{(k)} + \left[\Delta X^{(k)} - \eta^{(k)} \Delta g^{(k)}\right] \cdot \frac{\left[\eta^{(k)} \Delta g^{(k)}\right]^T}{\left[\Delta g^{(k)}\right]^T \eta^{(k)} \Delta g^{(k)}}. \; .$$

Завдання:

1. Мінімізувати алгоритмом Пірсона наступні функції:

а) $f(X) = 100\left(X_2 - X_1^2\right)^2 + \left(1 - X_1\right)^2, \quad X^{(0)} = \left[-0.5\; 0.5\right]^T;$

б) $f(X) = 4\left(X_1 - 5\right)^2 + \left(X_2 - 6\right)^2, \quad X^{(0)} = \left[8\; 9\right]^T;$

в) $f(X) = X_1^2 + X_2^2 - 4, \quad X^{(0)} = \left[4\; 4\right]^T;$

Порівняти траєкторії оптимізації в просторі X при використанні методів змінної метрики.

2. Мінімізувати наступні цільові функції алгоритмом Пірсона і порівняти їх траєкторії оптимізації в просторі X:

а) $f(X) = 3\left(X_1 - 4\right)^2 + 5\left(X_2 + 3\right)^2 + 7\left(2X_3 + 1\right)^2;$

б) $f(X) = 1 - 2X_1 - 2X_2 - 4X_1X_2 + 10X_1^2 + 2X_2^2;$

в) $f(X) = X_1^4 + X_2^4 + 2X_1^2X_2^2 - 4X + 3;$

г) $f(X) = (X_1^2 + X_2 - 11)^2 + (X_1 + X_2^2 - 7)^2;$

д) $f(X) = X_1^3 + X_2^3 - 3X_1 - 2X_2 + 2.$

Завдання для самостійної роботи:

Вивести рекурентне рівняння для визначення матриць напрямків за допомогою алгоритмів Пірсона.

Контрольні запитання.

Як уникнути труднощів в зв’язку з тим, що матриця напрямків *X* в процесі пошуку перестає бути додатно визначеною? Швидкість збіжності найлегших градієнтних методів, методу найшвидшого спуску залишається недопустимо низькою при рішенні ряду практичних задач. Метод найшвидшого спуску часто використовується при реалізації градієнтних методів як початкової процедури. Результати досліджень градієнтних методів [17] показали переваги квазіньютонівських методів. Однак точність розрахунків на ЕОМ здійснює більший вплив на реалізацію квазіньютонівських методів, ніж на реалізацію методів спряжених градієнтів. Це дозволило зробити висновок про те, що у розрахунках на мікроЕОМ (подібних до тих, що використовуються в управлінні технологічними процесами) метод Флетчера-Рівса може виявитись більш ефективним. Д. Хіммельблау [13] провів дослідження збіжності градієнтних методів. Процедура ДФП представляється кращою в змісті загального методу нелінійного програмування при відсутності обмежень по відношенню до отримання вірних значень вектору *X* в точці мінімуму цільової функції. Метод Ньютона завжди видавався легшим; що стосується інших методів, то, за виключенням методу Флетчера-Рівса, який не спрацьовує в деяких тестах, всі вони видались рівної ефективності. Використання методів змінної метрики з перезавданням матриці η є найбільш придатним на ранніх етапах оптимізації, тоді як відмова від перезавдання віддається перевага на більш пізніх етапах, коли *X* буває настільки близько до Х, що ефективними стають властивості спряженості алгоритмів змінної метрики. На основі порівняння часу рішення найліпшими видались алгоритми ДФП; сприятливими – алгоритми Флетчера-Рівса, Пірсона №3; надійними – алгоритм Пірсона №2.

Порівняння ефективності алгоритмів методів нелінійного програмування з використанням похідних

Як критерій для оцінювання алгоритмів градієнтних методів нелінійного програмування при відсутності обмежень часто розглядаються наступні:

1. Успіх в досягненні оптимального рішення (в межах заданої степені точності) для широкого класу задач.
2. Число необхідних визначень цільової функції.
3. Машинний час, який потрібен для реалізації алгоритмів (в межах бажаної степені точності).

Метод зветься збіжним, якщо виконується нерівність:

$$\frac{\left|\varepsilon^{k+1}\right|}{\left|\varepsilon^{k}\right|} \le 1, \tag{3.23}$$

де $\varepsilon = X^{(k)} - X^{*}$, де $X^{(k)}$ – результат K-ої ітерації; X^{*} – рішення виконується на кожній ітерації.

Оскільки при розрахунках частіше застосовують кінцеві десяткові дроби, навіть самий ефективний алгоритм потребує нескінченної послідовності ітерацій. Тому, в першу чергу, інтерес представляють асимптотичні властивості збіжності методів, що вивчаються. Алгоритм має збіжність порядку r, якщо

$$\frac{\left|\varepsilon^{k+1}\right|}{\left|\varepsilon^{k}\right|} = C,$$

де C – постійна величина.

Якщо $r = 1$ чи $r = 2$, то алгоритм характеризується лінійною чи квадратичною швидкістю збіжності відповідно. При $r = 1$ і $C = 0$ алгоритм характеризується суперлінійною швидкістю збіжності.

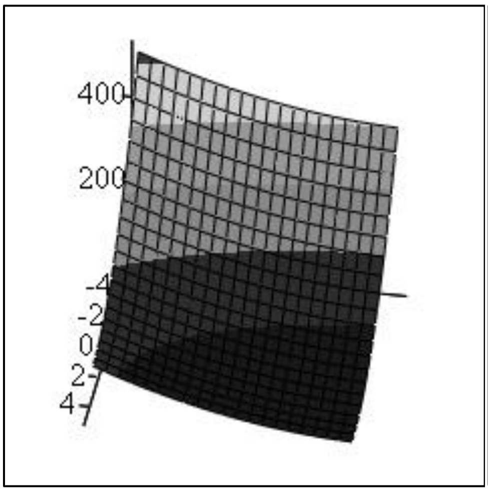
Метод релаксації

Введіть $y(x_0, x_1)$ – функцію:

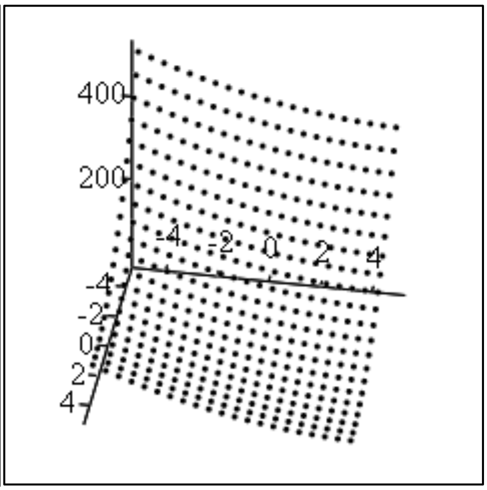
$$y(x0, x1) := 4(x0 - 5)^2 + (x1 - 6)^2$$

Кількість змінних:

$$n := 2$$



y



y

Розрахуємо частинні похідні функції y(x0,x1):

$$\frac{\partial}{\partial x0} y(x0, x1) \rightarrow 8 \cdot x0 - 40$$

$$\frac{\partial}{\partial x1} y(x0, x1) \rightarrow 2 \cdot x1 - 12$$

Введіть $y(x_0, x_1)$ – функцію ще раз, але у вигляді $y(x)$:

$$y(x) := 4\left(x_0 - 5\right)^2 + \left(x_1 - 6\right)^2$$

ліва границя

$$XLB := \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

права границя

$$XRB := \begin{pmatrix} 8 \\ 8 \end{pmatrix}$$

початкова точка

$$X0 := \begin{pmatrix} 2 \\ 3 \end{pmatrix}$$

Введіть раніше розраховані частинні похідні функції $y(x_0, x_1)$:

$$y\_x0(x) := 8 \cdot x_0 - 40$$

крок

$$h := 0.1$$

величина приросту для розрахунку частинних похідних

$$dx := 0.01$$

точність обчислень

$$\varepsilon := 0.001$$

$$y\_x1(x) := 2 \cdot x_1 - 12$$

```
scan_min(y, a, b, k, h) := | x1 ← a
                           | F1 ← y(x1)
                           | x ← a
                           | for i ∈ 1 .. |b_k − a_k| / h  if b_k ≠ a_k
                           |    | x_k ← x_k + h
                           |    | F ← y(x)
                           |    | if F < F1
                           |    |    | F1 ← F
                           |    |    | x1 ← x
                           | x1

msign(val) := | 1  if val > 0
              | 0  otherwise

search_f_(vval, val, n) := | i ← 0
                           | for j ∈ 0, 1 .. n − 1
                           |    | i ← j  if |val_j| = vval
                           |    | T_j ← msign(val_j)
                           | (i, T)
```

$$ones\_matrix(x, y) := x = y$$

```
relacs_min(y, X0, XLB, XRB, n, dx, h, ε, get_val_) := | j ← 0
                                                      | kr ← 1
                                                      | H ← h·matrix(n, n, ones_matrix)
                                                      | while kr
                                                      |   | (i, T) ← search_f_(get_val_(X0)_1, get_val_(X0)_0, n)
                                                      |   | j ← j + 1
                                                      |   | t ← T_i
                                                      |   | x ← X0 − t·H^<i>  if (XRB ≥ X0 − t·H^<i>)·(XLB ≤ X0 − t·H^<i>) = n
                                                      |   | xk ← X0 − (t − 1)·H^<i>  if ⌊XRB ≥ X0 − (t − 1)·H^<i>⌋·⌊XLB ≤ X0 − (t − 1)·H^<i>⌋ = n
                                                      |   | X ← scan_min(y, x, xk, i, dx)
                                                      |   | kr ← 0  if sqrt(Σ(X − X0)^2) ≤ ε
                                                      |   | X0 ← X
                                                      | (X, j)
```

$$get\_val\_2(X) := \begin{pmatrix} stack(y\_x0(X), y\_x1(X)) \\ max(|y\_x0(X)|, |y\_x1(X)|) \end{pmatrix}$$

$$\begin{pmatrix} xres_0 \\ i_0 \end{pmatrix} := relacs\_min(y, X0, XLB, XRB, n, dx, h, \varepsilon, get\_val\_2)$$

$$xres_0 = \begin{pmatrix} 5 \\ 6 \end{pmatrix}$$

$$i_0 = 61$$

$$Arg := xres_0$$

$$Funct := y(Arg)$$

$$Iter := i_0$$

Розв'язок за допомогою методу релаксації:

Мінімум функції на заданому інтервалі знаходиться у точці:

$$Arg = \begin{pmatrix} 5 \\ 6 \end{pmatrix}$$

Значення мінімуму функції:

$$Funct = 0$$

Кількість ітерацій:

$$Iter = 61$$

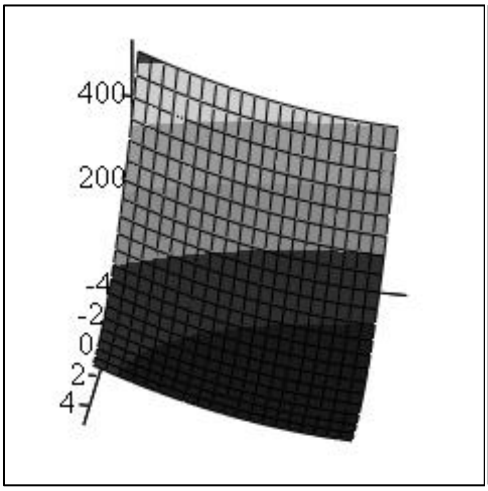
Метод градієнту

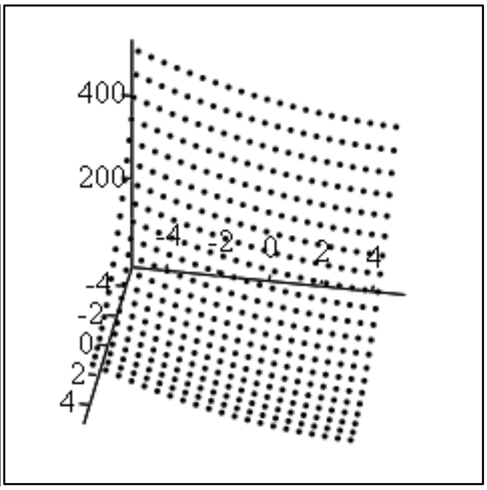
Введіть $y(x_0,x_1)$ – функцію:

$$y(x0, x1) := 4\,(x0 - 5)^2 + (x1 - 6)^2$$

Кількість змінних:

$$n := 2$$





y y

Розрахуємо частинні похідні функції $y(x_0, x1)$:

$$\frac{\partial}{\partial x0} y(x0, x1) \rightarrow 8 \cdot x0 - 40$$

$$\frac{\partial}{\partial x1} y(x0, x1) \rightarrow 2 \cdot x1 - 12$$

Введіть $y(x_0, x_1)$ – функцію ще раз, але у вигляді $y(x)$:

$$y(x) := 4 \left| x_0 - 5 \right|^2 + \left| x_1 - 6 \right|^2$$

ліва границя

$$XLB := \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

права границя

$$XRB := \begin{pmatrix} 8 \\ 8 \end{pmatrix}$$

початкова точка

$$X0 := \begin{pmatrix} 2 \\ 3 \end{pmatrix}$$

Введіть раніше розраховані частинні похідні функції $y(x_0, x_1)$:

$$y\_x0(x) := 8 \cdot x_0 - 40$$

крок

$$h := 0.1$$

величина приросту для розрахунку частинних похідних

$$dx := 0.01$$

точність обчислень

$$\varepsilon := 0.001$$

$$y\_x1(x) := 2 \cdot x_1 - 12$$

$$not(in) := \left| \begin{array}{l} n \leftarrow rows(in) \\ \text{for } i \in 0..n-1 \\ \quad in_i \leftarrow \neg in_i \\ in \end{array} \right.$$

$$baunds\_check(resx, x, xb) := \left| \begin{array}{l} n \leftarrow rows(x) \\ \text{for } i \in 0..n-1 \\ \quad resx_i \leftarrow xb_i \ \text{ if } \ x_i \\ resx \end{array} \right.$$

$$gradient\_min(y, grad, X0, XLB, XRB, h, \varepsilon) := \left| \begin{array}{l} kr \leftarrow 1 \\ \text{while } kr \\ \quad \left| \begin{array}{l} h\_ \leftarrow h \\ kr2 \leftarrow 1 \\ \text{while } kr2 \\ \quad \left| \begin{array}{l} Y0 \leftarrow y(X0) \\ X \leftarrow X0 - h\_ \cdot grad(X0) \\ X \leftarrow baunds\_check(X, not(XRB > X), XRB) \\ X \leftarrow baunds\_check(X, not(XLB < X), XLB) \\ Y \leftarrow y(X) \\ h\_ \leftarrow \dfrac{h\_}{2} \ \text{ if } \ Y > Y0 \\ kr2 \leftarrow 0 \ \text{ if } \ Y \le Y0 \end{array} \right. \\ kr \leftarrow 0 \ \text{ if } \ \sqrt{\sum (X - X0)^2} \le \varepsilon \\ X0 \leftarrow X \end{array} \right. \\ X \end{array} \right.$$

$$grad(x) := \begin{pmatrix} y\_x0(x) \\ y\_x1(x) \end{pmatrix}$$

$$Arg := gradient\_min(y, grad, X0, XLB, XRB, h, \varepsilon)$$

$$Funct := y(Arg)$$

Розв'язок за допомогою методу градієнту:

Мінімум функції на заданому інтервалі знаходиться у точці:

$$\mathrm{Arg} = \begin{pmatrix} 5 \\ 5.996 \end{pmatrix}$$

Значення мінімуму функції:

$$\mathrm{Funct} = 1.379 \times 10^{-5}$$

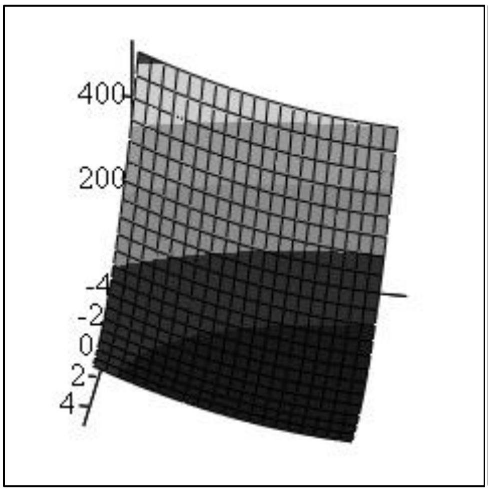
Метод найшвидшого спуску

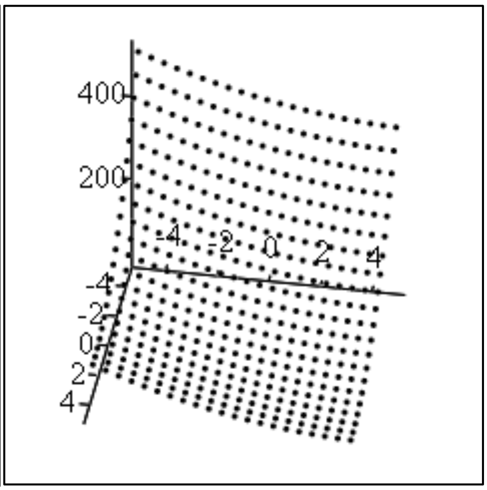
Введіть $y(x_0, x_1)$ – функцію:

$$y(x0, x1) := 4\,(x0 - 5)^2 + (x1 - 6)^2$$

Кількість змінних:

$$n := 2$$





y y

Розрахуємо частинні похідні функції $y(x_0, x_1)$:

$$\frac{\partial}{\partial x0} y(x0, x1) \rightarrow 8 \cdot x0 - 40$$

$$\frac{\partial}{\partial x1} y(x0, x1) \rightarrow 2 \cdot x1 - 12$$

Введіть $y(x_0, x_1)$ – функцію ще раз, але у вигляді $y(x)$:

$$y(x) := 4\left|x_0 - 5\right|^2 + \left|x_1 - 6\right|^2$$

ліва границя

$$XLB := \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

права границя

$$XRB := \begin{pmatrix} 8 \\ 8 \end{pmatrix}$$

початкова точка

$$X0 := \begin{pmatrix} 2 \\ 3 \end{pmatrix}$$

Введіть раніше розраховані частинні похідні функції $y(x_0, x_1)$:

$$y\_x0(x) := 8 \cdot x_0 - 40$$

крок

$$h := 0.1$$

величина приросту для розрахунку частинних похідних

$$dx := 0.01$$

точність обчислень

$$\varepsilon := 0.001$$

$$y\_x1(x) := 2 \cdot x_1 - 12$$

$$\mathrm{not}(in) := \left| \begin{array}{l} n \leftarrow \mathrm{rows}(in) \\ \text{for } \ i \in 0..\, n-1 \\ \quad in_i \leftarrow \neg in_i \\ in \end{array} \right.$$

$$\mathrm{baunds\_check}(resx, x, xb) := \left| \begin{array}{l} n \leftarrow \mathrm{rows}(x) \\ \text{for } \ i \in 0..\, n-1 \\ \quad resx_i \leftarrow xb_i \ \text{ if } \ x_i \\ resx \end{array} \right.$$

```
gradient_min(y,grad,X0,XLB,XRB,h,ε) := | kr ← 1
                                       | while kr
                                       |   | h_ ← h
                                       |   | kr2 ← 1
                                       |   | while kr2
                                       |   |   | Y0 ← y(X0)
                                       |   |   | X ← X0 − h_·grad(X0)
                                       |   |   | X ← baunds_check(X,not(XRB > X),XRB)
                                       |   |   | X ← baunds_check(X,not(XLB < X),XLB)
                                       |   |   | Y ← y(X)
                                       |   |   | h_ ← h_/2  if  Y > Y0
                                       |   |   | kr2 ← 0  if  Y ≤ Y0
                                       |   | kr ← 0  if  sqrt(Σ(X − X0)^2) ≤ ε
                                       |   | X0 ← X
                                       | X
```

```
look_for_h(y,X,a,b,h) := | x1 ← a
                         | F1 ← y(x1,X)
                         | x ← a
                         | for  i ∈ 1..(b − a)/h   if  b ≠ a
                         |   | x ← x + h
                         |   | F ← y(x,X)
                         |   | if (F ≤ 1)
                         |   |   | F1 ← F
                         |   |   | x1 ← x
                         | x1
```

$$\mathrm{spusk}(f, X, XLB, XRB, grad, Z, \varepsilon) := \left| \begin{array}{l} kr \leftarrow 1 \\ i \leftarrow 0 \\ \text{while } kr \\ \quad \left| \begin{array}{l} F \leftarrow f(X) \\ Xold \leftarrow X \\ X \leftarrow X - look\_for\_h(Z, X, 0, 0.1, 0.01) \cdot grad(X) \\ X \leftarrow baunds\_check(X, not(XRB > X), XRB) \\ X \leftarrow baunds\_check(X, not(XLB < X), XLB) \\ kr \leftarrow 0 \;\; \text{if } (X = Xold) \vee \left(\sqrt{\sum grad(X)^2} < \varepsilon \right) \\ i \leftarrow i + 1 \end{array} \right. \\ \begin{pmatrix} X \\ i \end{pmatrix} \end{array} \right.$$

$$grad(x) := \begin{pmatrix} y\_x0(x) \\ y\_x1(x) \end{pmatrix}$$

$$z(h2, X0) := \frac{d}{dh2} y\left(\begin{pmatrix} X0_0 - h2 \cdot grad(X0)_0 \\ X0_1 - h2 \cdot grad(X0)_1 \end{pmatrix} \right)$$

$$\begin{pmatrix} xres_2 \\ i_2 \end{pmatrix} := \mathrm{spusk}(y, X0, XLB, XRB, grad, z, \varepsilon)$$

$$xres_2 = \begin{pmatrix} 5 \\ 6 \end{pmatrix}$$

$$i_2 = 39$$

$$Arg := xres_2$$

$$Funct := y(Arg)$$

$$Iter := i_2$$

Розв'язок за допомогою методу найшвидшого спуску:

Мінімум функції на заданому інтервалі знаходиться у точці:

$$Arg = \begin{pmatrix} 5 \\ 6 \end{pmatrix}$$

Значення мінімуму функції:

$$Funct = 2.485 \times 10^{-7}$$

Кількість ітерацій:

$$Iter = 39$$

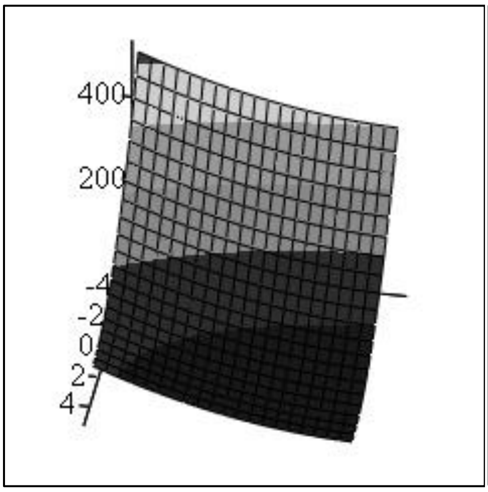
Метод Девідона-Флетчера-Пауелла

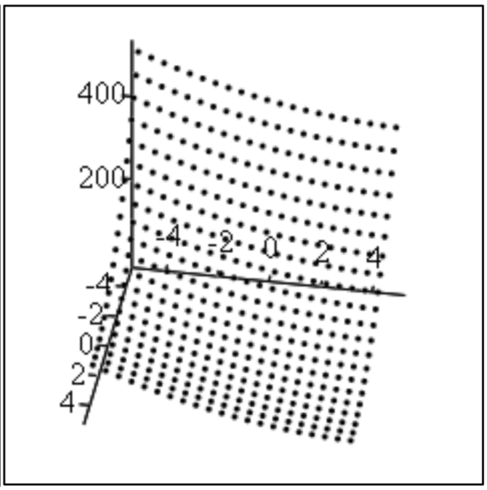
Введіть $F(x_1, x_2)$ – функцію:

$$F(x1, x2) := 4\,(x1-5)^2 + (x2-6)^2$$

Кількість змінних:

$$n := 2$$





F F

Введіть значення початкової точки та точність обчислень:

$$Func(x) := F\left(x_0, x_1\right)$$

початкова точка

$$x := \begin{pmatrix} 2 \\ 3 \end{pmatrix}$$

межі зміни кроку при одномірній оптимізації

$$gr\_h \equiv \begin{pmatrix} 0.1 \\ 1 \end{pmatrix}$$

точність обчислень

$$\varepsilon := 0.0001$$

одинична матриця

$$D := \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

крок одномірної оптимізації

$$step\_odn \equiv 0.01$$

величина приросту для розрахунку частинних похідних

$$dx \equiv 0.001$$

функція grad обчислює значення градієнту в точці x

$$
grad(x,n) := \left|
\begin{array}{l}
\text{for } i \in 0 .. n-1 \\
\quad \left|
\begin{array}{l}
x1 \leftarrow x \\
x1_i \leftarrow x_i + dx \\
pr_i \leftarrow \dfrac{Func(x1) - Func(x)}{dx}
\end{array}
\right. \\
\text{return } pr
\end{array}
\right.
$$

функція norm використовується для отримання нормалізованого вектора градієнту

$$
norm(x,n) := \left|
\begin{array}{l}
s \leftarrow 0 \\
pr \leftarrow grad(x,n) \\
\text{for } i \in 0 .. n-1 \\
\quad s \leftarrow s + \left| pr_i \right|^2 \\
\text{return } \sqrt{s}
\end{array}
\right.
$$

функція scan використовується для отримання оптимального кроку h_j методом сканування

$$
scan(x,s) := \left|
\begin{array}{l}
x1 \leftarrow gr\_h_0 \\
F1 \leftarrow Func(x + x1 \cdot s) \\
x2 \leftarrow gr\_h_0 \\
\text{for } i \in 1 .. \dfrac{\left| gr\_h_1 - gr\_h_0 \right|}{step\_odn} \\
\quad \left|
\begin{array}{l}
x2 \leftarrow x2 + step\_odn \\
F \leftarrow Func(x + x2 \cdot s) \\
\text{if } F < F1 \\
\quad \left|
\begin{array}{l}
F1 \leftarrow F \\
x1 \leftarrow x2
\end{array}
\right.
\end{array}
\right. \\
\text{return } x1
\end{array}
\right.
$$

```
F_Pauel(x,D,e,n) := | k ← 1
                    | steps ← 0
                    | for i ∈ 0..n − 1
                    |   | XX_i ← x
                    |   | DD_i ← D
                    | while 1
                    |   | j ← 0
                    |   | F_j ← Func(XX_j)
                    |   | while 1
                    |   |   | S_j ← −DD_j · grad(XX_j, n)
                    |   |   | h_j ← scan(XX_j, S_j)
                    |   |   | step_steps ← h_j
                    |   |   | steps ← steps + 1
                    |   |   | dex_j ← h_j · S_j
                    |   |   | XX_(j+1) ← XX_j + dex_j
                    |   |   | F_(j+1) ← Func(XX_(j+1))
                    |   |   | if norm(XX_(j+1), n) > e
                    |   |   |   | if j ≥ n − 1
                    |   |   |   |   | k ← k + 1
                    |   |   |   |   | XX_0 ← XX_(j+1)
                    |   |   |   |   | break
                    |   |   |   | otherwise
                    |   |   |   |   | ■
                    |   |   |   |   | ■
                    |   |   |   |   | ■
                    |   |   |   |   | ■
                    |   |   |   |   | ■
                    |   |   | return ( XX_(j+1)
                    |   |   |          Func(XX_(j+1))
                    |   |   |          step^T
                    |   |   |          k )            otherwise
```

$$Rez := F\_Pauel(x, D, \varepsilon, n)$$

$$Funct := Rez_1$$

$$Arg := Rez_0$$

$$h := Rez_2$$

$$Iter := Rez_3$$

Розв'язок за допомогою методу Девідона-Флетчера-Пауелла:

Масив оптимальних кроків:

h =

	0	1	2	3	4	5	6	7	8	9
0	0.13	0.49	0.17	0.47	0.1	0.1	0.1	0.1	0.1	0.65

Мінімум функції на заданому інтервалі знаходиться у точці:

$$Arg = \begin{pmatrix} 5 \\ 6 \end{pmatrix}$$

Значення мінімуму функції:

$$Funct = 1.218 \times 10^{-6}$$

Кількість ітерацій:

$$Iter = 6$$

Оптимальний розв'язок:

$$M := Minimize(Func, x)$$

$$M = \begin{pmatrix} 5 \\ 6 \end{pmatrix}$$

$$Fmin := Func(M)$$

$$Fmin = 0$$

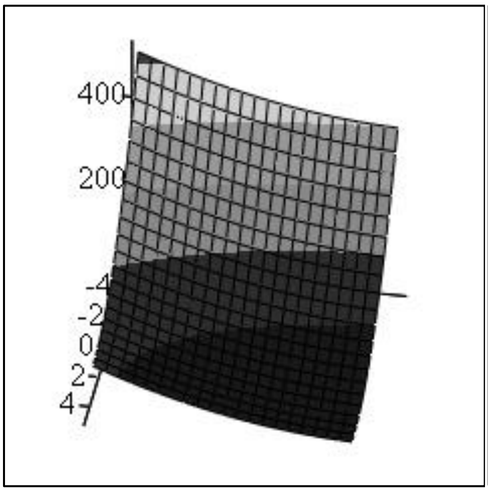
Метод спряженого градієнту Флетчера-Рівса

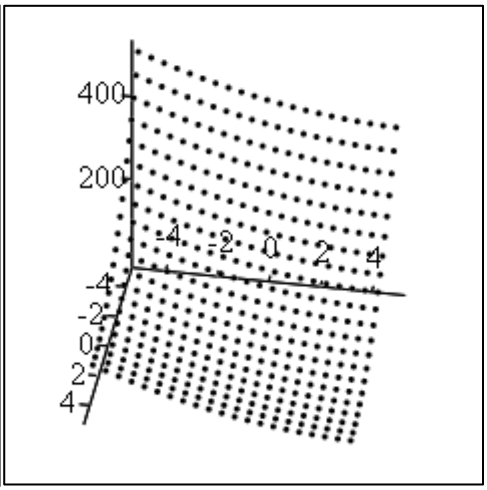
Введіть $F(x_1, x_2)$ – функцію:

$$F(x1, x2) := 4(x1 - 5)^2 + (x2 - 6)^2$$

Кількість змінних:

$$n := 2$$





F F

Введіть значення початкової точки та точність обчислень:

$$\mathrm{Func}(x) := F\left(x_0, x_1\right)$$

початкова точка

$$x := \begin{pmatrix} 2 \\ 3 \end{pmatrix}$$

межі зміни кроку при одномірній оптимізації

$$gr\_h \equiv \begin{pmatrix} 0.1 \\ 1 \end{pmatrix}$$

точність обчислень

$$\varepsilon := 0.0001$$

крок одномірної оптимізації

$$step\_odn \equiv 0.01$$

величина приросту для розрахунку частинних похідних

$$dx \equiv 0.001$$

функція grad обчислює значення градієнту в точці *x*

$$
grad(x,n) := \left|\begin{array}{l}
\text{for } i \in 0..n-1 \\
\quad \left|\begin{array}{l}
x1 \leftarrow x \\
x1_i \leftarrow x_i + dx \\
pr_i \leftarrow \dfrac{Func(x1) - Func(x)}{dx}
\end{array}\right. \\
\text{return } pr
\end{array}\right.
$$

функція norm використовується для отримання нормалізованого вектора градієнту

$$
norm(x,n) := \left|\begin{array}{l}
s \leftarrow 0 \\
pr \leftarrow grad(x,n) \\
\text{for } i \in 0..n-1 \\
\quad s \leftarrow s + \left|pr_i\right|^2 \\
\text{return } \sqrt{s}
\end{array}\right.
$$

функція scan використовується для отримання оптимального кроку h_j методом сканування

$$
scan(x,s) := \left|\begin{array}{l}
x1 \leftarrow gr\_h_0 \\
F1 \leftarrow Func(x + x1 \cdot s) \\
x2 \leftarrow gr\_h_0 \\
\text{for } i \in 1..\dfrac{\left|gr\_h_1 - gr\_h_0\right|}{step\_odn} \\
\quad \left|\begin{array}{l}
x2 \leftarrow x2 + step\_odn \\
F \leftarrow Func(x + x2 \cdot s) \\
\text{if } F < F1 \\
\quad \left|\begin{array}{l}
F1 \leftarrow F \\
x1 \leftarrow x2
\end{array}\right.
\end{array}\right. \\
\text{return } x1
\end{array}\right.
$$

```
F_Rivsa(x,e,n) := | k ← 1
                  | steps ← 0
                  | for i ∈ 0.. n − 1
                  |   XX_i ← x
                  | while 1
                  |   | j ← 0
                  |   | F_j ← Func(XX_j)
                  |   | S_j ← −grad(XX_j,n)
                  |   | while 1
                  |   |   | if norm(XX_j,n) > e
                  |   |   |   | h_j ← scan(XX_j,S_j)
                  |   |   |   | step_steps ← h_j
                  |   |   |   | steps ← steps + 1
                  |   |   |   | XX_{j+1} ← XX_j + h_j · S_j
                  |   |   |   | F_{j+1} ← Func(XX_{j+1})
                  |   |   |   | if j ≥ n − 1
                  |   |   |   |   | k ← k + 1
                  |   |   |   |   | XX_0 ← XX_{j+1}
                  |   |   |   |   | break
                  |   |   |   | otherwise
                  |   |   |   |   | ■
                  |   |   |   |   | ■
                  |   |   |   |   | ■
                  |   |   | return ( XX_j ; Func(XX_j) ; step^T ; k ) otherwise
```

Rez := F_Rivsa(x,ε,n)

$$Arg := Rez_0$$
$$Funct := Rez_1$$
$$h := Rez_2$$
$$Iter := Rez_3$$

Розв'язок за допомогою методу спряженого градієнту Флетчера - Рівса:

Масив оптимальних кроків:

h =

	0	1	2	3	4	5	6	7	8	9
0	0.13	0.48	0.12	0.14	0.4	0.1	0.1	0.1	0.1	0.1

Мінімум функції на заданому інтервалі знаходиться у точці:

$$Arg = \begin{pmatrix} 4.999 \\ 6 \end{pmatrix}$$

Значення мінімуму функції:

$$Funct = 1.211 \times 10^{-6}$$

Кількість ітерацій:

$$Iter = 7$$

Оптимальний розв'язок:

$$M := Minimize(Func, x)$$
$$M = \begin{pmatrix} 5 \\ 6 \end{pmatrix}$$
$$Fmin := Func(M)$$
$$Fmin = 0$$

Лабораторна робота № 4

Методи нелінійного програмування при наявності обмежень

При рішенні задач нелінійного програмування з обмеженням зустрічаються більші труднощі, ніж при рішенні задач безумовної оптимізації за тієї причини, що рішення, яке потрібно віднайти повинно відповідати додатковим вимогам, тобто задовольняти умовам задачі. Для рішення задачі нелінійного програмування. що має обмеження, часто застосовують один з наступних підходів:

1) перебудова задачі нелінійного програмування з обмеженням в еквівалентну тій послідовність задач безумовної оптимізації за допомогою штрафних функцій.

2) використання методів лінійного програмування для рішення задач нелінійного програмування з застосуванням процедури послідовної лінійної апроксимації [15].

3) використання методу ковзного допуску, що дозволяє покращити значення цільової функції [13].

Методи штрафних функцій

В основу методів штрафних функцій в області нелінійного програмування покладена ідея перебудови загальної нелінійної задачі в послідовність задач без обмежень шляхом додавання до цільової функції однієї або кількох функцій, які встановлюють обмеження для того, щоб обмеження, як такі, в задачі оптимізації не фігурували. В цьому випадку мінімізація може здійснюватись за допомогою більш простих алгоритмів. При використанні методів штрафних функцій отримується оптимальний ефект за рахунок постійного компромісу між необхідністю задоволення обмежень і процесом мінімізації f(x), який досягається шляхом присвоєння належної ваги цільової функції і функціям, які визначають обмеження. Методи штрафних функцій можна поділити на 2 класи: параметричні і непараметричні методи. Параметричні методи характеризуються наявністю одного чи кількох певним чином підібраних параметрів, які входять у склад (структуру) штрафної функції як вагових коефіцієнтів. Параметричні методи розпадаються на 3 категорії: методи внутрішньої точки; методи зовнішньої точки; комбіновані методи. При використанні методів внутрішньої точки рівень цільової функції утримується вдалині від межі допустимої області (тобто точка $X^{(k)}$ постійно знаходиться всередині допустимої області) за допомогою штрафної функції. Методи зовнішньої точки, навпаки, генерують послідовність точок, які виходять за межі допустимої області, але дають в межах допустимі рішення. Штрафна функція не дозволяє вектору X надто відходити від межі допустимої області. Формально перебудова загальної задачі НП, в задачу мінімізації обмежень проводиться шляхом переходу до задачі мінімізації:

$$P\left(X^{(k)}, P^{(k)}\right) = f\left(X^{(k)}\right) + \sum_{i=1}^{m} \rho_i^{(k)} H\left(h_i\left(X^{(k)}\right)\right) + \sum_{i=m+1}^{p} \rho_i^{(k)} G\left(g_i\left(X^{(k)}\right)\right), \tag{4.1}$$

де $P(X(k), \rho(k))$ – узагальнена приєднана функція чи штрафна функція; $\rho_i^{(k)} \geq 0$ – вагові коефіцієнти; K – кількість завершених етапів обчислювального оптимізаційного процесу. При будь-якому виборі функціоналів $H(h_i(X))$ і $G(g_i(X))$ потрібно, щоб:

$$\left.\begin{aligned} &\lim_{k\to\infty} \sum_{i=m+1}^{p} \rho_i^{(k)} G\left(g_i\left(X^{(k)}\right)\right) = 0 \\ &\lim \sum_{i=1}^{m} \rho_i^{(k)} H\left(h_i\left(X^{(k)}\right)\right) = 0 \\ &\lim \left| p\left(X^{(k)}, \rho^{(k)}\right) - f\left(X^{(k)}\right) \right| = 0 \end{aligned}\right\} \tag{4.2}$$

По мірі розвитку процесу оптимального пошуку вплив, що входять в $P(X(k), \rho(k))$ функцій обмежень на значення даної приєднаної функції постійно слабне, а в межі повністю зникає. Значить, екстремум $P(X)$ співпадає з екстремумом $f(X)$.

Основні типи штрафів

Для обліку обмежень-рівностей часто використовують квадратичний штраф (рис.4.1):

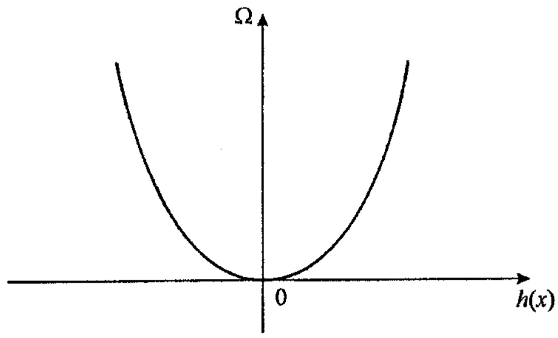
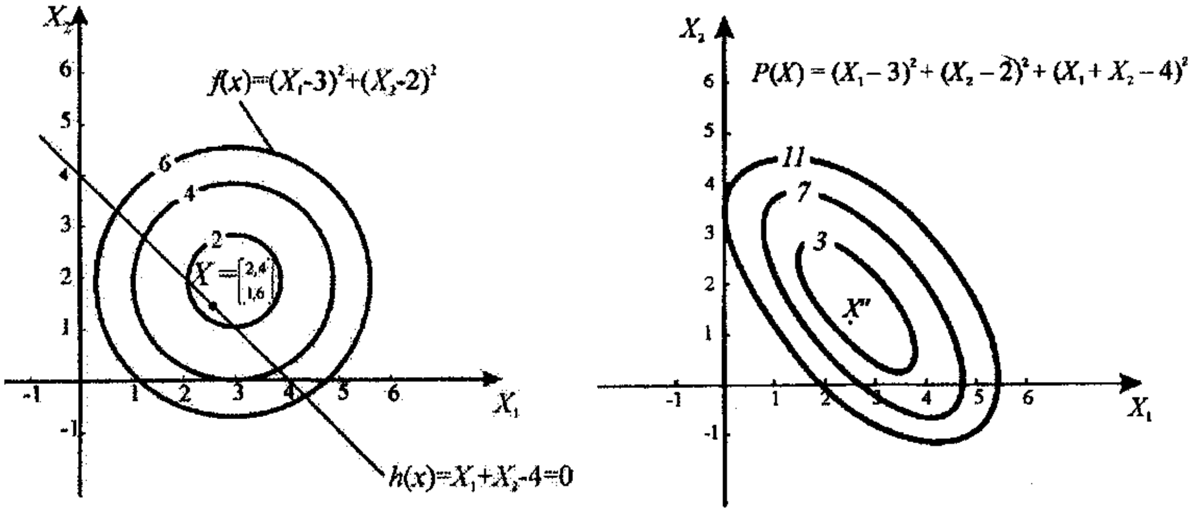


Рис.4.1. Квадратичний штраф

При мінімізації цей штраф запобігає відхиленню значення $h(X)$ від нуля. На рис.4.2 графічно показана штрафна функція, отримана шляхом додавання до $f(X)$ квадратичним штрафом (з ваговим коефіцієнтом, що дорівнює одиниці).



а б

Рис.4.2. Рівні цільової (а) та штрафної функції (б)

При врахуванні обмежень-нерівностей використовують різні типи штрафів. Простішим серед них є нескінченний бар’єр, показаний на рис.4.3.

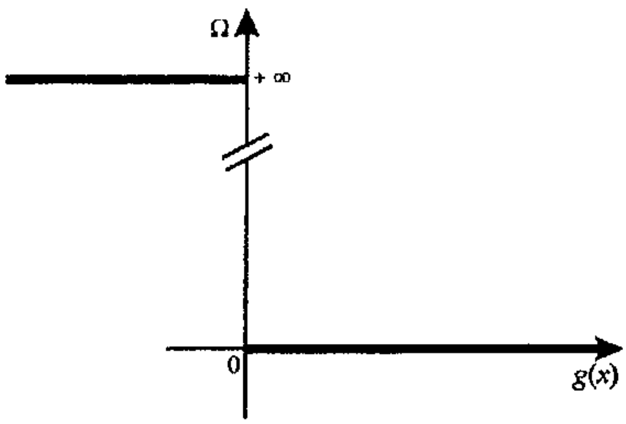


Рис.4.3. Нескінчений штраф

Відповідне визначення приймає нескінченно великі значення в недопустимих точках і нульове значення - в допустимих. У даному випадку штрафна функція $P(X, \rho)$ – розривна на межі допустимої області. В машинній реалізації нескінченних штрафів використовують велике додатне число. Наприклад, цей штраф використовується у формулі

$$\Omega = 10^{20} \sum_{j=g} \left| g_i \;\; X \; \right|. \qquad (4.3)$$

де g – множина індексів порушень обмежень, $g_j(X)<0$ при $j \in g$. Другим широко використовуваним типом штрафу є логарифмічний (рис.4.4).

$$\Omega = -\rho \ln\left[g(X) \right]. \tag{4.4}$$

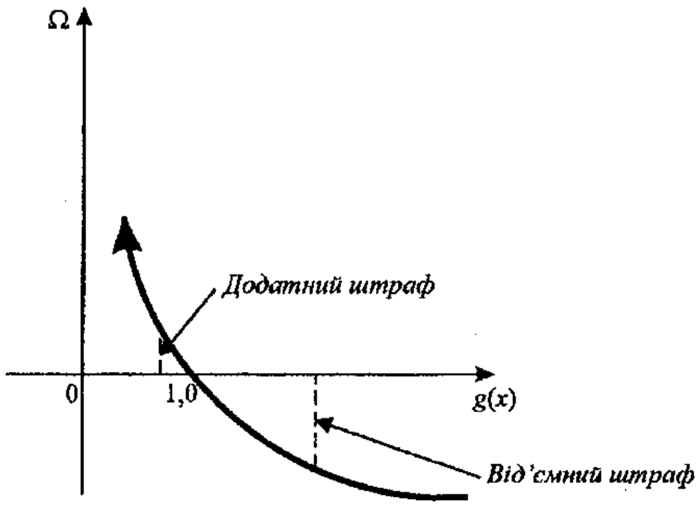


Рис.4.4. Логарифмічний штраф

Цей штраф додатній при всіх X таких, що $0 < g(X) < 1$ і від'ємний при $g(X) > 1$. Логарифмічний штраф – бар'єрна функція, не визначена в недопустимих точках (тобто для X таких, що $g(X) < 0$). Ітераційний процес починається з допустимої точки початкової при додатному значенні ρ. Після рішення кожної підзадачі безумовної мінімізації параметр ρ зменшується і в межах прагне до нуля. Штраф, заданий зворотною функцією рис.4.5.

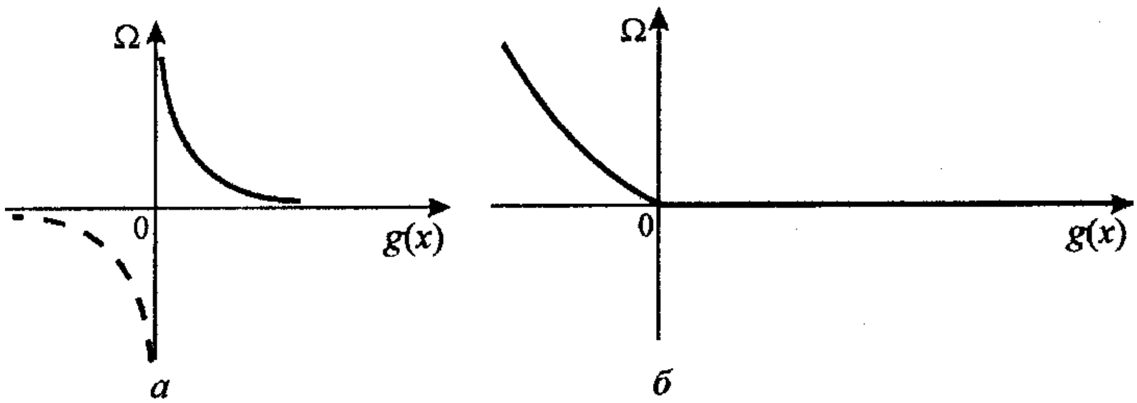


Рис.4.5. Штраф, заданий зворотною функцією

$$\Omega = \rho\left[\frac{1}{g(X)} \right] \tag{4.5}$$

Штраф, заданий зворотною функцією, не має від'ємних значень в допустимій області. Цей штраф є бар'єром; можливі труднощі, які пов'язані з появою недопустимих точок.

Метод поступової безумовної мінімізації

Алгоритм методу поступової безумовної мінімізації (МПБМ), розвинутий Фіакко і Мак-Корміком, який застосовується для рішення нелінійного програмування в який $f(X)$ і $g(X)$ (i = 1, ..., m) можуть бути нелінійними функціями нелінійних змінних, а $h_i(X)$ (i = 1, ..., m) повинні бути лінійними функціями незалежних змінних. При таких умовах гарантується збіжність послідовності проміжних рішень до оптимального рішення задачі нелінійного програмування. Метод МПБМ в основному приводиться до рішення деякої послідовності задач без обмежень, причому в межах знаходиться мінімум початкової вихідної задачі нелінійного програмування. В варіанті метода 1967 р. задачу перебудовують в послідовність задач без обмежень, до речі в межах знаходиться мінімум початкової вихідної задачі

$$P\left(X^{k}, r^{k}\right) = f\left(X^{k}\right) + \left(r^{k}\right)^{-\frac{1}{2}} \sum_{i=1}^{m} h_i^2\left(X^{k}\right) + r^{k} \sum_{i=m+1}^{\rho} \frac{1}{g_i\left(X^{k}\right)}, \tag{4.6}$$

де значення вагових коефіцієнтів r додатні і утворюють одноманітно спадну послідовність $\{r \mid r^{(0)} > r^{(1)} > ... > 0\}$.

У варіанті МПБМ 1970 р. застосовується логарифмічний штраф

$$P\left(X^{k}, r^{k}\right) = f\left(X^{k}\right) + \frac{1}{r^{k}} \sum_{i=1}^{m} h_i^2\left(X^{k}\right) - r^{k} \sum_{i=m+1}^{\rho} \ln g_i\left(X^{k}\right), \tag{4.7}$$

Як і у варіанті (4.7) тут використовується квадратичній штраф для обліку обмежень-рівностей. Процедура мінімізації функцій (4.7) і (4.8) починається з внутрішньої (або граничної) точки, тобто з точки $X^{(0)}$, в якій всі граничні умови у вигляді нерівностей задоволені. Після визначення $r(0)$ точка $X^{(1)}$ розраховується мінімізацією $P(X, r^{(0)})$. Потім визначається $r^{(1)}$ і знаходиться $X^{(2)}$ і т.д. Швидкість збіжність залежить від початкового вибору $r^{(0)}$ і від способу редагування даного параметру. Одним серед засобів вибору початкового значення r(0), який було запропоновано Фіакко і Мак-Корміком, є $r^{(0)} = 1$. При цьому вказано, що ефективність алгоритму не зміниться значно, якщо послідовність $r^{(1)}$, $r^{(2)}$, ..., $r^{(k)}$ ініціювати простим співвідношенням

$$r^{k} = \frac{r^{k-1}}{C},$$

де $C > 1$ є константою (завжди вважають $C = 4$). На рис.6.6 показано схему методу.

Завдання:

1. Мінімізувати $f(X)=100(X_2-X_1^2)^2+(1-X_1)^2$ при обмеженнях:

$g_1(X) = X_1 + 1 \geq 0;$
$g_2(X) = 1 - X_2 \geq 0;$
$g_3(X) = 4X_2 - X_1 - 1 \geq 0;$
$g_4(X) = 1 - 0.5X_1 - X_2 \geq 0;$

Записати для цієї задачі штрафні функції, використовуючи логарифмічний штраф; штраф, який задано оберненою функцією. Дослідити кожну із функцій у точках $X^{(0)} = [-1\ \ 1]^T$ та $[-0.5\ 0.5]^T$, звернувши увагу на труднощі, що зустрічаються. Реалізувати пошук, використавши метод змінної метрики.

2. Мінімізувати функцію $f(X)=-X_1^2-X_2^2$ при обмеженнях

$X_1 \geq 0;$
$X_2 \geq 0;$
$X_1 + 2X_2 \leq 3.$

3. Використовуючи МПБМ, мінімізувати функцію $f(X)=\dfrac{4}{X_1}+\dfrac{9}{X_2}+(X_1+X_2)$ при обмеженнях

$X_1 \geq 0;$
$X_2 \geq 0;$
$X_1 + 2X_2 \leq 4.$

4. Мінімізувати функцію $f(X)=X_1^2+X_2^2+X_3^2$ при обмеженнях

$X_1 + X_2 + X_3 \geq 3;$
$X_1X_2X_3 \geq 3;$
$X_1 > 0;$
$X_2 \geq 0;$
$X_3 \geq 0;$

5. Мінімізувати функцію $f(X)=(X_1-X_2)^2+(X_2-1)^2$ при обмеженнях

$$g_1(X)=\frac{X_1^2}{4}+X_2^2+1\geq 0,$$

$$h_1(X)=X_1-2X_2+1=0$$

а) Побудувати функцію $P(X, r)$ при $r = 0{,}04$. Зобразити побудовану P – функцію графічно;

б) Визначити напрямок оптимізаційного пошуку з початкової внутрішньої точки $X^{(0)} = [0{,}748\ 0{,}548]^T$, за допомогою методу градієнту;

в) Знайти вектор $X^{(1)}$;

г) Чи може одна з точок $X^{(k)}$ бути поза допустимої області?

д) Реалізувати пошук за допомогою методів змінної метрики.

Таким чином, ці методичні вказівки є введенням в теорію оптимізації і знайомить з додатком цієї теорії до рішення ряду задач. Практичний інтерес при розробці оптимальних систем керування представляють методи нелінійного програмування. Оскільки розмірність інженерних задач, як правило, достатньо велика, розрахунки у відповідності з алгоритмами оптимізації потребують значних витрат часу, оптимізаційні методи орієнтовані головним чином на реалізацію за допомогою ЕОМ.

Список рекомендованої літератури

1. *Балакирев В.С, Володин В.М., Цирлин А.М.* Оптимальное управление процессами химической технологи. – М.: Химия, 1978. -383с.
2. *Цирлин А. М.* Оптимальное управление технологическими процессами. – М.: Энергоатомиздат, 1986. - 400с.
3. *Бояринов А. И., Кафаров В. В.* Методы оптимизации химической технологии. - М.: Химия, 1975. - 576с.
4. *Банди Б.* Методы оптимизации. Вводный курс. – М.: Радио и связь, 1988. - 128с.
5. *Поляк Б. Т.* Введение в оптимизацию. – М.: Наука, 1983. - 384с.
6. *Реклейтис Г., Рейвиндран А., Рэгсдел К.* Оптимизация в технике: В 2 кн. – М.: Мир, 1986. - 670с.
7. *УальдД. Д.* Методы поиска экстремума. - М.: Наука, 1967. - 218 с.
8. *Химмельблау Д.* Прикладное нелинейное программирование. – М.: Мир, 1975.-534с.
9. *Ладієва Л. Р.* Оптимальне керування системами. – К.: НМЦВО, 2000. - 187с.
10. *Островский Г.М., Бережинский Т.А.* Оптимизация химико-технологических процессов. Теория и практика. – М.: Химия, 1984.-240 с.
11. *Пападимитриу Х., Стайглиц К.* Комбинаторная оптимизация. Алгоритмы и сложность. – М.: Мир, 1985. - 512с.

ЗМІСТ

Навчально-методичне видання

Методи нелінійного програмування

МЕТОДИЧНІ ВКАЗІВКИ
ДО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ
з курсу «Оптимізація технологічних процесів та систем керування»для студентів спеціальності "Автоматизоване управління технологічними процесами" напряму "Автоматизація та комп'ютерно-інтегровані технологічні комплекси"

Укладачі: А.І. Жученко, д.т.н., проф.
Л. Р. Ладієва, к. т. н., доц.
О.В. Снігур