

МIНIСТЕРСТВО ОСВIТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**Лабораторна робота №** 1

з дисципліни “ Математичні методи оптимізації ”

тема “Знаходження безумовних екстремумів функцій багатьох змінних”

|  |  |  |
| --- | --- | --- |
| Виконав  студент VI курсу  групи КВ-64М  Подольський Сергій Валентинович  (*прізвище, ім’я, по батькові*)  варіант № 1 |  | Умовно зарахована  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладачем  Онай Микола Володимирович  (*прізвище, ім’я, по батькові*) |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Штрафні бали:   |  |  | | --- | --- | | **Термін здачі** | **Оформлення звіту** | |  |  | | Нараховані бали:   |  |  |  | | --- | --- | --- | | **Корект. виконання завд. (3 бала)** | **Відп. на теор. питання (4 бала)** | **Відп. на прогр. питання (2 бала)** | |  |  |  | | Сумарний бал:   |  | | --- | |  | |

Київ 2011

# Постановка задачі за варіантом

1. Знайти всі локальні екстремуми функції

на інтервалі , побудувати її графік та відмітити на ньому всі знайдені екстремуми. Якщо завдання буде виконане у *MatLab*, то для пошуку локального екстремуму рекомендується використовувати функцію *fminbnd*.

1. Знайти всі локальні екстремуми функції

на інтервалі ; та побудувати її графік. Якщо завдання буде виконане у *MatLab*, то для пошуку екстремумів рекомендується використовувати функцію *fmincon*, для побудови графіку функцію *surf* та для побудови сітки, необхідної для функції *surf*, функцію *meshgrid*.

1. Зробити висновки.

# Математичне підґрунтя для виконання даної лабораторної роботи

**Теорема 1 (Необхідні умови першого порядку).** Якщо – точка локального екстремуму диференційованої в точці функції , то всi частиннi похiднi функцiї f дорiвнюють нулю в точці :

**Теорема 2 (Необхідні умови другого порядку).** Якщо – точка локального мінімуму диференційованої два рази в точці функції , то виконується умова

Ця умова означає, що матриця

невід’ємно визначена.

**Теорема 3 (Достатні умови другого порядку).** Нехай функція

диференційована два рази в точці і виконуються умови:

Тоді – точка локального мінімуму задачі на екстремум

Друга умова теореми означає, що матриця

додатньо визначена.

# Графік першої, заданої за варіантом, функції з відміченими на ньому локальними екстремумами

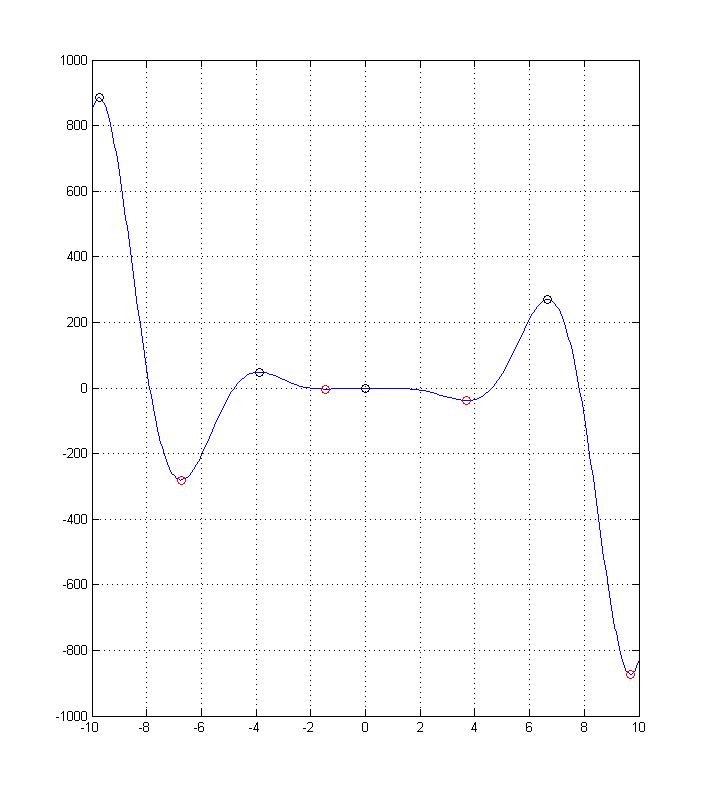


Рис.  1. Графік першої, заданої за варіантом, функції з відміченими на ньому локальними екстремумами

# Перелік всіх знайдених екстремальних точок та відповідних значень першої, заданої за варіантом, функції

xmin =

-6.7407 -1.4548 3.7098 9.7045

ymin =

-280.7327 -3.2464 -39.0412 -873.0622

xmax =

-9.7433 -3.8930 0 6.6657

ymax =

884.5268 48.4284 0 269.7871

# Графік другої, заданої за варіантом, функції

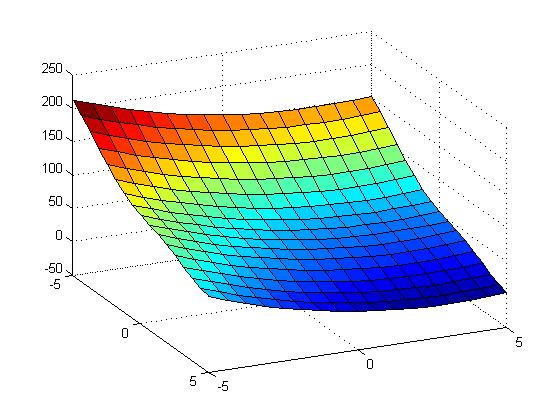


Рис.  2. Графік другої, заданої за варіантом, функції

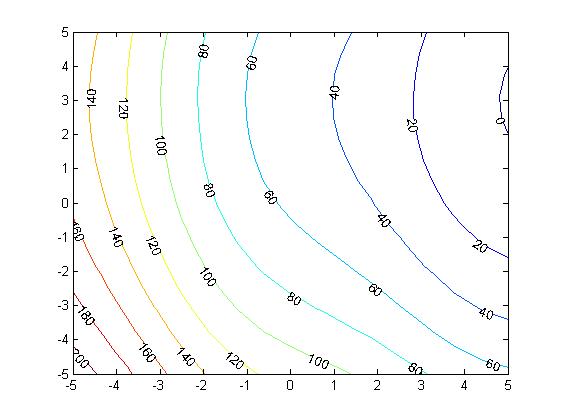


Рис.  3. Лінії рівня другої, заданої за варіантом, функції

# Перелік всіх знайдених екстремальних точок та відповідних значень другої, заданої за варіантом, функції

xmin =

5.0000

3.0000

fmin =

-1.0783

xmax =

-5

-5

fmax =

212.5110

# Висновки

Оскільки функція *fminbnd* знаходить лише один локальний мінімум, то для пошуку всіх локальних мінімумів було написано і використано рекурсивну функцію allmin, що знаходить і повертає у порядку за неспаданням усі значення аргументу та відповідні значення локальних мінімумів функції, заданої за варіантом. Дана рекурсивна функція використовує стандартну функцію fminsearch для пошуку мінімуму заданої функції з початковим наближенням. Для пошуку всіх максимумів була написана функція allmax, яка використовує функцію allmin для пошуку мінімумів вихідної функції із протилежним знаком. Усі мінімуми та максимуми відмічені на графіку функції (рис. 1) автоматично за допомогою скрипта: локальні максимуми відмічені колами чорного кольору, мінімуми – червоного.

Друга функція, задана за варіантом, містить лише один мінімум на максимум, які лежать на границях обмежень, що спостерігається також на графіку (рис. 2), а також більш чітко візуалізується лініями рівня (рис. 3).

# Додаток 1: MATLAB-функція знаходження всіх мінімумів функції

function [xmin, ymin] = allmin (fun, a, b)

% Поиск локальных минимумов fun внутри заданного интервала (а, Ь)

% Использование [xmin, ymin] = allmin(fun, a, b)

% задание погрешности для проверки условия попадания точки внутрь (а, Ь)

e = 1e-4\*(b - a);

% вызов рекурсивной подфункции

xmin = loc\_min(fun, a, b, e, 1);

% сортировка точек минимума в порядке возрастания

xmin = sort(xmin);

% вычисление значений функции в точках локального минимума

ymin = feval(fun, xmin);

function xmin = loc\_min(fun, a , b, e, new\_set)

% Рекурсивная подфункция для поиска всех локальных минимумов

% Начальная инициализация для накапливаемых результатов

persistent n pnt

if new\_set ~= 0

pnt = [ ];

n = 0;

end;

% задание начального приближения

c = 0.5\*(a + b);

% поиск локального минимума на [а, b]

z = fminsearch(fun, c);

if and(z > a + e , z < b - e )

% найден новый локальный минимум увеличиваем счетчик

% и добавляем локальный минимум в массив

n = n + 1;

pnt(n) = z;

% поиск локальных минимумов на подынтервалах

loc\_min(fun, a, z, e, 0);

loc\_min(fun, z, b, e, 0);

end

% возвращаем массив найденных точек

xmin = pnt;

# Додаток 2: MATLAB-функція знаходження всіх максимумів функції

function [xmax, ymax] = allmax (fun, a, b)

mirror = @(x) -fun(x);

[xmax ymax] = allmin(mirror, a, b);

ymax = -ymax;

# Додаток 3: автоматизований MATLAB-скрипт виконання завдання №1

clc;

fun = @(x) x.^3 .\* cos(x) - 2 .\* x .\* sin(x);

a = -10;

b = 10;

[xmin ymin] = allmin(fun, a, b)

[xmax ymax] = allmax(fun, a, b)

% Plot

figure

% Plot function graph

fplot(fun,[a b])

hold on

grid on

% plot minima points as red circles

plot(xmin, ymin, 'ro')

% plot maxima circles as black circles

maxima = plot(xmax, ymax, 'ro');

set(maxima, 'Color', 'black')

# Додаток 4: автоматизований MATLAB-скрипт виконання завдання №2

clc

% Source function

fun = @(x1, x2) (x1-7).^2 + (x2-3).^2 + 5.\*sin(x1) - cos(x1);

% x1 bounds

a1 = -5;

b1 = 5;

% x2 bounds

a2 = -5;

b2 = 5;

% Count of nodes on grid per each dimension

nodes = 16;

% Create Mesh Grid

[X Y] = meshgrid(a1:(b1-a1)/nodes:b1, a2:(b2-a2)/nodes:b2);

Z = fun(X, Y);

% Plot countours (level lines)

[CMatr h] = contour(X, Y, Z);

clabel(CMatr, h)

% Plot surface

figure

surf(X, Y, Z)

% Supply a starting point and invoke an optimization routine

x0 = [0; 0]; % Starting guess at the solution

lb = [a1 a2]; % Lower bounds

ub = [b1 b2]; % Upper bounds

myfun = @(x) fun(x(1), x(2));

[xmin, fmin] = fmincon(myfun, x0, [], [], [], [], lb, ub)

[xmax, fmax] = fmincon(@(x) -myfun(x), x0, [], [], [], [], lb, ub);

% Print X of max

xmax

% Print function max value

fmax = - fmax