Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Рыбинский государственный авиационный технический университет

имени П.А. Соловьева»

Институт «Информационные технологии и системы управления»

Кафедра вычислительных систем

**РАСЧЕТНО-ГРАФИЧЕСКАЯ РАБОТА**

по дисциплине:

«Методы оптимизации»

Вариант № 5

Студент группы ПИМ-24 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ананьев Г.Е.

*(Код)                                   (Подпись, дата)                                    (Фамилия И. О.)*

Руководитель к.т.н., проф. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Вишняков В.А.

*(Уч. степень, звание)                          (Подпись, дата)                                    (Фамилия И. О.)*

Рыбинск 2025

Содержание

[Задача 1. Найти экстремум функции 3](#_Toc194770967)

[Задача 2 Найти максимум функции 5](#_Toc194770968)

[Задача 3 Составить алгоритм определения и вычислить минимум функции 7](#_Toc194770969)

[Приложение А. Исходный код программы для решения задачи методом покоординатного движения Гаусса-Зейделя 9](#_Toc194770970)

# Задача 1. Найти экстремум функции

Для решения задачи воспользуемся методом дифференциального исчисления безусловного экстремума.

Требуется найти значение оптимального вектора xо = (x1о, x2о).

Необходимое условие экстремума определяется системой уравнений:

Решим её:

Из второго уравнения следует, что

Подставляя это значение в первое уравнение, получаем . Отсюда находим . Используя второе уравнение с учётом найденного , получим . Таким образом, .

Далее проверим достаточное условие экстремума:

Найдём вторую частную производную по x1:

Отсюда a11=8

Найдём смешанную производную второго порядка:

Отсюда a12=a21=­4

Найдём вторую частную производную по x2:

Отсюда a22=6

Таким образом матрица A принимает вид:

Вычислим главные миноры матрицы:

Все миноры матрицы являются положительными.

Ответ: в точке находится минимум. Минимальная величина функции равна .

Задача 2 Найти максимум функции

При условиях

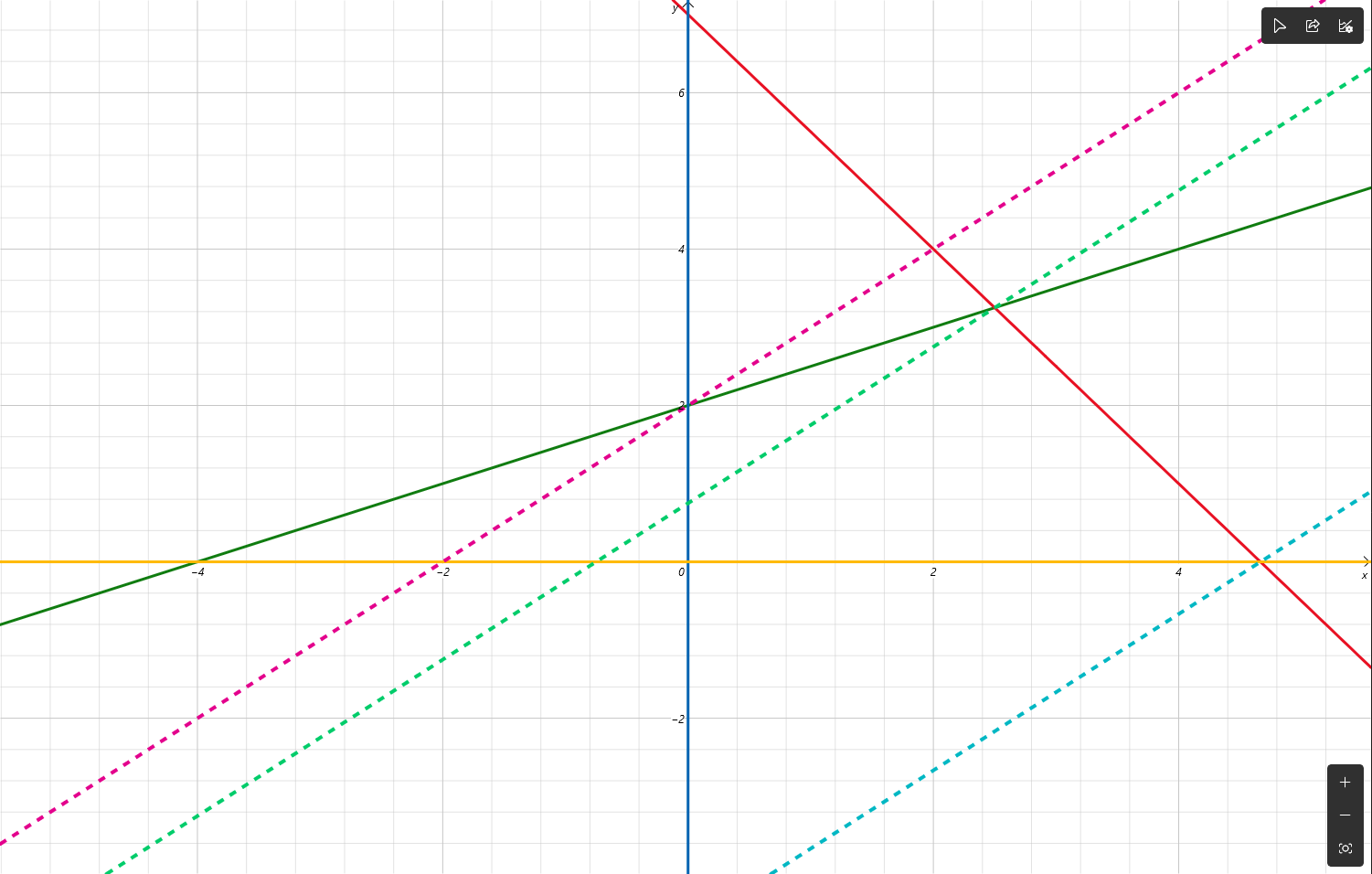
Воспользуемся методом линейного программирования для решения данной задачи.

Запишем ограничения в другом виде:

Область отыскания решения в данном случае представляет собой четырехугольник ABCD, изображённый на Рисунок 1 - График функций зад. 2.

Запишем целевую функцию в виде x2 = x1 – f(x). При функция проходит через точку D. С уменьшением f(x) прямая линия, определяемая функцией, будет перемещаться параллельно самой себе вверх. При f(x) = fmin(x)=2 прямая проходит через точку B. Решением задачи является значение функции f(x), при котором она проходит через точку D. То есть , , .

Ответ:



C

D

B

A

Рисунок 1 - График функций зад. 2

Задача 3 Составить алгоритм определения и вычислить минимум функции

методом Гаусса-Зейделя при движении из точки с координатами *x*1 = 0, *x*2 = 0.

Вычислим частные производные функции f(x):

Примем ε=0.1.

Для решения задачи была реализована программа, выполняющая итерации для нахождения решения.

Функция f(x) не ограничена снизу. Также отсутствуют ограничения на x. Соответственно, при некоторых размерах шага метод может не сойтись, т.к. функция будет бесконечно убывать.

Решение описано в виде таблицы, в которой i-ая строка содержит:

* x+- вектор, полученный добавлением Δxi к соответствующей шагу итерации переменной в векторе xi-1
* x- - вектор, полученный вычитанием Δxi из соответствующей шагу итерации переменной в векторе xi-1
* df – значение частной производной f(x) по соответствующей шагу итерации переменной в точке xi-1
* Δx – шаг i-ой итерации. Вычисляется как |t \* df|
* f(x+) – значение функции в точке x+
* f(x-) – значение функции в точке x-
* x - выбранная на шаге i точка x. Выбирается как аргумент минимального значения функции f(x+) и f(x-)
* f(x) – значение функции в выбранной точке x

Решения, полученные с использованием программы представлены в Таблица 1 - Решение при t = 0.1, Таблица 2 - Решение при t=0.5, Таблица 3 - Решение при t=1

Таблица 1 - Решение при t = 0.1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | x+ | x- | df | Δx | f(x+) | f(x-) | x | f(x) |
| 0 |  |  |  |  |  |  | (0; 0) | -4 |
| 1 | (0.20; 0.00) | (-0.20; 0.00) | -2.00 | 0.20 | -4.39 | -3.61 | (0.20; 0.00) | -4.39 |
| 2 | (0.28; 0.20) | (-0.28; 0.20) | 2.80 | 0.28 | -3.53 | -5.10 | (0.20; -0.28) | -5.10 |
| 3 | (0.36; -0.28) | (0.04; -0.28) | -1.60 | 0.16 | -5.33 | -4.83 | (0.36; -0.28) | -5.33 |
| 4 | (-0.07; 0.36) | (-0.49; 0.36) | 2.08 | 0.21 | -4.86 | -5.72 | (0.36; -0.49) | -5.72 |
| 5 | (0.47; -0.49) | (0.25; -0.49) | -1.12 | 0.11 | -5.83 | -5.59 | (0.47; -0.49) | -5.83 |
| 6 | (-0.33; 0.47) | (-0.64; 0.47) | 1.55 | 0.16 | -5.57 | -6.05 | (0.47; -0.64) | -6.05 |
| 7 | (0.54; -0.64) | (0.40 ; -0.64) | -0.69 | 0.07 | -6.09 | -6.00 | (0.54 ; -0.64) | -6.09 |

Таблица 2 - Решение при t=0.5

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | x+ | x- | df | Δx | f(x+) | f(x-) | x | f(x) |
| 0 |  |  |  |  |  |  | (0; 0) | -4 |
| 1 | (1.00 ; 0.00) | (-1.00; 0.00) | -2.00 | 1.00 | -5.00 | -3.00 | (1.00 ; 0.00) | -5.00 |
| 2 | (1.00 ; 1.00) | (-1.00; 1.00) | 2.00 | 1.00 | -2.00 | -6.00 | (1.00 ; -1.00) | -6.00 |
| 3 | (2.00 ; -1.00) | (0.00 ; -1.00) | 2.00 | 1.00 | 0.00 | -6.00 | (0.00 ; -1.00) | -6.00 |

Таблица 3 - Решение при t=1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | x+ | x- | df | Δx | f(x+) | f(x-) | x | f(x) |
| 0 |  |  |  |  |  |  | (0; 0) | -4 |
| 1 | (2.00 ; 0.00) | (-2.00; 0.00) | -2.00 | 2.00 | 0.00 | -8.00 | (-2.00; 0.00) | -8.00 |
| 2 | (5.00 ; -2.00) | (-5.00; -2.00) | 5.00 | 5.00 | 42.00 | -8.00 | (-2.00; -5.00) | -8.00 |

При t > 1 метод не сходится – функция бесконечно убывает.

Ответ: минимальное значение, полученное при решении задачи при

# Приложение А. Исходный код программы для решения задачи методом покоординатного движения Гаусса-Зейделя

#include <iostream>

#include <format>

typedef long long ll;

typedef long double ld;

using namespace std;

ld compute(ld x1, ld x2) {

return x1 \* x1 \* x1 - x1 \* x2 + x2 \* x2 - 2 \* x1 + 3 \* x2 - 4;

}

ld computeDerivative(ld x1, ld x2, ll i) {

if (i & 1) // Нечёт шаг - шаг по x1

return 3 \* x1 \* x1 - x2 - 2;

return -x1 + 2 \* x2 + 3;

}

int main() {

ld eps = 0.1;

ld dx = 0;

ld t = 1;

ld x1 = 0, x2 = 0;

ld fprev;

ld fcur = compute(x1, x2);

ld \*x1p = &x1;

ld \*x2p = &x2;

cout << format("{:<3} {:^20} {:^20} {:^7} {:^7} {:^7} {:^7} {:^20} {:^7}", "i ", "xp", "xm", "df", "dx", "f(xp)", "f(xm)", "x", "f(x)") << endl;

ll i = 1;

do {

fprev = fcur;

ld df = computeDerivative(x1, x2, i);

dx = abs(t \* df);

\*x1p += dx;

pair<ld, ld> x1x2p = {\*x1p, \*x2p};

ld fp = compute(x1, x2);

\*x1p -= 2 \* dx;

pair<ld, ld> x1x2m = {\*x1p, \*x2p};

ld fm = compute(x1, x2);

fcur = fm;

if (fp < fcur) {

\*x1p += 2 \* dx;

fcur = fp;

}

swap(x1p, x2p);

cout << format("{:<3} {:^20} {:^20} {:^7.2f} {:^7.2f} {:^7.2f} {:^7.2f} {:^20} {:^7.2f}",

i++, format("({:<5.2f}; {:<5.2f})", x1x2p.first, x1x2p.second), format("({:<5.2f}; {:<5.2f})", x1x2m.first, x1x2m.second), df,dx, fp, fm, format("({:<5.2f}; {:<5.2f})",x1, x2), fcur) << endl;

} while (abs(fcur - fprev) > eps);

return 0;

}