Цель работы: знакомство с оптимизационными задачами, изучение различных

методов одномерной оптимизации и сравнение эффективности их применения

для конкретных целевых функций.

**Задача №22 (Золотое сечение)**

Цель: Какой из прямоугольников с периметром 80 см. имеет наибольшую площадь? Вычислите площадь этого прямоугольника.

Методами: Золотое сечение;

Для решения этой задачи методом золотого сечения мы будем итеративно искать прямоугольник

с наибольшей площадью среди всех прямоугольников с периметром 80 см, изменяя одну из его

сторон. Начнем с широкого интервала возможных значений стороны и будем сужать его до достижения

оптимального значения площади

#include <iostream>

#include <cmath>

double calculateRectangleArea(double x, double y) {

return x \* y;

}

int main() {

double epsilon = 0.001; // Погрешность для остановки итераций

double a = 0.0; // Начальное значение для одной из сторон

double b = 40.0; // Начальное значение для другой стороны

double phi = (1 + sqrt(5)) / 2; // Золотое сечение

double x1 = b - (b - a) / phi; // Вычисляем первую точку внутри интервала

double x2 = a + (b - a) / phi; // Вычисляем вторую точку внутри интервала

double f1 = calculateRectangleArea(x1, 80 - 2 \* x1); // Площадь первого прямоугольника

double f2 = calculateRectangleArea(x2, 80 - 2 \* x2); // Площадь второго прямоугольника

while (fabs(b - a) > epsilon) { // Условие остановки (достаточно малая разница между границами)

if (f1 < f2) {

a = x1;

x1 = x2;

f1 = f2;

x2 = a + (b - a) / phi;

f2 = calculateRectangleArea(x2, 80 - 2 \* x2);

} else {

b = x2;

x2 = x1;

f2 = f1;

x1 = b - (b - a) / phi;

f1 = calculateRectangleArea(x1, 80 - 2 \* x1);

}

}

// Найденные значения сторон прямоугольника с наибольшей площадью

double length = (a + b) / 2;

double width = 80 - 2 \* length;

double maxArea = calculateRectangleArea(length, width);

std::cout << "Наибольшая площадь: " << maxArea << " квадратных сантиметров" << std::endl;

std::cout << "Длина: " << length << " см, Ширина: " << width << " см" << std::endl;

return 0;

}

Этот код использует метод золотого сечения для поиска прямоугольника с максимальной

площадью среди прямоугольников с периметром 80 см. Он итеративно сужает интервал

возможных значений сторон до достижения заданной погрешности и находит максимальную

площадь и соответствующие стороны прямоугольника.

**Задача №22 (Метод деление интервала пополам)**

Для решения этой задачи методом деления интервала пополам в C++, нам нужно сначала понять, как связаны периметр и площадь прямоугольника. Затем мы будем итеративно приближаться к прямоугольнику с наибольшей площадью, изменяя его стороны.

1. Определите формулу для площади прямоугольника в зависимости от его сторон: Пусть длина прямоугольника будет x см, а ширина будет (80 - 2x) см (поскольку периметр равен 80 см и состоит из двух длин и двух ширин). Площадь S прямоугольника равна S = x \* (80 - 2x).
2. Найдите производную S по x и приравняйте ее к нулю, чтобы найти точку экстремума. В данном случае, это будет точка максимума.
3. Решите уравнение для x. Найденное значение x будет являться длиной прямоугольника, а ширина будет (80 - 2x).
4. Рассчитайте площадь S для найденных значений x и (80 - 2x).
5. Сравните площади всех найденных прямоугольников и выберите прямоугольник с наибольшей площадью.

Вот C++ код, который реализует этот алгоритм:

#include <iostream>

#include <cmath>

double findMaxRectangleArea() {

double left = 0.0;

double right = 40.0; // Максимальная длина прямоугольника, так как периметр равен 80

while (fabs(right - left) > 1e-6) {

double x1 = (left \* 2 + right) / 3.0;

double x2 = (left + right \* 2) / 3.0;

double area1 = x1 \* (80 - 2 \* x1);

double area2 = x2 \* (80 - 2 \* x2);

if (area1 < area2) {

left = x1;

} else {

right = x2;

}

}

// Найденная длина x будет длиной прямоугольника с наибольшей площадью

double length = (left + right) / 2.0;

double width = 80 - 2 \* length;

double maxArea = length \* width;

return maxArea;

}

int main() {

double maxArea = findMaxRectangleArea();

std::cout << "Наибольшая площадь прямоугольника: " << maxArea << " кв. см." << std::endl;

return 0;

}

Этот код использует метод деления интервала пополам для поиска длины прямоугольника, который имеет наибольшую площадь среди всех прямоугольников с периметром 80 см. Результат будет выведен на экран.