Times New Roman

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра информатики и программирования

ЛАБОРАТОРНАЯ РАБОТА №2

Вариант №24

студента 3 курса 341 группы

направления 02.03.03 – Математическое обеспечение и администрирование

информационных систем

факультета компьютерных наук и информационных технологий

Филиппенко Дмитрия Александровича

к.ф.-м.н., доцент

Рогачко Е. С.

Саратов 2024

Введение

В данной работе рассматриваются методы оптимизации для нахождения минимума функций. Применяются следующие методы:

- Метод поразрядного поиска
- Метод дихотомии
- Метод золотого сечения
- Метод параболической аппроксимации

Задание 1

Постановка задачи

Требуется найти минимум функции $f(x) = 3x^2 - 2x - 2$ на интервале $x \in [-1, 1]$ с использованием методов поразрядного поиска, дихотомии и золотого сечения.

Метод поразрядного поиска

Метод поразрядного поиска последовательно уменьшает шаг и меняет направление поиска, чтобы найти точку минимума функции.

```
import math

def f(x):
    return 3 * x ** 2 - 2 * x - 2

a, b = -1, 1
eps = 10 ** (-9)
delta = (b - a) / 4  # начальныйшагдискретизации
x0 = a
f0 = f(x0)

while math.fabs(delta) > eps:  # покашагдискретизациибольшеэпсилон
    x1 = x0 + delta
    f1 = f(x1)
    if f0 <= f1 or a >= x0 or x0 >= b:
        delta *= -1 / 4  # уменьшаемшагименяемнаправление
    x0, f0 = x1, f1

print(f'x = {x0}\nf = {f0}')
```

Листинг 1: Метод поразрядного поиска

Результат работы:

$$x = 0.3333333320915699, \quad f(x) = -2.3333333333333335$$

Метод дихотомии

Метод дихотомии использует две пробные точки, близкие к середине интервала, чтобы последовательно сужать интервал поиска.

```
def f(x):
    return 3 * x ** 2 - 2 * x - 2

a, b = -1, 1
eps = 10 ** (-9)
eps_n = 1
delta = eps

while eps_n > eps:
    x1 = (a + b - delta) / 2
    x2 = (a + b + delta) / 2
    if f(x1) <= f(x2):
        b = x2  # сужаеминтервалдо   [a, x2]
    else:
        a = x1  # сужаеминтервалдо   [x1, b]
    eps_n = (b - a) / 2  # достигнутаяточность

x_min = (a + b) / 2
f_min = f(x_min)
print(f'x = {x_min}\nf = {f_min}')
```

Листинг 2: Метод дихотомии

Результат работы:

$$x = 0.33333328303046816, \quad f(x) = -2.333333333333326$$

Метод золотого сечения

Метод золотого сечения применяет пропорции золотого сечения для поиска минимального значения функции.

```
def f(x):
    return 3 * x ** 2 - 2 * x - 2

a, b = -1, 1
eps = 10 ** (-6)
tau = (5 ** 0.5 - 1) / 2  # коэффициентзолотогосечения

x1 = a + (3 - 5 ** 0.5) / 2 * (b - a)
x2 = a + (5 ** 0.5 - 1) / 2 * (b - a)
f1 = f(x1)
f2 = f(x2)
eps_n = (b - a) / 2

while eps_n > eps:
    if f1 <= f2:
```

```
        b = x2
        x2 = x1
        f2 = f1
        x1 = b - tau * (b - a)
        f1 = f(x1)
    else:
        a = x1
        x1 = x2
        f1 = f2
        x2 = a + tau * (b - a)
        f2 = f(x2)
    eps_n *= tau

x_min = (a + b) / 2
f_min = f(x_min)
print(f'x = {x_min}\nf = {f_min}')
```

Листинг 3: Метод золотого сечения

Результат работы:

$$x = 0.333333264898966, \quad f(x) = -2.3333333333333193$$

Задание 2

Постановка задачи

Найти минимум функции $f(x) = 3x^3 + 6x^2 + x + 1$ на интервале $x \in [-3, 3]$ с использованием метода параболической аппроксимации.

Метод параболической аппроксимации

Метод параболической аппроксимации строит параболу по трем точкам и находит вершину параболы как предполагаемый минимум функции.

```
import math

def f(x):
    return 3 * x ** 3 + 6 * (x ** 2) + x + 1

a, b = -3, 3
eps = 10 ** (-6)
x1, x2, x3 = a, (a + b) / 2, b
f1, f2, f3 = f(x1), f(x2), f(x3)

a0 = f1
a1 = (f2 - f1) / (x2 - x1)
a2 = 1 / (x3 - x2) * ((f3 - f1) / (x3 - x1) - (f2 - f1) / (x2 - x1))
x_min = 1 / 2 * (x1 + x2 - a1 / a2)
f_min = f(x_min)
delta = 1
```

```python
while math.fabs(delta) > eps:
    delta = x_min
    if f_min < f2:
        x1, x2, x3 = x2, x_min, x3
        f1, f2, f3 = f2, f_min, f3
    else:
        x1, x2, x3 = x1, x_min, x2
        f1, f2, f3 = f1, f_min, f2
    a0 = f1
    a1 = (f2 - f1) / (x2 - x1)
    a2 = 1 / (x3 - x2) * ((f3 - f1) / (x3 - x1) - (f2 - f1) / (x2 - x1))
    x_min = 1 / 2 * (x1 + x2 - a1 / a2)
    f_min = f(x_min)
    delta -= x_min

print(f'x = {x_min}\nf = {f_min}')
```

Листинг 4: Метод параболической аппроксимации

Результат работы:

$$x = -1.2440167337295778, \quad f(x) = 3.265811649490149$$

Заключение

В данной работе исследованы методы оптимизации для поиска минимума функций. Проведенные эксперименты показали, что все методы успешно находят минимальные значения на заданных интервалах.