Санкт-Петербургский национальный исследовательский университет ИТМО

Факультет Программной Инженерии и Компьютерной техники

Методы оптимизации

Практическая работа №3

Вариант №10

Выполнил:

Состанов Тимур Айратович

P3214

Преподаватель:
Селина Е. Н.

Санкт-Петербург
2024

Исходная функция:

$$f(x) = 5x^2 - 8x^{\frac{5}{4}} - 20x$$

Точность: $\varepsilon = 0.0001$

Промежуток: $a, b = [3, 3.5]$

Метод квадратичной аппроксимации:

Рабочие формулы:

$$\bar{x} = \frac{1}{2}\frac{({x_2}^2 - {x_3}^2)f_1 + ({x_3}^2 - {x_1}^2)f_2 + ({x_1}^2 - {x_2}^2)f_3}{(x_2 - x_1)f_1 + (x_3 - x_1)f_2 + (x_1 - x_2)f_3}$$

Код программы:

```
equation = input("Введите уравнение:\n")
left, right = map(float, input("Введите границы:\n").split())
accuracy = float(input("Введите точность:\n"))  # 0.0001
delta = (right - left) / 10


def equation_solve(x):
    formatted_equation = equation.replace('x', str(x))
    return eval(formatted_equation)


f_min, x_min, minimum_point = 0, 0, 0


def check(x_1, x_3, f_min, minimum_point, x_min):
    if (abs((f_min - equation_solve(minimum_point)) / equation_solve(minimum_point)) < 
accuracy and
            abs((x_min - minimum_point) / minimum_point) < accuracy):
        print("Функция достигает своего минимума в точке:", minimum_point)
        print("Значение функции в этой точке:", equation_solve(minimum_point))
    else:
        if x_1 < minimum_point < x_3:
            x_1 = min(x_min, minimum_point)
            x_2, x_3 = x_1 + delta, x_1 - delta
            f_1, f_2, f_3 = equation_solve(x_1), equation_solve(x_2), 
equation_solve(x_3)
            f_min = min(f_1, f_2, f_3)

            x_values = [x_1, x_2, x_3]
            x_min = x_values[[f_1, f_2, f_3].index(f_min)]
            try:
                minimum_point = (0.5 * (
                        (x_2 ** 2 - x_3 ** 2) * f_1 + (x_3 ** 2 - x_1 ** 2) * f_2 + 
(x_1 ** 2 - x_2 ** 2) * f_3)
                                 / ((x_2 - x_3) * f_1 + (x_3 - x_1) * f_2 + (x_1 - x_2) 
* f_3))
            except ZeroDivisionError:
                x_1 = x_min
                solver(x_1)
            check(x_1, f_min, minimum_point, x_min)
        else:
            x_1 = minimum_point
            solver(x_1)
```

```python
def solver(x_1):
    global f_min, x_min, minimum_point
    f_1 = equation_solve(x_1)

    x_2 = x_1 + delta
    f_2 = equation_solve(x_2)

    x_3 = x_1 - delta if f_2 >= f_1 else x_1 + 2 * delta

    f_3 = equation_solve(x_3)

    f_min = min(f_1, f_2, f_3)
    x_values = [x_1, x_2, x_3]
    x_min = x_values[[f_1, f_2, f_3].index(f_min)]

    try:
        minimum_point = (0.5 * ((x_2 ** 2 - x_3 ** 2) * f_1 + (x_3 ** 2 - x_1 ** 2) *
f_2 + (x_1 ** 2 - x_2 ** 2) * f_3)
                         / ((x_2 - x_3) * f_1 + (x_3 - x_1) * f_2 + (x_1 - x_2) * f_3))
    except ZeroDivisionError:
        x_1 = x_min
        solver(x_1)
    check(x_1, x_3, f_min, minimum_point, x_min)


x_1 = (right + left) / 2
solver(x_1)
```

Вывод программы:

```
Введите уравнение:
5 * x**2 - 8 * x**(5/4) - 20 * x
Введите границы:
3 3.5
Введите точность:
0.0001
Функция достигает своего минимума в точке: 3.3531995063724103
Значение функции в этой точке: -47.14489104832254
```

Проверка результата:

min(5 * x**2 - 8 * x**(5/4) - 20 * x)

NATURAL LANGUAGE | MATH INPUT | EXTENDED KEYBOARD | EXAMPLES | UPLOAD | RANDOM

Input interpretation

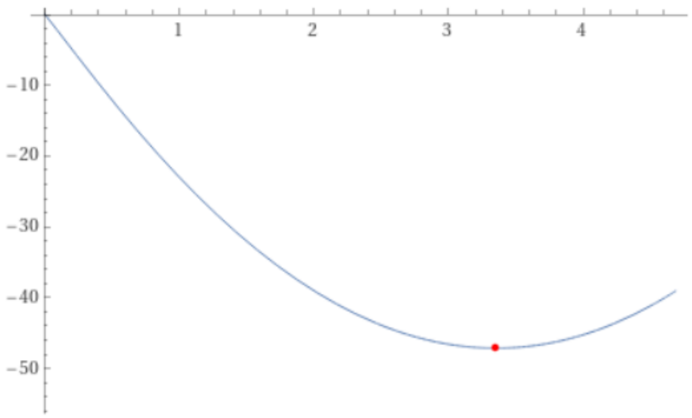
minimize	$5x^2 - 8x^{5/4} - 20x$

Global minimum

More digits | Exact form | Step-by-step solution

$\min\{5x^2 - 8x^{5/4} - 20x\} \approx -47.145$ at $x \approx 3.3532$

Plot



(x from −2.6 to 4.7)