Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет Программной инженерии и компьютерной техники

09.03.2024

Лабораторная работа №2
Методы оптимизации
Вариант 9

Раевский Григорий, группа

Содержание

Задание

Решить задачу четырьмя методами:методом половинного деления, методом золотого сечения, методом хорд и методом Ньютона. По 5 шагов каждого решения вручную + программа. Функция: $f(x) = \frac{1}{3}x^3 - 5x + x * \ln x$, $[a;b] = [1.5;2], \epsilon = 0.02$

Метод половинного деления

1 итерация

$x_1 = \frac{1.5+2-0.02}{2} = 1.74, x_2 = \frac{1.5+2+0.02}{2} = 1.76$

$y_1 = f(1.74) \approx -5.98023$ и $y_2 = f(1.76) \approx -5.98779$

Так как $y_1 > y_2$, то $a = 1.74$. Так как $2 - 1.74 = 0.26 > 2 * 0.02$, то переходим к следующей итерации.

2 итерация

$x_1 = \frac{1.74+2-0.02}{2} = 1.86, x_2 = \frac{1.74+2+0.02}{2} = 1.88$

$y_1 = f(1.86) \approx -6.00078$ и $y_2 = f(1.88) \approx -5.99832$

Так как $y_1 < y_2$, то $b = 1.88$. Так как $1.88 - 1.74 = 0.14 > 2 * 0.02$, то переходим к следующей итерации.

3 итерация

$x_1 = \frac{1.74+1.88-0.02}{2} = 1.8, x_2 = \frac{1.74+1.88+0.02}{2} = 1.82$

$y_1 = f(1.8) \approx -5.99798$ и $y_2 = f(1.82) \approx -6.00059$

Так как $y_1 > y_2$, то $a = 1.8$. Так как $1.88 - 1.8 = 0.08 > 2 * 0.02$, то переходим к следующей итерации.

4 итерация

$x_1 = \frac{1.8+1.88-0.02}{2} = 1.83, x_2 = \frac{1.8+1.88+0.02}{2} = 1.85$

$y_1 = f(1.83) \approx -6.00127$ и $y_2 = f(1.85) \approx -6.00136$

Так как $y_1 > y_2$, то $a = 1.83$. Так как $1.88 - 1.83 = 0.05 > 2 * 0.02$, то переходим к следующей итерации.

5 итерация

$x_1 = \frac{1.83+1.88-0.02}{2} = 1.845, x_2 = \frac{1.83+1.88+0.02}{2} = 1.865$

$y_1 = f(1.845) \approx -6.0015$ и $y_2 = f(1.865) \approx -6,00032$

Так как $y_1 < y_2$, то $b = 1.865$. Так как $1.865 - 1.83 = 0.035 < 2 * 0.04$, то следующей итерации не будет.

Финал

$x_m = \frac{1.83+1.865}{2} = 1.8475$ и $y_m = f(1.8475) \approx -6.00145$

Код

```python
import numpy as np
def func(x):
    result = (1 / 3) * (x ** 3) - 5 * x + x * np.log(x)
    return result
def solution(a, b, e):
    i = 1
    while b - a >= 2 * e:
        x1 = (a + b - e) / 2
        x2 = (a + b + e) / 2

        y1 = func(x1)
        y2 = func(x2)

        if y1 > y2:
            a = x1
        else:
            b = x2

        i += 1
    return a, b

# f(x) = 1/3 * x**3-5*x+x*ln(x) a = 1.5 b = 2 e =0.02

print("For f(x) = (1/3)*x^3-5*x+x*ln(x) enter a, b ([a,b]) and epsilon: ")

inp_a = float(input())
inp_b = float(input())
inp_e = float(input())
```

```

res_a, res_b = solution(inp_a,inp_b,inp_e)
xm = (res_a+res_b)/2

print("Result is: xm = ", round(xm,5), ", ym = ", round(func(xm),5))
```

Листинг 1: Python example

Входные данные для кода

```
1.5
2
0.02
```

Здесь $a = 1.5, b = 2, \epsilon = 0.02$

Результат работы программы

```
Result is: xm =  1.8475 , ym =  -6.00145
```

Метод золотого сечения

Начальные значения: $x_1 = 1.691, x_2 = 1.809$ и $y_1 = -5.955$ и $y_2 = -5.999$. Отметим, что $y_1 > y_2$.

1 итерация

Тогда $a = 1.69098, x_1 = x_2 = 1.80902, y_1 = y_2 = -5.99937$, а $x_2 = 1.691 + 0.618(2 - 1.691) = 1.88196$ и $y_2 = f(x_2) = -5.99799$. Текущий отрезок - [1.69098, 2] с длиной $0.30902 > 0.02$

$y_1 < y_2$.

2 итерация

Тогда $b = 1.88196, x_2 = x_1 = 1.80902, y_2 = y_1 = -5.99937$, а $x_1 = 1.882 - 0.618(1.882 - 1.691) = 1.76393$ и $y_1 = f(x_1) = -5.98908$. Текущий отрезок - [1.69098, 1.88196] с длиной $0.19098 > 0.02$.

$y_1 > y_2$

3 итерация

Тогда $a = 1.76393, x_1 = x_2 = 1.80902, y_1 = y_2 = -5.99937$, а $x_2 = 1.764 + 0.618(1.882 - 1.764) = 1.83688$ и $y_2 = f(x_2) = -6.0015$. Текущий отрезок - [1.76393, 1.88196] с длиной $0.11803 > 0.02$.

$y_1 > y_2$

4 итерация

Тогда $a = 1.80902, x_1 = x_2 = 1.83688, y_1 = y_2 = -6.0015$, а $x_2 = 1.809 + 0.618(1.882 - 1.809) = 1.8541$ и $y_2 = f(x_2) = -6.00107$. Текущий отрезок - [1.80902, 1.88196] с длиной $0.07294 > 0.02$.

$y_1 < y_2$

5 итерация

Тогда $b = 1.8541, x_2 = x_1 = 1.83688, y_2 = y_1 = -6.00107$, а $x_1 = 1.8541 + 0.618(1.809 - 1.8541) = 1.82624$ и $y_1 = f(x_1) = -6.00107$. Текущий отрезок - [1.80902, 1.8541] с длиной $0.04508 > 0.02$.

$y_1 > y_2$

Код

```python
import numpy as np

def func(x):
    result = (1 / 3) * (x ** 3) - 5 * x + x * np.log(x)
    return round(result,5)


def golden_ratio(a, b, e):
    phi = (1 + np.sqrt(5)) / 2

    x1 = round(b - (b - a) / phi,5)
    x2 = round(a + (b - a) / phi,5)
    y1 = func(x1)
    y2 = func(x2)
```

```python

    while abs(b - a) > e:

        if y1 < y2:
            b = x2
            x2 = x1
            y2 = y1
            x1 = round(b - (b - a) / phi,5)
            y1 = func(x1)
        else:
            a = x1
            x1 = x2
            y1 = y2
            x2 = round(a + (b - a) / phi,5)
            y2 = func(x2)

    return (a + b) / 2


#a = 1.5
#b = 2
#e = 0.02

# f(x) = 1/3 * x**3-5*x+x*ln(x) a = 1.5 b = 2 e =0.02

print("For f(x) = (1/3)*x^3-5*x+x*ln(x) enter a, b ([a,b]) and epsilon: ")

inp_a = float(input())
inp_b = float(input())
inp_e = float(input())

xm = golden_ratio(inp_a,inp_b,inp_e)

fm = func(xm)

print("Result is: xm = ",xm, ", ym = ", fm)
```

Листинг 2: Python example

Входные данные для кода

```
1.5
2
0.02
```

Результат работы программы

```
Result is: xm =  1.84549 , ym =  -6.00149
```

Метод хорд

$f'(x) = x^2 - 4 + \ln(x)$.

1 итерация

Пусть $\widetilde{x} = 1.5 - \frac{-1.34453}{-1.34453-0.69315}(1.5 - 2) \approx 1.82992$. Так как $f'(\widetilde{x}) = -0.04712 < 0$, то $a = 1.82992$, b = 2.

2 итерация

$\widetilde{x} = 1.84075$. Так как $f'(\widetilde{x}) = -0.00147 < 0.02$, что меньше заданной точности, то $\widetilde{x}$ можно считать точкой минимума.

Финал

$\widetilde{x} = 1.84075$ - точка минимума.

Код

```
import numpy as np

def func(x):
    return round((1 / 3) * x ** 3 - 5 * x + x * np.log(x),5)

def func_derivative(x):
    return round(np.log(x) + x ** 2 - 4,5)

def chord(a, b, e):
    x = a - (func_derivative(a)/(func_derivative(a)-func_derivative(b)))*(a-b)

    while(func_derivative(x) > e):
        if (func_derivative(x) < 0):
            a = x
        else:
            b = x

        x = a - (func_derivative(a) / (func_derivative(a) - func_derivative(b))) * (a - b)

    return round(x,5)

#inp_a = 1.5
#inp_b = 2
#inp_e = 0.02

# f(x) = 1/3 * x**3-5*x+x*ln(x) a = 1.5 b = 2 e =0.02

print("For f(x) = (1/3)*x^3-5*x+x*ln(x) enter a, b ([a,b]) and epsilon: ")

inp_a = float(input())
inp_b = float(input())
inp_e = float(input())

xm = chord(inp_a,inp_b,inp_e)

```

```python
fm = func(xm)

print("Result is: xm = ",xm, ", ym = ", fm)
```

Листинг 3: Python example

Входные данные для кода

```
1.5
2
0.02
```

Результат работы программы

```
Result is: xm =  1.82992 , ym =  -6.00127
```

Метон Ньютона

$f'(x) = x^2 + \ln x - 4, f''(x) = 2x + \frac{1}{x}$

За x_0 возьмем $\frac{a+b}{2}$, т. е. $x_0 = 1.75$

1 итерация

Проверяем условие $|f'(x)| \leq \epsilon$: $f(1.75) = -0.37788$ и $|-0.37788| \geq 0.02$, т. е. условие нарушается. Тогда $x = x_0 - \frac{f'(x_0)}{f''(x_0)} = 1.75 - \frac{-0.37788}{4.07143} \approx 1.84281$.

2 итерация

Проверяем условие $|f'(x)| \leq \epsilon$: $f(1.84281) = 0.00724$ и $|0.00724| \leq 0.02$, т. е. условие выполняется. Тогда $x_m = 1.84281$ - точка минимума, в которой функция равна $y_m = f(x_m) = -6.00153$.

Финал

$x_m = 1.84281, y_m = -6.00153$ - минимум.

Код

```python
import numpy as np

def func(x):
    result = (1 / 3) * (x ** 3) - 5 * x + x * np.log(x)
    return round(result,5)

def func_der(x):
    result = np.log(x)+x**2-4
    return round(result,5)

def func_der2(x):
    result = 2*x+(1/x)
    return round(result,5)

def newton(a,b,e):
    x0 = (a + b ) / 2
    while True:
        f_prime = func_der(x0)
        f_dprime = func_der2(x0)

        if(abs(f_prime) <= e):
            break

        x0 = round(x0 - (f_prime/f_dprime),5)

    return round(x0,7)

#a = 1.5
#b = 2
#e = 0.02

# f(x) = 1/3 * x**3-5*x+x*ln(x) a = 1.5 b = 2 e =0.02
```

```python

print("For f(x) = (1/3)*x^3-5*x+x*ln(x) enter a, b ([a,b]) and epsilon: ")

inp_a = float(input())
inp_b = float(input())
inp_e = float(input())

xm = newton(inp_a,inp_b,inp_e)

fm = func(xm)

print("Result is: xm = ",xm, ", ym = ", fm)
```

Листинг 4: Python example

Входные данные для кода

```
1.5
2
0.02
```

Результат работы программы

```
Result is: xm =  1.84281 , ym =  -6.00153
```

Выводы

В ходе лабораторной работы была исследована задача нахождения минимума функции $f(x) = \frac{1}{3}x^3 - 5x + x \ln x$ на интервале $[1.5; 2]$ с заданной точностью $\epsilon = 0.02$. Для решения этой задачи были применены четыре метода: метод половинного деления, метод золотого сечения, метод хорд и метод Ньютона.

Каждый из методов позволил приблизиться к точке минимума. Методы половинного деления и золотого сечения продемонстрировали схожие результаты, что обусловлено их свойствами работы в условиях заданной точности и выбора начального интервала. Метод хорд и метод Ньютона также показали эффективность в решении задачи.

Исходя из результатов, можно сделать вывод, что применение различных методов оптимизации позволяет эффективно решать задачи нахождения экстремумов функций. Каждый метод имеет свои особенности и область применения, что делает его удобным инструментом в определённых ситуациях.