

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	4
Метод половинного деления	4
1 итерация	4
2 итерация	4
3 итерация	4
4 итерация	4

5 итерация	4
Финал	5
Код	5
Входные данные для кода	6
Результат работы программы	6
Метод золотого сечения	6
1 итерация	6
2 итерация	6
3 итерация	7
4 итерация	7
5 итерация	7
Код	7
Входные данные для кода	9
Результат работы программы	9
Метод хорд	9
1 итерация	9
2 итерация	9
Финал	9
Код	9
Входные данные для кода	11
Результат работы программы	11
Метод Ньютона	11
1 итерация	11
2 итерация	11
Финал	11
Код	12
Входные данные для кода	13
Результат работы программы	13

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

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

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

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

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

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

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

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

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

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

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

$$y_1 = f(1.83) \approx -6.00127 \text{ и } y_2 = f(1.85) \approx -6.00137$$

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

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

$$y_1 = f(1.845) \approx -6.00150 \text{ и } y_2 = f(1.865) \approx -6.00032$$

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

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

Код

```

1  import numpy as np
2  def func(x):
3      result = (1 / 3) * (x ** 3) - 5 * x + x * np.log(x)
4      return result
5  def solution(a, b, e):
6      i = 1
7      while b - a >= 2 * e:
8          x1 = (a + b - e) / 2
9          x2 = (a + b + e) / 2
10
11         y1 = func(x1)
12         y2 = func(x2)
13
14         if y1 > y2:
15             a = x1
16         else:
17             b = x2
18
19         i += 1
20     return a, b
21
22 # f(x) = 1/3 * x**3-5*x+x*ln(x) a = 1.5 b = 2 e =0.02
23
24 print("For f(x) = (1/3)*x^3-5*x+x*ln(x) enter a, b ([a,b]) and epsilon: ")
25
26 inp_a = float(input())
27 inp_b = float(input())
28 inp_e = float(input())

```

```

29
30 res_a, res_b = solution(inp_a,inp_b,inp_e)
31 xm = (res_a+res_b)/2
32
33 print("Result is: xm = ", round(xm,5), ", ym = ", round(func(xm),5))

```

Листинг 1: Python example

Входные данные для кода

```

1 1.5
2 2
3 0.02

```

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

Результат работы программы

```

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

```

Метод золотого сечения

Начальные значения: $x_1 = 1.691$, $x_2 = 1.809$ и $y_1 = -5.9549$ и $y_2 = -5.9994$. Отметим, что $y_1 > y_2$.

1 итерация

Тогда $a = 1.691$, $x_1 = x_2 = 1.809$, $y_1 = y_2 = -5.9994$, а $x_2 = 1.691 + 0.618(2 - 1.691) = 1.882$ и $y_2 = f(x_2) = -5.998$.

Текущий отрезок - $[1.691, 2]$ с длиной $0.309 > 0.02$

$y_1 < y_2$.

2 итерация

Тогда $b = 1.882$, $x_2 = x_1 = 1.809$, $y_2 = y_1 = -5.9994$, а $x_1 = 1.882 - 0.618(1.882 - 1.691) = 1.764$ и $y_1 = f(x_1) = -5.9891$. Текущий отрезок - $[1.691, 1.882]$ с длиной $0.191 > 0.02$.

$$y_1 > y_2$$

3 итерация

Тогда $a = 1.764, x_1 = x_2 = 1.809, y_1 = y_2 = -5.9994$, а $x_2 = 1.764 + 0.618(1.882 - 1.764) = 1.837$ и $y_2 = f(x_2) = -6.0015$. Текущий отрезок - $[1.764, 1.882]$ с длиной $0.118 > 0.02$.

$$y_1 > y_2$$

4 итерация

Тогда $a = 1.809, x_1 = x_2 = 1.837, y_1 = y_2 = -6.0015$, а $x_2 = 1.809 + 0.618(1.882 - 1.809) = 1.854$ и $y_2 = f(x_2) = -6.0012$. Текущий отрезок - $[1.809, 1.882]$ с длиной $0.073 > 0.02$.

$$y_1 < y_2$$

5 итерация

Тогда $b = 1.8541, x_2 = x_1 = 1.8369, y_2 = y_1 = -6.0015$, а $x_1 = 1.8541 + 0.618(1.809 - 1.8541) = 1.8262$ и $y_1 = f(x_1) = -6.0011$. Текущий отрезок - $[1.809, 1.8541]$ с длиной $0.045 > 0.02$.

$$y_1 > y_2$$

Код

```

1  import numpy as np
2
3  def func(x):
4      result = (1 / 3) * (x ** 3) - 5 * x + x * np.log(x)
5      return round(result, 5)
6
7
8  def golden_ratio(a, b, e):
9      phi = (1 + np.sqrt(5)) / 2
10
11     x1 = round(b - (b - a) / phi, 5)
12     x2 = round(a + (b - a) / phi, 5)
13     y1 = func(x1)
14     y2 = func(x2)

```

```

15
16     while abs(b - a) > e:
17
18         if y1 < y2:
19             b = x2
20             x2 = x1
21             y2 = y1
22             x1 = round(b - (b - a) / phi,5)
23             y1 = func(x1)
24         else:
25             a = x1
26             x1 = x2
27             y1 = y2
28             x2 = round(a + (b - a) / phi,5)
29             y2 = func(x2)
30
31     return (a + b) / 2
32
33
34 #a = 1.5
35 #b = 2
36 #e = 0.02
37
38 # f(x) = 1/3 * x**3-5*x+x*ln(x) a = 1.5 b = 2 e =0.02
39
40 print("For f(x) = (1/3)*x^3-5*x+x*ln(x) enter a, b ([a,b]) and epsilon: ")
41
42 inp_a = float(input())
43 inp_b = float(input())
44 inp_e = float(input())
45
46 xm = golden_ratio(inp_a,inp_b,inp_e)
47
48 fm = func(xm)
49
50 print("Result is: xm = ",xm, ", ym = ", fm)

```

Листинг 2: Python example

Входные данные для кода

```
1 1.5
2 2
3 0.02
```

Результат работы программы

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

Метод хорд

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

1 итерация

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

2 итерация

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

Финал

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

Код

```

1 import numpy as np
2
3 def func(x):
4     return round((1 / 3) * x ** 3 - 5 * x + x * np.log(x),5)
5
6 def func_derivative(x):
7     return round(np.log(x) + x ** 2 - 4,5)
8
9 def chord(a, b, e):
10     x = a - (func_derivative(a)/(func_derivative(a)-func_derivative(b)))*(a-b)
11
12     while(func_derivative(x) > e):
13         if (func_derivative(x) < 0):
14             a = x
15         else:
16             b = x
17
18         x = a - (func_derivative(a) / (func_derivative(a) - func_derivative(b))) * (a - b)
19
20     return round(x,5)
21
22 #inp_a = 1.5
23 #inp_b = 2
24 #inp_e = 0.02
25
26 # f(x) = 1/3 * x**3-5*x+x*ln(x) a = 1.5 b = 2 e =0.02
27
28 print("For f(x) = (1/3)*x^3-5*x+x*ln(x) enter a, b ([a,b]) and epsilon: ")
29
30 inp_a = float(input())
31 inp_b = float(input())
32 inp_e = float(input())
33
34 xm = chord(inp_a,inp_b,inp_e)
35
36 fm = func(xm)

```

```

37
38 print("Result is: xm = ", xm, ", ym = ", fm)

```

Листинг 3: Python example

Входные данные для кода

```

1 1.5
2 2
3 0.02

```

Результат работы программы

```

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

```

Метод 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

```
1 import numpy as np
2
3 def func(x):
4     result = (1 / 3) * (x ** 3) - 5 * x + x * np.log(x)
5     return round(result,5)
6
7 def func_der(x):
8     result = np.log(x)+x**2-4
9     return round(result,5)
10
11 def func_der2(x):
12     result = 2*x+(1/x)
13     return round(result,5)
14
15 def newton(a,b,e):
16     x0 = (a + b) / 2
17     while True:
18         f_prime = func_der(x0)
19         f_dprime = func_der2(x0)
20
21         if(abs(f_prime) <= e):
22             break
23
24         x0 = round(x0 - (f_prime/f_dprime),5)
25
26     return round(x0,7)
27
28 #a = 1.5
29 #b = 2
30 #e = 0.02
31
32 # f(x) = 1/3 * x**3-5*x+x*ln(x) a = 1.5 b = 2 e =0.02
33
34 print("For f(x) = (1/3)*x^3-5*x+x*ln(x) enter a, b ([a,b]) and epsilon: ")
35
```

```

36 inp_a = float(input())
37 inp_b = float(input())
38 inp_e = float(input())
39
40 xm = newton(inp_a,inp_b,inp_e)
41
42 fm = func(xm)
43
44 print("Result is: xm = ",xm, ", ym = ", fm)

```

Листинг 4: Python example

Входные данные для кода

```

1 1.5
2 2
3 0.02

```

Результат работы программы

```

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

```

TODO

проверить метод золотого, везде к 5 знаку, больше вычислений везде. Везде одна точность после запятой (5 знаков). Выводы по работе. Больше инфы