

Федеральное государственное автономное образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»
Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №5
«Интерполяция функции»
по дисциплине «Вычислительная математика»

Вариант: 9

Преподаватель:
Машина Екатерина Алексеевна

Выполнил:
Пронкин Алексей Дмитриевич
Группа: Р3208

Санкт-Петербург, 2025 г.

Цель работы

Цель лабораторной работы: решить задачу интерполяции, найти значения функции при заданных значениях аргумента, отличных от узловых точек.

Исходная таблица $y = f(x)$

i	x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$	$\Delta^5 y$	$\Delta^6 y$
0	1,05	0,1213	1,0103	0,0040	-0,0077	0,0014	0,0391	-0,1478
1	1,15	1,1316	1,0143	-0,0037	-0,0063	0,0405	-0,1087	
2	1,25	2,1459	1,0106	-0,0100	0,0342	-0,0682		
3	1,35	3,1565	1,0006	0,0242	-0,0340			
4	1,45	4,1571	1,0248	-0,0098				
5	1,55	5,1819	1,0150					
6	1,65	6,1969						

Шаг сетки: $h = 0,10$.

1. Интерполяция в точке $X_1 = 1,562$ (формула Ньютона обратнo-направленная)

Параметр $p = \frac{X_1 - x_n}{h} = \frac{1,562 - 1,65}{0,10} = -0,88$.

$$\begin{aligned} f(X_1) \approx & y_n + p \nabla y_n + \frac{p(p+1)}{2!} \nabla^2 y_n + \frac{p(p+1)(p+2)}{3!} \nabla^3 y_n \\ & + \frac{p(p+1)(p+2)(p+3)}{4!} \nabla^4 y_n + \frac{p(p+1)(p+2)(p+3)(p+4)}{5!} \nabla^5 y_n \\ & + \frac{p(p+1) \dots (p+5)}{6!} \nabla^6 y_n. \end{aligned}$$

Подстановка значений

$$\begin{aligned} f(1,562) \approx & 6,1969 + \underbrace{(-0,88) 1,0150}_{-0,8932} + \underbrace{\frac{-0,88 \cdot 0,12}{2} (-0,0098)}_{+0,00052} + \underbrace{\frac{-0,88 \cdot 0,12 \cdot 1,12}{6} (-0,0340)}_{+0,00067} \\ & + \underbrace{\frac{-0,88 \cdot 0,12 \cdot 1,12 \cdot 2,12}{24} (-0,0682)}_{+0,00071} + \underbrace{\frac{-0,88 \cdot \dots \cdot 3,12}{120} (-0,1087)}_{+0,00071} + \underbrace{\frac{-0,88 \cdot \dots \cdot 4,12}{720} (-0,1478)}_{+0,00066} \\ \approx & \boxed{5,30697}. \end{aligned}$$

Погрешность после учёта члена $\nabla^6 y$ — менее $3 \cdot 10^{-4}$.

2. Интерполяция в точке $X_2 = 1,362$ (первая формула Гаусса)

Выбираем центральный узел $x_0 = 1,35$ (строка $i = 3$),

$$p = \frac{X_2 - x_0}{h} = \frac{1,362 - 1,35}{0,10} = 0,12.$$

$$\begin{aligned}
f(X_2) \approx & y_0 + p \Delta y_0 + \frac{p(p-1)}{2!} \Delta^2 y_{-1/2} + \frac{p(p+1)(p-1)}{3!} \Delta^3 y_{-1} \\
& + \frac{p(p+1)(p-1)(p-2)}{4!} \Delta^4 y_{-3/2} + \frac{p(p+1)(p-1)(p-2)(p+2)}{5!} \Delta^5 y_{-2} \\
& + \frac{p(p+1) \dots (p-3)}{6!} \Delta^6 y_{-5/2}.
\end{aligned}$$

Подстановка значений

$$\begin{aligned}
f(1,362) \approx & 3,1565 \underbrace{+ 0,12 \, 1,0006}_{+0,12007} + \underbrace{\frac{0,12(-0,88)}{2} (-0,0100)}_{+0,00053} + \underbrace{\frac{0,12 \cdot 1,12 \cdot (-0,88)}{6} (-0,0063)}_{+0,00012} \\
& + \underbrace{\frac{0,12 \cdot 1,12 \cdot (-0,88) \cdot (-1,88)}{24} (0,0405)}_{+0,00038} + \underbrace{\frac{0,12 \dots 2,12}{120} (0,0391)}_{+0,00015} + \underbrace{\frac{0,12 \dots (-2,88)}{720} (-0,1478)}_{+0,00028} \\
\approx & \boxed{3,27803}.
\end{aligned}$$

Погрешность после члена $\Delta^6 y$ — порядка $6 \cdot 10^{-4}$.

Итоговые значения функции

$$f(1,562) = 5,307 \ (\pm 0,0003), \quad f(1,362) = 3,278 \ (\pm 0,0006).$$

Программная реализация задачи:

```

1  # -----
2  #  ФУНКЦИИ, ИНТЕГРАЛЫ КОТОРЫХ ВЫЧИСЛЯЕМ
3  # -----
4  def f(x):
5      """
6      3(x)^2
7      """
8      return 3 * x ** 2
9
10
11 def f2(x):
12     """
13     3(x)^3+4x
14     """
15     return 3 * x ** 3 + 4 * x
16
17
18 def f3(x):
19     """
20     3(x)^2+4x+5
21     """
22     return 3 * x ** 2 + 4 * x + 5
23
24
25 # -----
26 #  МЕТОДЫ ВЫЧИСЛЕНИЯ ИНТЕГРАЛОВ
27 # -----

```

```

28
29 # -----
30 # прямоугольники
31 # -----
32 def left_rectangle_method(func, a, b, n):
33     """
34     Вычисление интеграла методом левых прямоугольников.
35
36     Параметры:
37         func: интегрируемая функция,
38         a, b: пределы интегрирования,
39         n: число разбиений.
40
41     Возвращает:
42         Приближённое значение интеграла.
43     """
44     h = (b - a) / n
45     total = 0.0
46     for i in range(n):
47         x_i = a + i * h # левая точка подинтервала
48         total += func(x_i)
49     return total * h
50
51
52 def right_rectangle_method(func, a, b, n):
53     """
54     Вычисление интеграла методом правых прямоугольников.
55
56     Параметры:
57         func: интегрируемая функция,
58         a, b: пределы интегрирования,
59         n: число разбиений.
60
61     Возвращает:
62         Приближённое значение интеграла.
63     """
64     h = (b - a) / n
65     total = 0.0
66     for i in range(1, n + 1):
67         x_i = a + i * h # правая точка подинтервала
68         total += func(x_i)
69     return total * h
70
71
72 def middle_rectangle_method(func, a, b, n):
73     """
74     Вычисление интеграла методом центральных прямоугольников.
75
76     Параметры:
77         func: интегрируемая функция,
78         a, b: пределы интегрирования,
79         n: число разбиений.
80
81     Возвращает:
82         Приближённое значение интеграла.
83     """
84     h = (b - a) / n
85     total = 0.0
86     for i in range(n):

```

```

87         x_i = a + i * h + h / 2 # центральная точка подинтервала
88         total += func(x_i)
89     return total * h
90
91
92 # -----
93 # трапеция
94 # -----
95 def trapezoid_method(func, a, b, n):
96     """
97     Вычисление интеграла методом трапеций.
98
99     Параметры:
100         func: интегрируемая функция,
101         a, b: пределы интегрирования,
102         n: число разбиений.
103
104     Возвращает:
105         Приближённое значение интеграла.
106     """
107     h = (b - a) / n
108     total = (func(a) + func(b)) / 2.0
109     for i in range(1, n):
110         x_i = a + i * h # левая точка подинтервала
111         total += func(x_i)
112     return total * h
113
114
115 # -----
116 # СИМПСОН
117 # -----
118 def simpson_method(func, a, b, n):
119     """
120     Вычисление интеграла методом Симпсона.
121
122     Параметры:
123         func: интегрируемая функция,
124         a, b: пределы интегрирования,
125         n: число разбиений.
126
127     Возвращает:
128         Приближённое значение интеграла.
129     """
130     h = (b - a) / n
131     total = func(a) + func(b)
132     for i in range(1, n, 2):
133         x_i = a + i * h # левая точка подинтервала
134         total += 4 * func(x_i)
135     for i in range(2, n, 2):
136         x_i = a + i * h
137         total += 2 * func(x_i)
138     return total * h / 3
139
140
141 # -----
142 # Вычисление интеграла с правилом Рунге
143 # -----
144 def integrate_with_runge(method, func, a, b, eps, order=1, initial_n=4):
145     """

```

```

146     Вычисление интеграла с использованием заданного метода и правила Рунге для
147 оценки погрешности.
148
149     Параметры:
150         method: функция для вычисления интеграла (например, left_rectangle_method или
151             right_rectangle_method),
152         func: интегрируемая функция,
153         a, b: пределы интегрирования,
154         eps: требуемая точность,
155         order: порядок метода (для прямоугольников order = 1),
156         initial_n: начальное число разбиений (по умолчанию 4).
157
158     Возвращает:
159         Кортеж (интегральное приближение, итоговое число разбиений).
160     """
161     n = initial_n
162     I_n = method(func, a, b, n)
163     n *= 2
164     I_2n = method(func, a, b, n)
165
166     # Правило Рунге: |I(2n) - I(n)| / (2^order - 1) < eps
167     while abs(I_2n - I_n) / (2 ** order - 1) > eps:
168         I_n = I_2n
169         n *= 2
170         I_2n = method(func, a, b, n)
171
172     return I_2n, n
173
174
175 def main():
176     valid_choices = {'1', '2', '3'}
177
178     function_choice = None
179     function_to_compute = None
180     print("Вычисление интеграла функции")
181     print("Выберите функцию, которую хотите проинтегрировать:")
182     print("1 - 3x^2")
183     print("2 - 3x^3 + 4x")
184     print("3 - 3x^2 + 4x + 5")
185
186     while function_choice not in valid_choices:
187         function_choice = input("Введите номер функции: ").strip()
188         if function_choice not in valid_choices:
189             print("Неверный ввод. Пожалуйста, введите 1, 2 или 3.")
190
191     if function_choice == '1':
192         function_to_compute = f
193     elif function_choice == '2':
194         function_to_compute = f2
195     elif function_choice == '3':
196         function_to_compute = f3
197
198
199     print("Выберите метод вычисления интеграла:")
200     print("1 - Метод прямоугольников")
201     print("2 - Метод трапеций")
202     print("3 - Метод Симпсона")
203
204     choice = None

```

```

205
206 while choice not in valid_choices:
207     choice = input("Введите номер метода: ").strip()
208     if choice not in valid_choices:
209         print("Неверный ввод. Пожалуйста, введите 1, 2 или 3.")
210
211 if choice == "1":
212     print("Какой прямоугольник будем использовать для вычислений?")
213     print("1 - Левый")
214     print("2 - Средний")
215     print("3 - Правый")
216
217     choice2 = None
218
219     while choice2 not in valid_choices:
220         choice2 = input("Введите номер метода: ").strip()
221         if choice2 not in valid_choices:
222             print("Неверный ввод. Пожалуйста, введите 1, 2 или 3.")
223
224         if choice2 == "1":
225             integration_method = left_rectangle_method
226             order = 1 # порядок метода для прямоугольников (левый)
227         elif choice2 == "2":
228             integration_method = middle_rectangle_method
229             order = 2 # порядок метода для прямоугольников (средний)
230         elif choice2 == "3":
231             integration_method = right_rectangle_method
232             order = 1 # порядок метода для прямоугольников (правый)
233     elif choice == "2":
234         integration_method = trapezoid_method
235         order = 2
236     elif choice == "3":
237         integration_method = simpson_method
238         order = 4
239     else:
240         print("Некорректный выбор метода.")
241         return
242
243 try:
244     a = float(input("Введите нижний предел интегрирования (a): "))
245     b = float(input("Введите верхний предел интегрирования (b): "))
246     eps = float(input("Введите требуемую точность (eps): "))
247 except ValueError:
248     print("Ошибка: необходимо вводить числовые значения.")
249     return
250
251 result, subdivisions = integrate_with_runge(
252     integration_method, function_to_compute, a, b, eps, initial_n=4, order=order
253 )
254
255 print("\nРезультаты вычисления:")
256 print(f"Приближённое значение интеграла: {result}")
257 print(f"Число разбиений для достижения требуемой точности: {subdivisions}")
258
259
260 if __name__ == "__main__":
261     main()

```