



Министерство науки и высшего образования Российской Федерации  
федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования  
«Московский государственный технический университет имени  
Н.Э. Баумана (национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Робототехники и комплексной автоматизации»  
КАФЕДРА «Системы автоматизированного проектирования (РК-6)»

**ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ №2**  
**по дисциплине «Вычислительная математика»**

Студент:	Гавриш Александр Александрович
Группа:	РК6-51Б
Тип задания:	лабораторная работа
Тема:	Использование аппроксимаций для численной оптимизации

Студент

\_\_\_\_\_

подпись, дата

Гавриш А.А.

Фамилия, И.О.

Преподаватель

\_\_\_\_\_

подпись, дата

Фамилия, И.О.

Москва, 2023

# Содержание

<b>Использование аппроксимаций для численной оптимизации</b>	<b>3</b>
1 Задание . . . . .	3
2 Цель выполнения лабораторной работы . . . . .	4
3 Разработка функций численного интегрирования на основе составных формул Симпсона и трапеций . . . . .	4
4 Численное интегрирование кривой наискорейшего спуска с помощью составных формул Симпсона и трапеций . . . . .	6
5 Построение log-log графиков зависимостей абсолютной погрешности численного интегрирования от шага интегрирования для составных формул Симпсона и трапеций . . . . .	8
6 Определение порядка точности формулы по графику . . . . .	12
7 Сравнение аналитических порядков точности с порядками точности, получаемыми с помощью соответствующих графиков . . . . .	13
8 Определение оптимального шага интегрирования для заданной формулы . . . . .	15
9 Преобразование задачи о минимизации функционала к полудискретной форме . . . . .	16
10 Преобразование задачи к полностью дискретной форме с помощью составной формулы Симпсона . . . . .	18
11 Минимизация функционала . . . . .	18
12 Оценка погрешности решения для различных конфигураций дискретизации . . . . .	21
13 Заключение . . . . .	25

# Использование аппроксимаций для численной оптимизации

## 1 Задание

Методы аппроксимации, такие как интерполяция и численное интегрирование, часто используются как составные блоки других, более сложных численных методов. В данной лабораторной работе мы рассмотрим одну из старейших задач вариационного исчисления: задачу о брахистохроне, т.е. задачу о кривой наискорейшего спуска. Она состоит в нахождении такой кривой, по которой материальная точка из точки  $(x, y) = (0, 0)$  достигнет точки  $(x, y) = (a, y_a)$  под действием силы тяжести за наименьшее время (здесь и далее ось  $y$  направлена вниз). Решением этой задачи является такая кривая  $y(x)$ , которая минимизирует следующий функционал, являющийся полным временем движения материальной точки:

$$\mathcal{F}[y] = \int_0^a \sqrt{\frac{1 + (y'(x))^2}{2gy(x)}} dx, \quad (1)$$

где  $g$  обозначает ускорение свободного падения, и  $y'(x) = dy/dx$ .

Представленная задача имеет аналитическое решение, которым является параметрически заданная циклоида:

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = C \begin{bmatrix} t - \frac{1}{2} \sin(2t) \\ \frac{1}{2} - \frac{1}{2} \cos(2t) \end{bmatrix}, \quad (2)$$

где  $t \in [0; T]$  и  $C, T$  являются константами, значения которых находятся из граничного условия.

В базовой части задания требуется воспользоваться численным интегрированием для нахождения полного времени движения материальной точки по кривой наискорейшего спуска. В продвинутой части требуется разработать метод для нахождения аппроксимации этой кривой. Здесь и далее принимается  $a = 2$  и  $y_a = 1$ . Константы циклоиды для этого граничного условия равны  $C = 1.03439984$ ,  $T = 1.75418438$ .

Требуется (базовая часть):

1. Разработать функции `composite_simpson(a, b, n, f)` и `composite_trapezoid(a, b, n, f)` численного интегрирования некоторой функции  $f$  на интервале  $[a; b]$  по  $n$  узлам с помощью составной формулы Симпсона и составной формулы трапеций соответственно.
2. Рассчитать интеграл (1) для функции  $y(x)$ , соответствующей кривой наискорейшего спуска, с помощью составной формулы Симпсона и составной формулы трапеций для множества  $n \in [3; 9999]$ .
3. На одной координатной плоскости постройте log-log график зависимости абсолютной погрешности численного интегрирования от шага интегрирования для обеих формул.

4. Объясните, каким образом по полученному графику можно определить порядок точности формулы.

5. Для обеих формул сравните известные аналитические порядки точности с порядками точности, получаемыми с помощью соответствующих графиков. Обоснуйте алгоритм сравнения.

6. Существует ли оптимальный шаг интегрирования для первой или второй формулы, минимизирующий достижимую погрешность? Обоснуйте свой ответ.

Требуется (продвинутая часть):

1. Используя кусочно-линейную интерполяцию с равноудаленными узлами, преобразовать задачу о минимизации функционала (1) к полудискретной форме, где аргументами минимизации будут параметры кусочно-линейной интерполяции.

2. Далее, используя составную формулу Симпсона, преобразовать задачу к полностью дискретной форме.

3. Решить полученную задачу минимизации, используя различные конфигурации дискретизации: с шагом интерполяции и шагом интегрирования от  $10^{-3}$  до 1.

4. Используя log-log графики и линии уровня, оценить зависимость погрешности решения от шага интерполяции и шага интегрирования.

## 2 Цель выполнения лабораторной работы

Цель выполнения лабораторной работы: с помощью численного интегрирования найти полное время движения материальной точки по кривой наискорейшего спуска (базовая часть), разработать метод для нахождения аппроксимации этой кривой (продвинутая часть).

## 3 Разработка функций численного интегрирования на основе составных формул Симпсона и трапеций

Составная формула Симпсона имеет следующий вид [1]:

$$\int_a^b f(x) dx = \frac{h}{3} \left[ f(x_1) + 2 \sum_{i=1}^{n/2-1} f(x_{2i+1}) + 4 \sum_{i=1}^{n/2} f(x_{2i}) + f(x_{n+1}) \right] - \frac{(b-a)h^4}{180} f^{(4)}(\xi), \quad (3)$$

где  $a$  и  $b$  - левая и правая границы интегрирования соответственно,  $x_i = a + (i-1)h$ ,  $i = 1, \dots, n+1$ ,  $n$  - четное число,  $h = \frac{b-a}{n}$  - шаг интегрирования.

Составная формула Симпсона относится к составным квадратурным формулам четвертого порядка точности. Это означает, что ее остаточный член имеет порядок  $O(h^4)$ , что нетрудно заметить из самой формулы (3). На основе этого будем считать, что при малых шагах интегрирования  $h$  величиной остаточного члена можно пренебречь без существенной потери точности. Исходя из этого, в нашей программной реализации остаточный член учитываться не будет.

Тогда итоговый вид составной формулы Симпсона будет выглядеть следующим образом:

$$\int_a^b f(x) dx \approx \frac{h}{3} \left[ f(x_1) + 2 \sum_{i=1}^{n/2-1} f(x_{2i+1}) + 4 \sum_{i=1}^{n/2} f(x_{2i}) + f(x_{n+1}) \right] \quad (4)$$

Составная формула Симпсона предполагает использование равномерной сетки узлов интегрирования. Для того, чтобы обеспечить правильное вычисление интеграла по этой формуле, необходимо, чтобы число отрезков интегрирования было четным (достигается при нечетном числе узлов). Поэтому в случае, если на вход программной реализации данной формулы поступает нечетное число узлов  $n$ , оно автоматически уменьшается на 1. Это позволяет гарантировать вычисление интеграла строго в соответствии с математическими свойствами формулы Симпсона.

Программная реализация составной формулы Симпсона (4) для численного интегрирования функции  $f$  на отрезке  $[a; b]$  по  $n$  узлам приведена на [листе 1](#).

#### Листинг 1. Программная реализация составной формулы Симпсона

---

```
def composite_simpson(a: float, b: float, n: int, f: Callable) -> float:
    if a > b:
        raise ValueError('a > b')
    if n < 3:
        raise ValueError('n < 3')
    if not isinstance(f, Callable):
        raise TypeError('f is not callable')

    if n % 2 == 0:
        n -= 1

    h = (b - a) / (n - 1)

    x_values = np.linspace(a, b, n)
    f_values = f(x_values)

    odd_sum = np.sum(f_values[2:-1:2])
    even_sum = np.sum(f_values[1:-1:2])

    return (h / 3) * (f_values[0] + 2 * odd_sum +
                      4 * even_sum + f_values[-1])
```

---

**Функция** `composite_simpson(a, b, n, f)` принимает на вход четыре аргумента: левую и правую границу интегрирования, число узлов, функцию для интегрирования. Первым делом необходимо проверить корректность входных аргументов, а именно: левая граница не должна быть больше правой, число узлов должно быть не меньше трех и аргумент `f` должен быть `Callable`, то есть должен поддерживать возможность его вызова в качестве функции.

Составная формула трапеций имеет следующий вид [1]:

$$\int_a^b f(x) dx = \frac{h}{2} \left[ f(x_1) + 2 \sum_{i=2}^n f(x_i) + f(x_{n+1}) \right] - \frac{(b-a)h^2}{12} f''(\xi), \quad (5)$$

где  $a$  и  $b$  - левая и правая границы интегрирования соответственно,  $x_i = a + (i-1)h$ ,  $i = 1, \dots, n+1$ ,  $n \in N$ ,  $h = \frac{b-a}{n}$  - шаг интегрирования.

Подобно составной формуле Симпсона, в нашей программной реализации мы опускаем остаточный член составной формулы трапеций, поскольку его величина пренебрежимо мала при малом шаге интегрирования. Несмотря на то, что составная формула трапеций, как и  $x$ -формула Симпсона, применима только для равномерной сетки узлов, в отличие от последней она не накладывает ограничений на чётность числа отрезков разбиения.

Программная реализация составной формулы трапеций (5) без остаточного члена для численного интегрирования функции  $f$  на отрезке  $[a; b]$  по  $n$  узлам приведена на [листеинге 2](#).

#### Листинг 2. Программная реализация составной формулы трапеций

---

```
def composite_trapezoid(a: float, b: float, n: int, f: Callable) -> float:
    if a > b:
        raise ValueError('a > b')
    if n < 2:
        raise ValueError('n < 2')
    if not isinstance(f, Callable):
        raise TypeError('f is not callable')

    h = (b - a) / (n - 1)
    x_values = np.linspace(a, b, n)
    f_values = f(x_values)

    sum = np.sum(f_values[1:-1])

    return (h / 2) * (f_values[0] + 2 * sum + f_values[-1])
```

---

**Функция** `composite_trapezoid(a, b, n, f)` принимает на вход четыре аргумента: левую и правую границу интегрирования, число узлов, функцию для интегрирования. Первым делом необходимо проверить корректность входных аргументов, а именно: левая граница не должна быть больше правой, число узлов должно быть не меньше двух и функция для интегрирования должна быть `Callable`, то есть должна поддерживать возможность ее вызова в качестве функции.

## 4 Численное интегрирование кривой наискорейшего спуска с помощью составных формул Симпсона и трапеций

Для дальнейших вычислений интеграла (1) выполним переход от интегрирования по  $x$  к интегрированию по параметру  $t$ , используя соотношения между  $x$  и  $t$ , определяемые параметрическими уравнениями (2) для циклоиды.

$$\int_0^2 \sqrt{\frac{1 + (y'(x))^2}{2gy(x)}} dx = \int_0^T \sqrt{\frac{1 + \left(\frac{y'(t)}{x'(t)}\right)^2}{2gy(t)}} x'(t) dt \quad (6)$$

где  $y'(t) = dy/dt$  и  $x'(t) = dx/dt$ .

Окончательно имеем следующую функцию для интегрирования:

$$f(t) = \sqrt{\frac{1 + \left(\frac{y'(t)}{x'(t)}\right)^2}{2gy(t)}} x'(t) \quad (7)$$

Найдем необходимые производные из аналитического решения (2):

$$\begin{aligned} \frac{dy}{dt} &= C \left[ \frac{1}{2} - \frac{1}{2} \cos(2t) \right] \frac{1}{dt} = C \sin(2t), \\ \frac{dx}{dt} &= C \left[ t - \frac{1}{2} \sin(2t) \right] \frac{1}{dt} = C [1 - \cos(2t)] \end{aligned} \quad (8)$$

Заметим, что при непосредственной подстановке производных, в точке  $t = 0$  возникает ситуация деления на ноль. Поэтому примем нижнюю границу интегрирования равной  $1e-7$  (в следующих пунктах рассмотрим и другие варианты решения данной проблемы).

Результаты интегрирования функции (7) составными формулами Симпсона и трапеций в зависимости от числа узлов интегрирования представлены на рисунке 1.

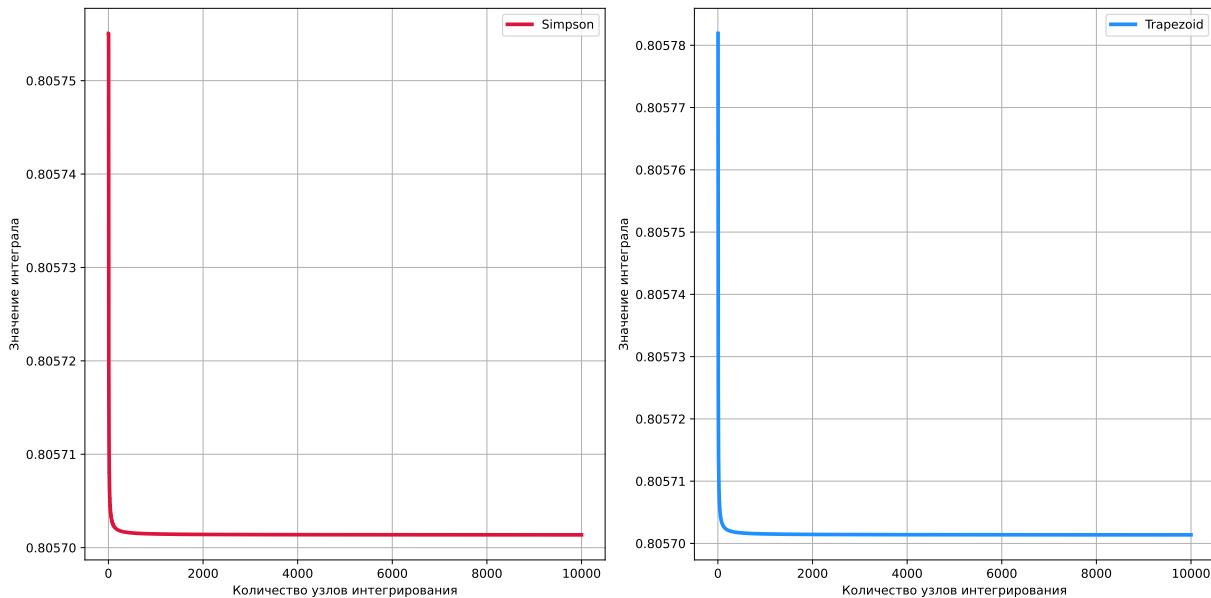


Рис. 1. Зависимость вычисленного значения интеграла от количества узлов интегрирования при использовании составной формулы Симпсона (слева) и составной формулы трапеций (справа)

Программная реализация интегрирования произвольной функции `integrant` на отрезке  $[T_0, T]$  с использованием составных формул Симпсона и трапеций для числа узлов  $N \in [3; 9999]$  приведена на [листинге 3](#).

Листинг 3. Функция численного интегрирования составными формулами Симпсона и трапеций

---

```
def integrate(integrant: Callable, T0: float) -> Tuple[np.ndarray, np.ndarray, np.ndarray]:
    N = np.arange(N_RANGE['min'], N_RANGE['max'], N_RANGE['step'])

    h = (T - T0) / (N - 1)
    simpson = [composite_simpson(T0, T, n, integrant) for n in N]
    trapezoid = [composite_trapezoid(T0, T, n, integrant) for n in N]

    return h, simpson, trapezoid
```

---

**Функция** `integrate` принимает аргумент `integrant` (должен поддерживать возможность его вызова в качестве функции) для интегрирования и левую границу интегрирования  $T_0$ . Согласно заданному диапазону  $N$  генерируется вектор узлов интегрирования. Далее с помощью спискового включения вычисляются значения интегралов по составным формулам Симпсона и трапеций для каждого числа узлов. Функция возвращает шаг интегрирования, результаты по Симпсону и трапециям.

## 5 Построение log-log графиков зависимостей абсолютной погрешности численного интегрирования от шага интегрирования для составных формул Симпсона и трапеций

Введем определение абсолютной погрешности. Абсолютной погрешностью приближенного значения  $a^*$  называют величину  $\Delta(a^*)$ , которая определена как:

$$\Delta(a^*) = |a - a^*|,$$

где  $a$  - точное значение.

Для нахождения абсолютной погрешности посчитаем истинное значение интеграла (1).

Подставим полученные ранее производные (8) в формулу (6):

$$\begin{aligned}
\int_0^T \sqrt{\frac{1 + \left(\frac{y'(t)}{x'(t)}\right)^2}{2gy(t)}} x'(t) dt &= \int_0^T \sqrt{\frac{1 + \left(\frac{C \sin(2t)}{C[1 - \cos(2t)]}\right)^2}{gC[1 - \cos(2t)]}} C[1 - \cos(2t)] dt = \\
&= \int_0^T \sqrt{\frac{1 - 2\cos(2t) + \cos^2(2t) + \sin^2(2t)}{(1 - \cos(2t))^2}} C[1 - \cos(2t)] dt = \\
&= \int_0^T \sqrt{\frac{2}{gC(1 - \cos(2t))}} C[1 - \cos(2t)] dt = \\
&= \int_0^T \sqrt{\frac{2C}{g} \frac{1 - \cos(2t)}{1 - \cos(2t)}} dt = \sqrt{\frac{2C}{g}} \int_0^T dt
\end{aligned}$$

Тогда точное значение интеграла (1):

$$T \sqrt{\frac{2C}{g}} \quad (9)$$

Вернемся к проблеме, описанной пунктом выше: при непосредственной подстановке производных, в точке  $t = 0$  возникает ситуация деления на ноль. Предложены следующие варианты решения:

1. Ввиду физического смысла задачи принять допущение, что в точке с координатами  $(x, y) = (0, 0)$  элементарное перемещение точки и ее скорость также равны 0  $\Rightarrow f(0) = 0$  ( $T_{full} = \int_a^b dS/v$ , где  $dS$  - элементарное перемещение точки по кривой).
2. Возвращать аналитически вычисленное значение  $\sqrt{\frac{2C}{g}}$  (9).
3. В качестве нижней границы интегрирования взять число в окрестности нуля ( $a = 0.0000001$ ) и, учитывая данное допущение, в качестве точного значения интеграла (1) использовать  $\sqrt{\frac{2C}{g}}(T - 10^{-7})$ .

Для каждого варианта решения построены log-log графики зависимости абсолютной погрешности Е численного интегрирования от шага интегрирования  $h$  для составных формул Симпсона и трапеций (рис. 2), (рис. 3), (рис. 4).

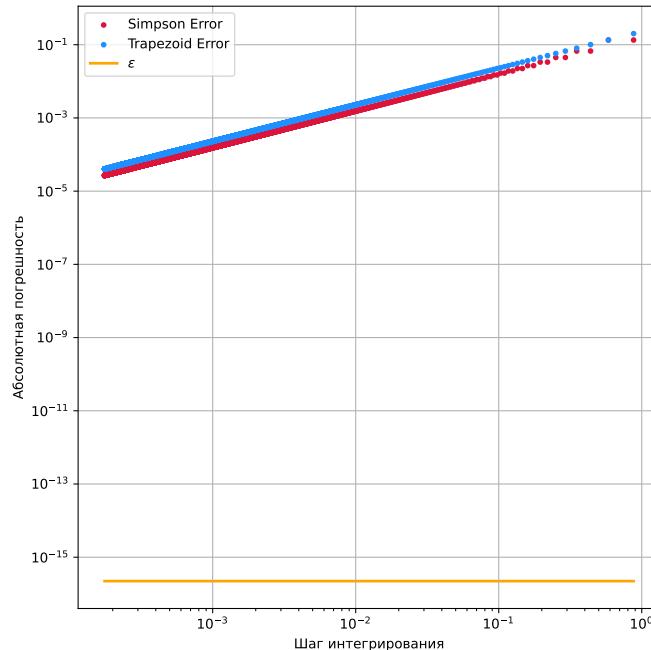


Рис. 2. Зависимость абсолютной погрешности численного интегрирования от шага интегрирования ( $func(0)=0$ )

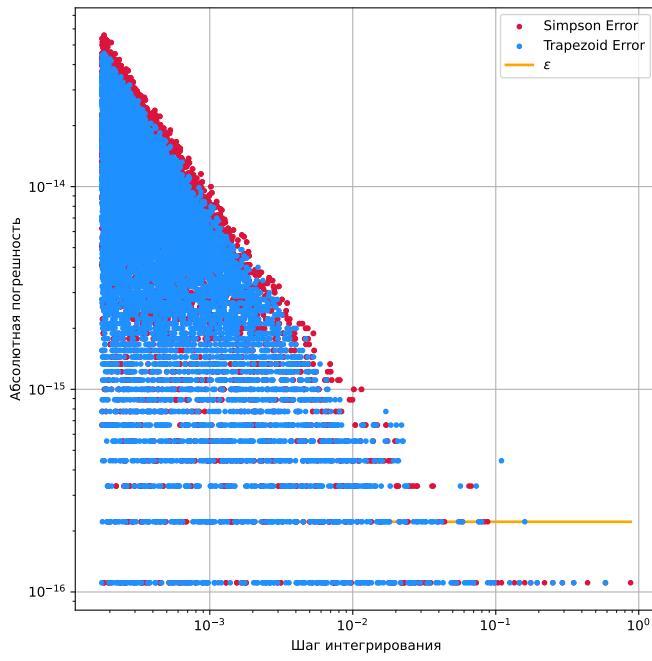


Рис. 3. Зависимость абсолютной погрешности численного интегрирования от шага интегрирования ( $\text{func}(0)=\sqrt{\frac{2C}{g}}$ )

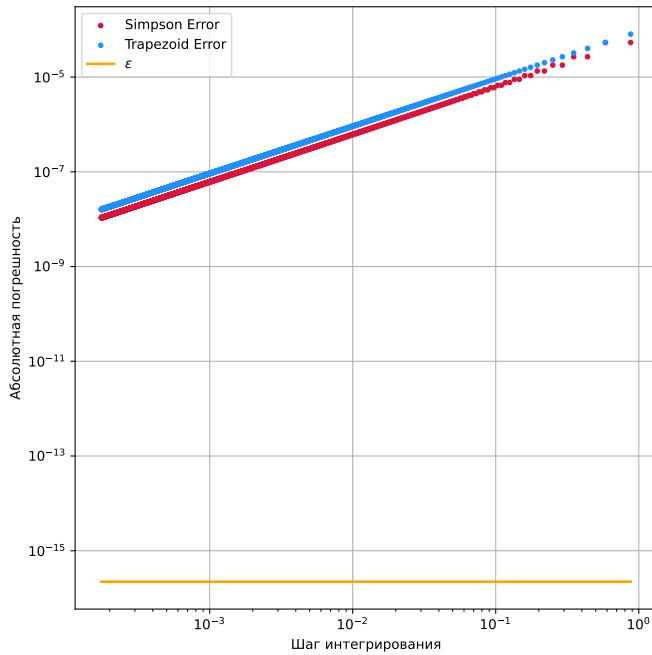


Рис. 4. Зависимость абсолютной погрешности численного интегрирования от шага интегрирования ( $a = 0.0000001$ )

## 6 Определение порядка точности формулы по графику

Порядок точности [1] - наибольшая степень полинома, для которой численный метод дает точное решение задачи. Иными словами, говорят, что численный метод имеет порядок точности  $d$ , если его отстаток  $R_n$  равен нулю для любого полинома степени  $d$ , но не равен нулю для полинома степени  $d + 1$ .

Обозначим порядок точности как  $O(h)$ . В рамках лабораторной работы порядок точности представляет зависимость точности от величины шага. На рисунке 5 представлен log-log график, поэтому порядок точности определяется как тангенс угла наклона графика, так как при логарифмировании степень точности численного метода становится тангенсом угла наклона.

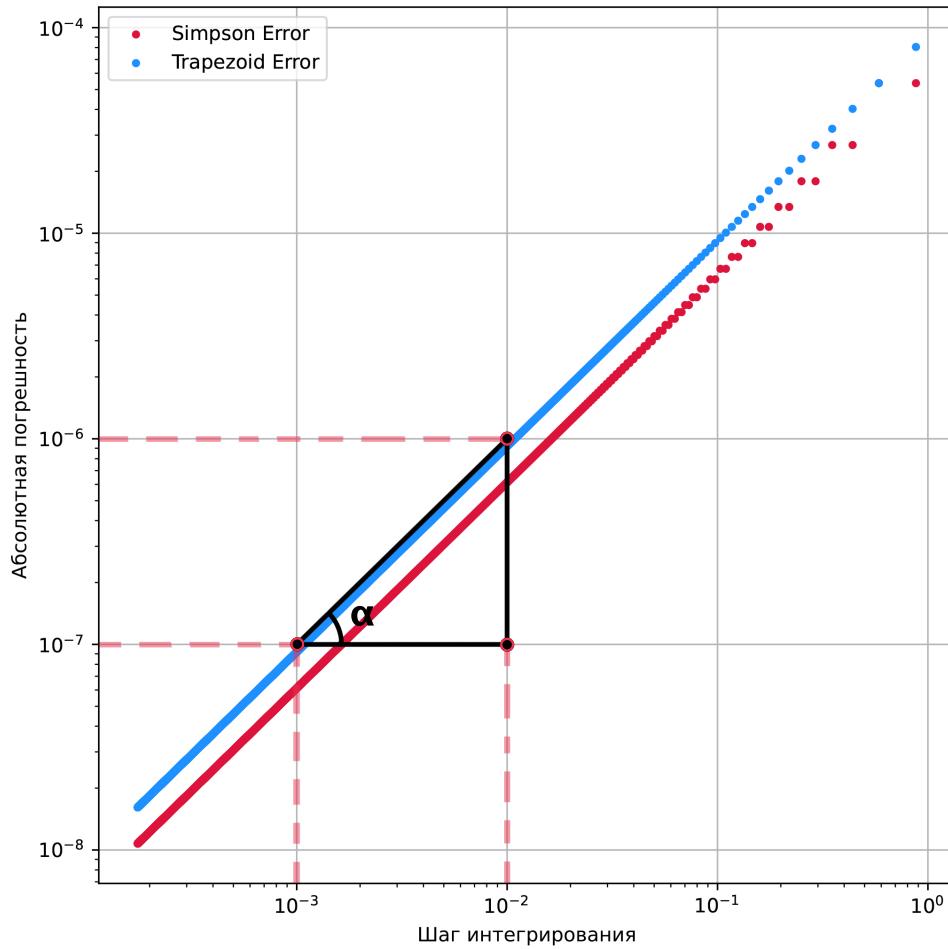


Рис. 5. Определение порядка точности как тангенс угла наклона

Заметим, что график для формулы Симпсона параллелен графику для формулы трапеций, соответственно углы равны.

$$\tan(\alpha) = \frac{\log_{10} 10^{-6} - \log_{10} 10^{-7}}{\log_{10} 10^{-2} - \log_{10} 10^{-3}} = 1 \quad (10)$$

Получаем  $O(h) = 1$  для формулы Симпсона и формулы трапеций.

## 7 Сравнение аналитических порядков точности с порядками точности, получаемыми с помощью соответствующих графиков

На рисунке 6 и рисунке 7 отображены реальный порядок точности интегрирования функции  $f(x)$  и аналитический порядок точности интегрирования функции  $f(x)$  для обеих формул (варианты решения 1, 3).

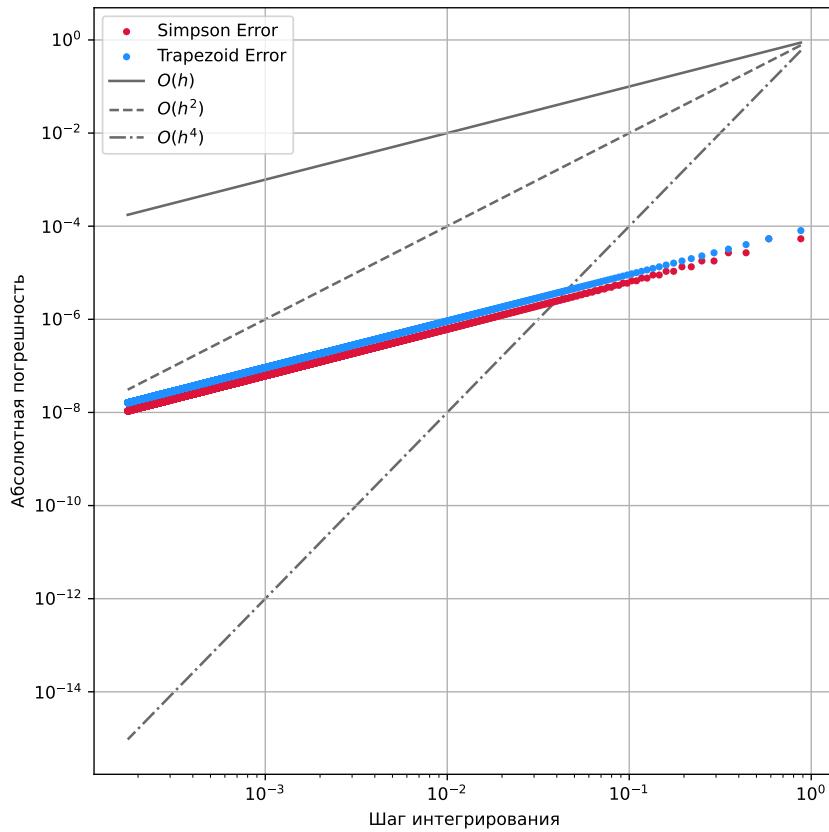


Рис. 6. Реальный порядок точности интегрирования функции  $f(x)$  и аналитический порядок точности интегрирования функции ( $a = 0.0000001$ )

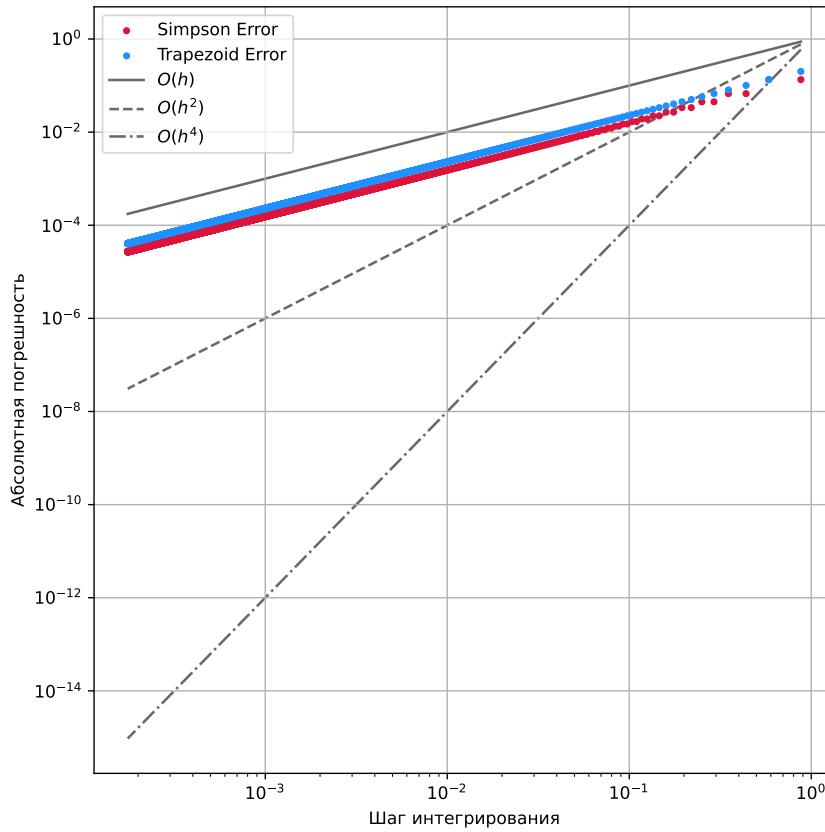


Рис. 7. Реальный порядок точности интегрирования функции  $f(x)$  и аналитический порядок точности интегрирования функции ( $\text{func}(0) = 0$ )

Остаточный член составной формулы Симпсона для функции  $f(x)$  имеет следующий вид [1]:

$$-\frac{(b-a)h^4}{180}f^{(4)}(\xi),$$

где  $\xi \in (a; b)$ ,  $f(x) \in C^4[a; b]$ .

Остаточный член составной формулы трапеций для функции  $f(x)$  имеет следующий вид [1]:

$$-\frac{(b-a)h^2}{12}f''(\xi),$$

где  $\xi \in (a; b)$ ,  $f(x) \in C^2[a; b]$ .

Тогда аналитический порядок точности для составной формулы Симпсона пропорционален  $O(h^4)$ , а аналитический порядок точности для составной формулы трапеций пропорционален  $O(h^2)$ .

В обоих случаях реальная точность пропорциональна  $O(h)$ , такое же значение было получено в (10), что в свою очередь связано с нарушением условия необходимой гладкости функции  $f(x)$ . Для случая, представленного на рисунке 3, значение абсолютной

погрешности находится около машинного  $\epsilon$  при любом шаге, поэтому предложенный анализ (сравнение точностей) для данного случая не проводился. Соответственно, любой шаг обеспечит хорошую точность.

Гладкая функция [1] - это функция, имеющая непрерывную производную на всем множестве определения. Рассматривают также гладкие функции высших порядков, а именно, функция с порядком гладкости  $r$  имеет непрерывную производную порядка  $r$ .

Найдем первую производную функции 7, используя математический калькулятор WolframAlpha:

$$f'(t) = \frac{2 \cos(2t)(1 - \cos^2(2t)) - 4 \cos(2t) \sin^2(2t)}{\sqrt{\frac{2}{g}(1 - \cos^2(2t))^2} \sqrt{\frac{\sin(2t)}{1 - \cos^2(2t)} + 1}} \quad (11)$$

Построим график производной 11 в графическом калькуляторе Desmos (рис. 8):

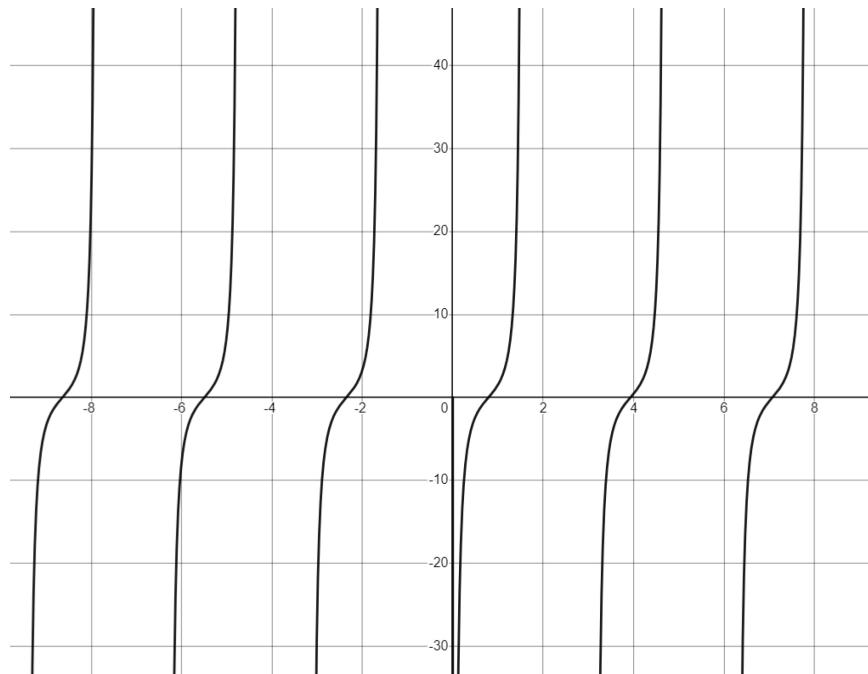


Рис. 8. График производной функции 7, построенный в Desmos

Заметим, что функция терпит разрыв в точках  $x = 0 \pm i\pi$ , где  $i \in Z$ , соответственно функция 11 не является непрерывной на участке  $[0; 2]$ , поэтому можно утверждать, что функция 7 недостаточно гладкая, отсюда такая разница в порядках точности.

## 8 Определение оптимального шага интегрирования для заданной формулы

По сравнению с дифференцированием, операция интегрирования способна к стабилизации вычислительной погрешности. Интуитивное объяснение этого эффекта за-

лючается в том, что интегрирование предполагает суммирование близких значений, в то время как дифференцирование вычисляет их разность.

Рассмотрим составную формулу Симпсона и предположим, что значение  $f(x)$  в точке  $x_i$  вычисляется с погрешностью округления  $e_i$ :

$$f(x_i) = \hat{f}(x_i) + e_i, \quad i = 1, \dots, n+1$$

Тогда полная погрешность округления, накапливаемая составной формулой Симпсона, может быть оценена следующим образом:

$$e(h) = \frac{h}{3} \left[ e_1 + 2 \sum_{i=1}^{n/2-1} e_{2i+1} + 4 \sum_{i=1}^{n/2} e_{2i} + e_{n+1} \right] \leq \frac{h}{e} \left[ |e_1| + 2 \sum_{i=1}^{n/2-1} |e_{2i+1}| + 4 \sum_{i=1}^{n/2} |e_{2i}| + |e_{n+1}| \right]$$

Предположим, что погрешность округления ограничена, например, минимальным эпсилоном:  $|e_i| \leq \epsilon, i = 1, \dots, n+1$ . Тогда полная погрешность погрешность оценивается как:

$$e(h) \leq \frac{h\epsilon}{3} \left[ 1 + 2 \left( \frac{n}{2} - 1 \right) + 4 \frac{n}{2} + 1 \right] = nh\epsilon = (b-a)\epsilon$$

Этот результат показывает, что верхняя грань для накопленной погрешности округления не зависит от  $h$  и  $n$ , что означает, что увеличение числа подотрезков не приводит к дестабилизации полной погрешности. Действительно, можно заметить, что полная погрешность интегрирования падает до тех пор, пока не достигнет значения, сравнимого с машинным эпсилоном, после чего уменьшение погрешности становится невозможным, и она стабилизируется на уровне машинного эпсилон.

## 9 Преобразование задачи о минимизации функционала к полудискретной форме

Задача минимизации состоит в нахождении такой функции  $y^*(x)$ , которая минимизирует функционал (1):

$$y^*(x) = \operatorname{argmin} F[y]$$

Для преобразования исходной задачи минимизации функционала (1) к полудискретной форме рассмотрим два подхода:

**Метод нахождения оптимальных параметров интерполяции из решения системы уравнений**

Пусть функция  $y(x)$  аппроксимируется кусочно-линейной функцией  $Y(X)$  на отрезке  $[0; a]$ . Тогда, учитывая, что  $y(x) = \sum_{i=1}^{n-1} \alpha_i + \beta_i x$  и  $y'(x) = \beta_i$ , функционал будет представлен в виде суммы интегралов:

$$F[y] = \sum_{i=1}^{n-1} \sqrt{\frac{1+\beta_i^2}{2g}} \int_{x_i}^{x_{i+1}} \frac{dx}{\sqrt{\alpha_i + \beta_i x}} = H(\alpha, \beta)$$

Для решения данной задачи необходимо найти  $\alpha_i$  и  $\beta_i$ , рассмотрим задачу минимизации при нахождении коэффициентов прямой, тогда задача интерполяции будет "вложена" в задачу оптимизации. Найдем производные  $H(\alpha, \beta)$  по  $\alpha_i$  и  $\beta_i$ , приравняем их к нулю, после чего выражаем соответствующие коэффициенты. Следует учесть граничные условия, т.к. именно на граничных узлах заданы значения соответствующих ординат. Также запишем уравнение для сопряжения соседних узлов и итоговую СЛАУ.

$$\begin{aligned} \frac{\partial H}{\partial \alpha_i} &= 0, \quad \frac{\partial H}{\partial \beta_i} = 0 \\ \frac{\delta H}{\delta \alpha_j} &= \sum_{i=1}^{n-1} \sqrt{\frac{1+\beta_i^2}{2g}} \int_{x_i}^{x_{i+1}} \frac{\delta}{\delta \alpha_j} \frac{dx}{\sqrt{\alpha_i + \beta_i x}} = \sqrt{\frac{1+\beta_j^2}{2g}} \int_{x_j}^{x_{j+1}} (\alpha_j + \beta_j x)^{-\frac{3}{2}} dx \\ \frac{\delta H}{\delta \beta_j} &= \sum_{i=1}^{n-1} \int_{x_i}^{x_{i+1}} \frac{\delta}{\delta \beta_j} \left[ \left( \frac{1+\beta_i^2}{2g(\alpha_i + \beta_i x)} \right)^2 \right] dx = \int_{x_j}^{x_{j+1}} \left( \frac{\alpha_j + \beta_j x}{2g(1+\beta_j^2)} \right)^2 \frac{\beta + j\alpha_j - x}{2(\alpha_j + \beta_j x)^2} dx = \\ &= \int_{x_j}^{x_{j+1}} \left( \frac{\alpha_j + \beta_j x}{2g(1+\beta_j^2)} \right)^{\frac{1}{2}} \frac{\beta_j \alpha_j - x}{2(\alpha_j + \beta_j x)^2} dx \end{aligned}$$

Тогда итоговая система, содержащая граничные условия и условия сопряжения соседних узлов будет выглядеть следующим образом:

$$\begin{cases} \sqrt{\frac{1+\beta_j^2}{2g}} \int_{x_j}^{x_{j+1}} (\alpha_j + \beta_j x)^{-\frac{3}{2}} dx = 0, & j = 1, 2, \dots, n-1 \\ \int_{x_j}^{x_{j+1}} \left( \frac{\alpha_j + \beta_j x}{2g(1+\beta_j^2)} \right)^{\frac{1}{2}} \frac{\beta_j \alpha_j - x}{2(\alpha_j + \beta_j x)^2} dx = 0, \\ Y_j(x_{j+1}) = Y_{j+1}(x_{j+1}), & j = 1, 2, \dots, n-2 \\ Y_1(0) = 0, \\ Y_n(a) = y_a \end{cases} \quad (12)$$

Решение полученной системы уравнений (12) позволит найти оптимальные значения параметров  $\alpha_j, \beta_j$ . Однако аналитическое решение затруднительно и нахождение коэффициентов прямых довольно нетривиальная задача.

### Прямое преобразование функционала

Пусть  $Y(x)$  - кусочно-линейная аппроксимация функции  $y(x)$ , где  $x \in [a; b]$ . Тогда справедливо следующее преобразование интеграла (1):

$$\int_a^b \sqrt{\frac{1+(y'(x))^2}{2g(y(x))}} dx \approx \sum_{j=1}^{N-1} \int_{x_j}^{x_{j+1}} \sqrt{\frac{1+(Y'_j(x))^2}{2g(Y_j(x))}} dx, \quad (13)$$

где  $N$  - количество интерполяционных узлов,  $Y_j(x) = y(x_j) + (y(x_{j+1}) - y(x_j)) \frac{x-x_j}{x_{j+1}-x_j}$ , для  $x \in [x_j; x_{j+1}]$ ,  $j = 1 \dots N - 1$ .

Коэффициенты  $\alpha$  и  $\beta$  найдем следующим образом:

$$\alpha = y(x_j), \quad \beta = \frac{x - x_j}{x_{j+1} - x_j},$$

где  $x \in [x_j; x_{j+1}]$ ,  $j = 1 \dots N - 1$ .

Формула (13) представляет функционал в полудискретном виде.

## 10 Преобразование задачи к полностью дискретной форме с помощью составной формулы Симпсона

Используя составную формулу Симпсона (4), было выполнено преобразование исходного интеграла (1) к полностью дискретной форме, используя формулу (13):

$$\sum_{j=1}^{N-1} \int_{x_j}^{x_{j+1}} \sqrt{\frac{1 + (Y'_j(x))^2}{2g(Y_j(x))}} dx \approx \frac{h}{3} \sum_{j=1}^{N-1} \left[ f_j(x_j) + 2 \sum_{i=1}^{n/2-1} f_j(x_{2i+1}) + 4 \sum_{i=1}^{n/2} f_j(x_{2i}) + f_j(x_{n+1}) \right], \quad (14)$$

$$\text{где } f_j(x) = \sqrt{\frac{1 + (Y'_j(x))^2}{2g(Y_j(x))}}.$$

Таким образом, функционал (1) был приведен к дискретной форме с параметрами дискретизации  $N$ ,  $n$ .

Разработана функция `functional_value(y_nodes, x_nodes, A, B, n_integr)`, которая вычисляет значение функционала (1) для аппроксимации неизвестной кривой  $y(x)$  (см. [листинг 4](#)).

Листинг 4. Функция расчета значения функционала

---

```
def functional_value(y_nodes: np.ndarray, x_nodes: np.ndarray,
                     A: float, B: float, n_integr: int) -> float:
    s = 0
    y = np.concatenate(([A], y_nodes, [B]))
    coeffs = coeff_lin(y, x_nodes)
    for i in range(len(x_nodes) - 1):
        s += composite_simpson(coeffs[i], func, x_nodes[i],
                               x_nodes[i + 1], n_integr, x_nodes, y)
    return s
```

---

## 11 Минимизация функционала

Требуется найти такой вектор значений функции  $y(x)$ , при котором значение выражения (14) будет минимальным. При этом должны быть выполнены некоторые условия, вытекающие из физического смысла задачи:

$$1. \quad y(x) > 0: \forall x \in [a; b]$$

$$2. \quad y(a) = 0, y(b) = 1$$

Таким образом, необходимо решить задачу условной минимизации.

Для ее решения была разработана функция `functional_value(y_nodes, x_nodes, A, B, n_integr)`, которая принимает на вход целевую функцию (см. [листинг 4](#)), вектор параметров минимизации `y_nodes` (вектор значений функции  $y(x)$  для  $x \in [a; b]$ ), граничные условия А ( $y(a) = 0$ ), В ( $y(b) = 1$ ), интерполяционные узлы по  $x$  и параметр дискретизации интегрирования соответственно (см. [листинг 5](#)). Непосредственно сама минимизация осуществляется с помощью библиотечной функции `scipy.optimize.minimize`, которая в качестве второго аргумента принимает первоначальное предположение.

Листинг 5. Функция решения задачи условной минимизации

---

```
def get_optim_nodes_efficient(f: callable, y_nodes: np.ndarray,
                               A: float, B: float, x_nodes: np.ndarray, n_integr: int):
    bounds = Bounds(0.0001, np.inf)
    result = minimize(f, y_nodes, args=(x_nodes, A, B, n_integr),
                      method='L-BFGS-B', bounds=bounds, options={'maxiter': 15000})
    return result.x, result.fun
```

---

На рисунках [9](#), [10](#), [11](#) представлены результаты оптимизации для различных параметров дискретизации, где в качестве первоначального предположения кривая  $y(x) = \frac{x}{2}$  (ось  $y$  направлена вниз).

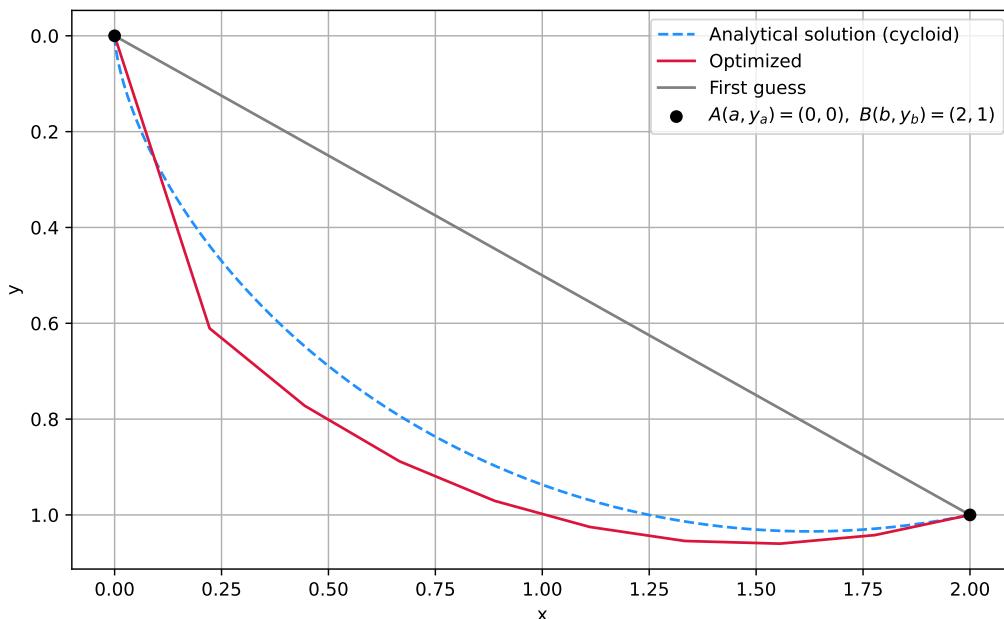


Рис. 9. Первоначальная, оптимизированная и аналитически полученная (циклоида) кривые ( $N = 10$ ,  $n = 10$ )

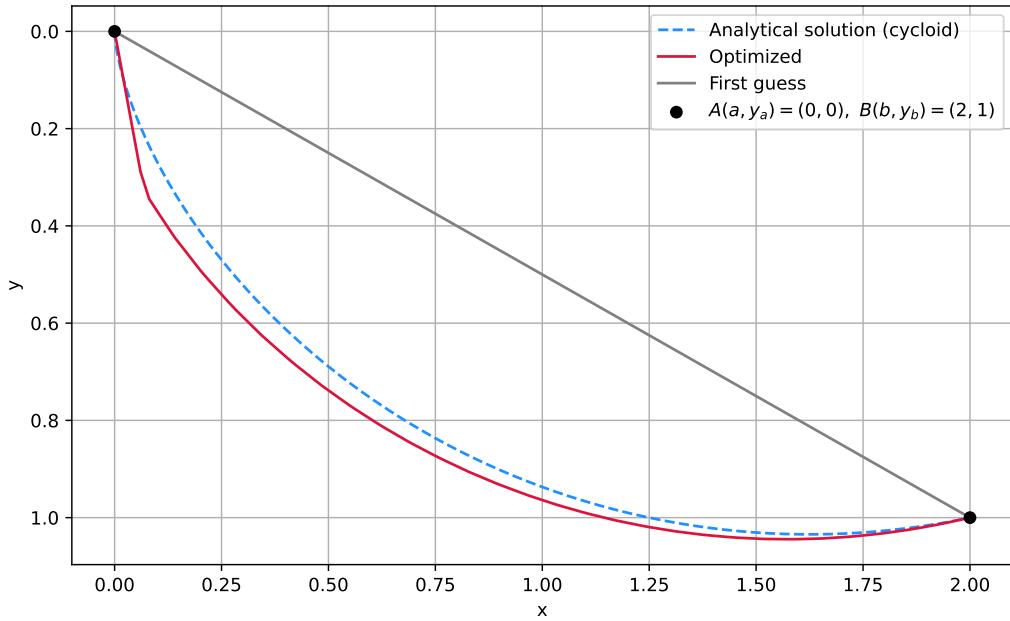


Рис. 10. Первоначальная, оптимизированная и аналитически полученная (циклоида) кривые ( $N = 30$ ,  $n = 30$ )

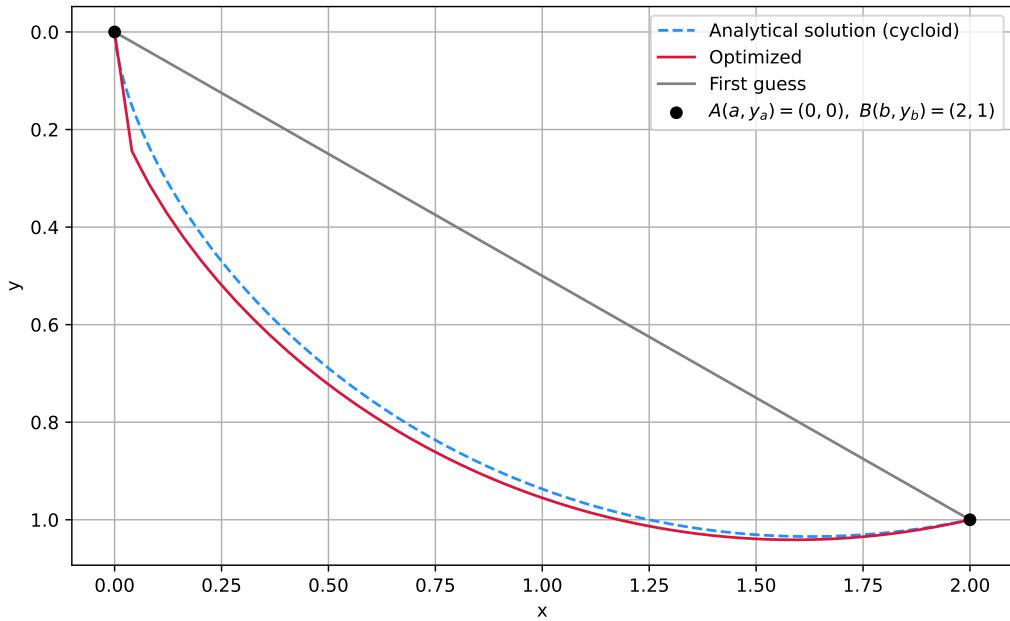


Рис. 11. Первоначальная, оптимизированная и аналитически полученная (циклоида) кривые ( $N = 50$ ,  $n = 50$ )

## 12 Оценка погрешности решения для различных конфигураций дискретизации

Для оценки зависимости погрешности решения от шага интерполяции и шага интегрирования использовалась  $L_2$  норма, она же евклидово расстояние:

$$E(\omega, z) = \left( \sum_{i=1}^n |\omega_i - z_i|^2 \right)^{1/2},$$

где  $\omega = [\omega_1, \dots, \omega_n]$ ,  $z = [z_1, \dots, z_n]$ .

Дальнейший анализ проведен для шагов интерполяции и интегрирования при  $N \in [3, 30]$ ,  $n \in [3, 30]$  в связи с длительным временем работы программы (см. 12).

Время выполнения для N=20: 11.008965003490449 минут  
 Время выполнения для N=21: 14.05917943318685 минут  
 Время выполнения для N=22: 18.040224822362266 минут  
 Время выполнения для N=23: 22.653334267934163 минут  
 Время выполнения для N=24: 28.326477833588918 минут  
 Время выполнения для N=25: 34.85047569274902 минут  
 Время выполнения для N=26: 42.53435918490092 минут  
 Время выполнения для N=27: 51.37312156359355 минут  
 Время выполнения для N=28: 62.052074821790065 минут  
 Время выполнения для N=29: 74.49827739795049 минут  
 Время выполнения для N=30: 89.32869220177332 минут

Рис. 12. Время на текущую итерацию с момента запуска

На рисунке 13 представлены результаты оптимизации для различных параметров дискретизации, где в качестве первоначального предположения кривая  $y(x) = \frac{x}{2}$ .

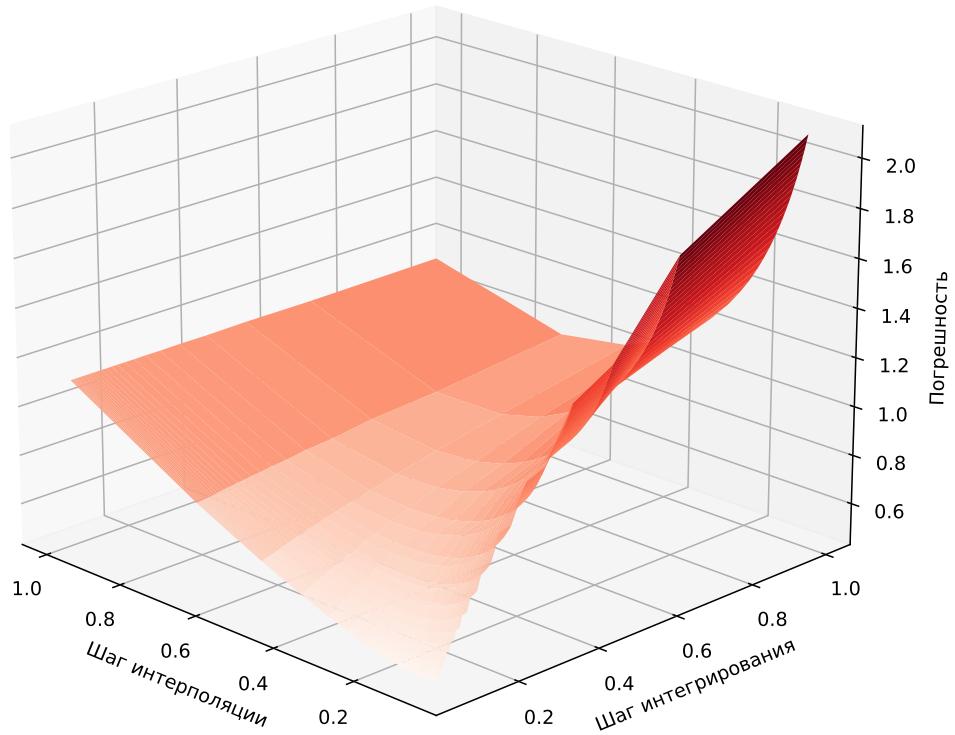


Рис. 13. Зависимость ошибки решения от шагов интегрирования и интерполирования при  $N \in [3; 30]$ ,  $n \in [3; 30]$

Для большей наглядности на рисунке 14 отображены линии уровня графика ошибки решения от шагов интегрирования и интерполирования при  $N \in [3; 30]$ ,  $n \in [3; 30]$ .

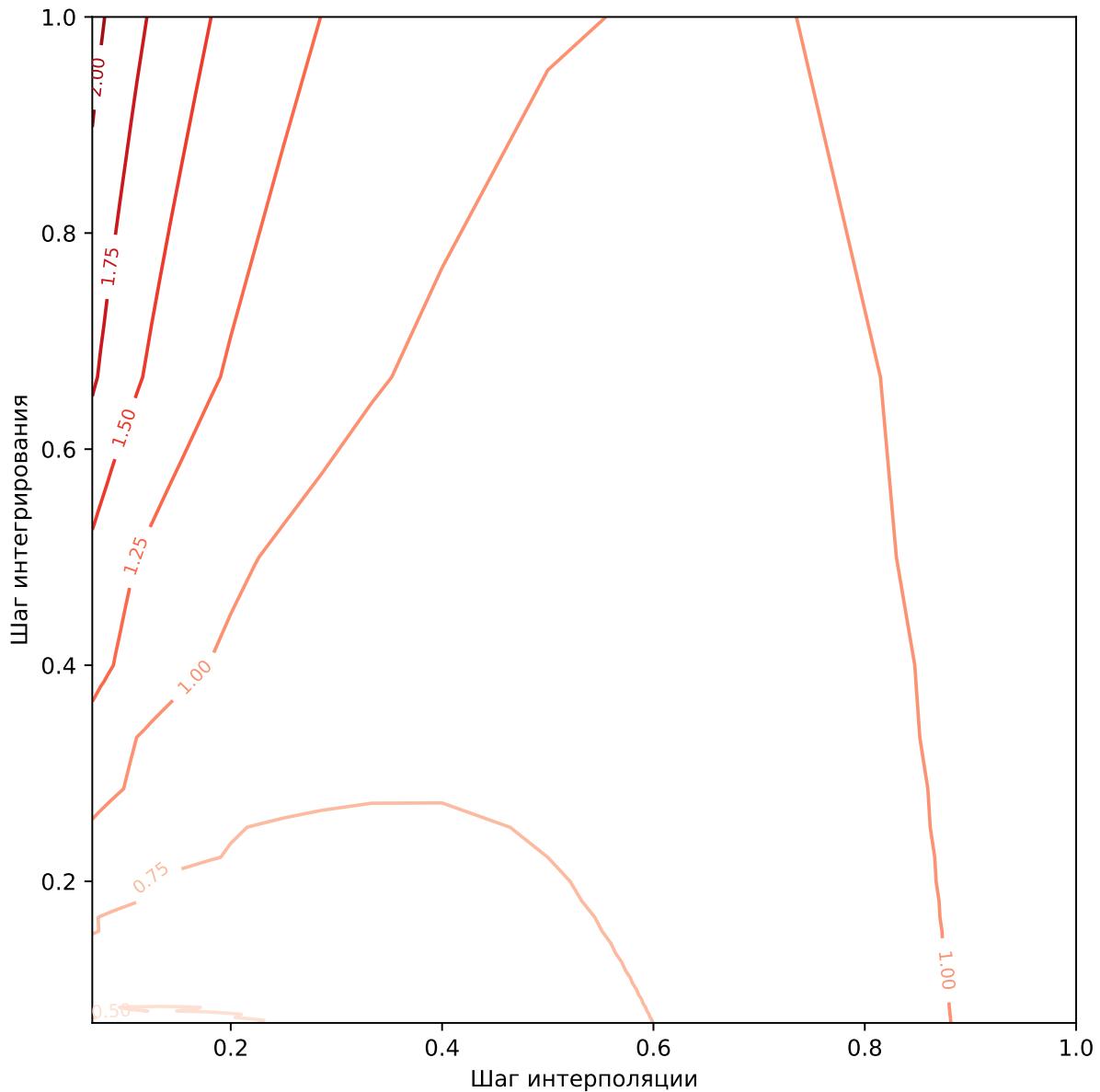


Рис. 14. Линии уровня графика ошибки решения от шагов интегрирования и интерполяции при  $N \in [3; 30]$ ,  $n \in [3; 30]$

Наблюдается значительная зависимость ошибки решения от шага интегрирования. Наилучший результат достигается при минимальном значении обоих шагов.

Вышеописанный анализ характеризует зависимость ошибки решения  $E$  как функцию от двух величин  $H, h$ , где  $H$  - шаг интерполяции,  $h$  - шаг интегрирования.

Тогда проведем анализ функции  $E(H, h)$  при поочередном фиксированном значении одной из переменных.

На рисунке 15 представлен график функции  $E(H = const, h)$ .

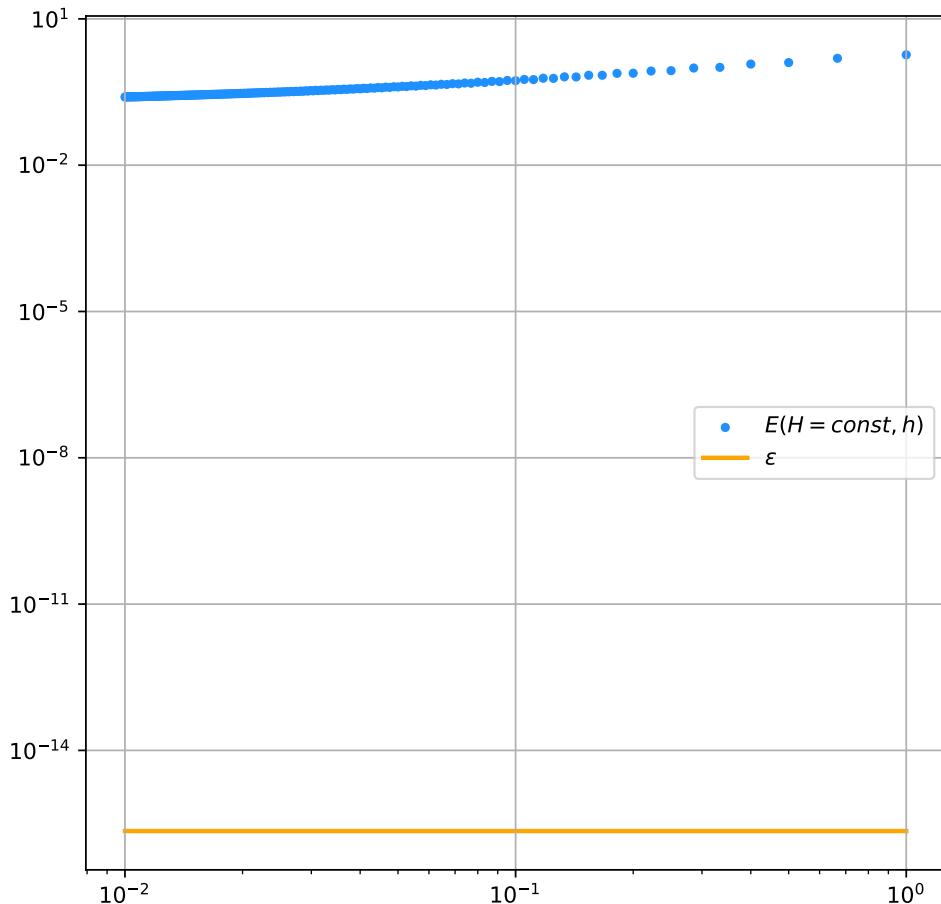


Рис. 15. Зависимость ошибки решения от шага интегрирования при фиксированном шаге интерполирования при  $N = 20$ ,  $n \in [3; 200]$

На рисунке 16 представлен график функции  $E(H, h = const)$ .

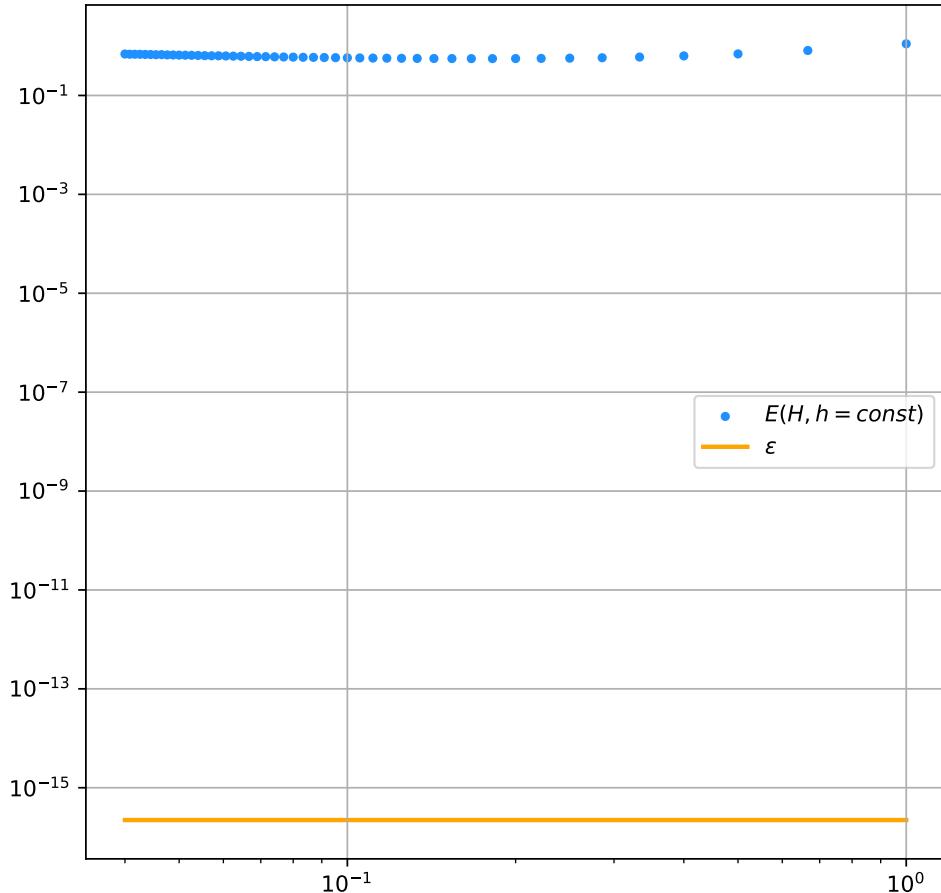


Рис. 16. Зависимость ошибки решения от шага интерполирования при фиксированном шаге интегрирования при  $N \in [3; 50]$ ,  $n = 20$

### 13 Заключение

В ходе предложенной лабораторной работы были реализованы и проанализированы методы численного интегрирования при решении прикладной задачи на примере составных формул Симпсона и трапеции.

Также, с помощью методов аппроксимации, таких как интерполяции и численное интегрирование, был разработан метод для нахождения аппроксимации кривой наискорейшего спуска. Проведен анализ влияния параметров дискретизации на ошибку решения, который показал значительную зависимость от шага интегрирования.

## Список использованных источников

- Першин А.Ю. Лекции по курсу «Вычислительная математика». Москва, 2021. С. 140. URL: <https://archrk6.bmstu.ru/index.php/f/810046>.
- Соколов, А.П. Инструкция по выполнению лабораторных работ (общая). Москва: Соколов, А.П., 2021. С. 9. URL: <https://archrk6.bmstu.ru>. (облачный сервис кафедры РК6).
- Першин А.Ю., Соколов А.П., Гудым А.В. Сборник постановок задач на лабораторные работы по курсу «Вычислительная математика»: Учебное пособие. [Электронный ресурс]. Москва, 2023. С. 47. URL: <https://archrk6.bmstu.ru>. (облачный сервис кафедры РК6).

## Выходные данные

*Гавриш А.А. Отчет о выполнении лабораторной работы №2 по дисциплине «Вычислительная математика». [Электронный ресурс] – Москва: 2023. – 26 с. URL: <https://sa2systems.ru:88> (система контроля версий кафедры РК6)*

Постановка:  ассистент кафедры РК-6, PhD Першин А.Ю.  
 Решение и вёрстка:  студент группы РК6-51Б, Гавриш А.А.

2023, осенний семестр