



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 по дисциплине «Вычислительная математика»

Студент:	Журавлев Николай Вадимович
Группа:	РК6-52Б
Тип задания:	Лабораторная работа
Тема:	Использование аппроксимаций для численной оптимизации

Студент

подпись, дата

Журавлев Н.В.
Фамилия, И.О.

Преподаватель

подпись, дата

Першин А.Ю.
Фамилия, И.О.

Москва, 2021

Содержание

Использование аппроксимаций для численной оптимизации	3
1 Введение	3
2 Цель выполнения лабораторной работы	3
3 Задачи на лабораторную работу	3
4 Выполненные задачи	4
5 Выполнение численного интегрирования с помощью составной Симпсона.	4
6 Выполнение численного интегрирования с помощью составной формулы трапеций.	5
7 Расчёт интеграла для функции, соответствующей кривой наискорейшего спуска.	6
8 Определение порядка точности формулы по полученному графику. . . .	8
9 Сравнение порядка для обеих формул, полученного с помощью графика, с аналитическим порядком точности.	9
10 Существование оптимального шага интегрирования для данной формулы, минимизирующий достижимую погрешность	9
11 Заключение	10

Использование аппроксимаций для численной оптимизации

1 Введение

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 направлена вниз). Решением этой задачи является такая кривая $y(x)$, которая минимизирует следующий функционал, являющийся полным временем движения материальной точки:

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

где g обозначает ускорение свободного падения, $y' = \frac{dy}{dx}$.

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

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

В базовой части 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 $C = 1.03439984$, $T = 1.75418438$.

2 Цель выполнения лабораторной работы

Цель выполнения лабораторной работы – изучение общих принципов численного интегрирования с использованием составных формул Симпсона и трапеций. Рассмотрение одной из задач вариационного исчисления: задачи о брахистохроне, т.е. задачи о кривой наискорейшего спуска. Разработка метода нахождения аппроксимации, в целях решения задачи минимизации.

3 Задачи на лабораторную работу

Базовая часть:

1. Написать функцию `composite_simpson(a, b, n, f)` численного интегрирования функции f на интервале $[a; b]$ по n узлам с помощью составной формулы Симпсона.
2. Написать функцию `composite_trapezoid(a, b, n, f)` численного интегрирования функции f на интервале $[a; b]$ по n узлам с помощью составной формулы трапеций.
3. Рассчитать интеграл $F[y] = \int_0^a \sqrt{\frac{1+(y'(x))^2}{2gy(x)}} dx$ для функции $y(x)$, соответствующей кривой наискорейшего спуска, с помощью составной формулы Симпсона и составной формулы трапеции для множества значений $n \in [3; 9999]$. Постройте log-log график зависимости абсолютной погрешности численного интегрирования от шага
4. интегрирования для обеих формул. Объясните, каким образом по полученному графику можно определить порядок формулы. Для обеих формул сравните порядок, полученный с помощью графика, с аналитическим порядком точности.
5. Существует ли оптимальный шаг интегрирования для данной формулы, минимизирующий достижимую погрешность? Обоснуйте свой ответ.

4 Выполненные задачи

1. Разработана функция для численного интегрирования функции с помощью составной формулы Симпсона.
2. Разработана функция для численного интегрирования функции с помощью составной формулы трапеции.
3. Построен log-log графика зависимости абсолютной погрешности численного интегрирования от шага интегрирования для формулы Симпсона и формулы трапеции.
4. Определение порядка точности формулы по графику. Сравнение порядка полученного с помощью графика, с аналитическим порядком точности.
5. Определение наличия оптимального шага интегрирования для данной формулы, минимизирующий достижимую погрешность.

5 Выполнение численного интегрирования с помощью составной Симпсона.

Разработана функция `composite_simpson(a, b, n, f)` (Листинг 1) для численного интегрирования функции f на интервале $[a; b]$ по n узлам с помощью составной формулы

Симпсона (3).

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

где $x_i = a + (i-1)h$, $h = \frac{b-a}{n}$ и $i = 1, \dots, n+1$.

Листинг 1. Реализация функции *composite_simpson*

```

1 def composite_simpson(a, b, n, f):
2     if n % 2 == 0:
3         n += 1
4     segments = n - 1
5     h = (b - a) / segments
6     simpson = (h / 3) * (f(a) + f(b))
7     for i in range(1, int((segments / 2))):
8         simpson += (h / 3) * 2 * (f(a + (2 * i) * h))
9     for i in range(1, int(segments / 2 + 1)):
10        simpson += (h / 3) * 4 * (f(a + (2 * i - 1) * h))
11    return simpson

```

6 Выполнение численного интегрирования с помощью составной формулы трапеций.

Разработана функция *composite_trapezoid(a, b, n, f)* (Листинг 2) для численного интегрирования функции f на интервале $[a; b]$ по n узлам с помощью составной формулы трапеций (4).

$$\int_a^b f(x)dx = \frac{h}{2} \left[f(x_1) + 2 \sum_{i=2}^n f(x_i) + f(x_{n+1}) \right] - \frac{(b-a)h^2}{12} f''(\varepsilon), \quad (4)$$

где $x_i = a + (i-1)h$, $h = \frac{b-a}{n}$ и $i = 1, \dots, n+1$.

Листинг 2. Реализация функции *composite_trapezoid*

```

1 def composite_trapezoid(a, b, n, f):
2     segments = n - 1
3     h = (b - a) / segments
4     trapezoid = (h / 2) * (f(a) + f(b))
5     for i in range(2, segments + 1):
6         trapezoid += (h / 2) * 2 * (f(a + (i - 1) * h))
7     return trapezoid

```

7 Расчёт интеграла для функции, соответствующей кривой наискорейшего спуска.

Для вычисления значений функций $y(x)$ и $y'(x)$ используются функции `qubic_spline_coeff` (Листинг 3), `qubic_spline` (Листинг 4) и `d_qubic_spline` (Листинг 5). Они вычисляют коэффициенты кубического сплайна, значение кубического сплайна в точке x и производную кубического сплайна соответственно.

Листинг 3. Реализация функции `qubic_spline_coeff`

```
1 def qubic_spline_coef(x_nodes, y_nodes, n):
2     h = [x_nodes[i + 1] - x_nodes[i] for i in range(n - 1)]
3     matrix_a = numpy.diag(numpy.r_[[1], [2 * (h[i + 1] + h[i]) for i in range(len(h) - 1)],
4                                     [1]])
5     matrix_a = matrix_a + numpy.diag(numpy.r_[[0], [h[i] for i in range(1, n - 1)], 1])
6     matrix_a = matrix_a + numpy.diag(numpy.r_[[h[i] for i in range(n - 2)], [0], -1])
7     a = numpy.array(y_nodes)
8
9     matrix_b = numpy.r_[[0], [3 * (a[i + 2] - a[i + 1]) / h[i + 1] - 3 * (a[i + 1] - a[i]) /
10                            h[i] for i in range(n - 2)], [0]]
11     c = numpy.linalg.solve(matrix_a, matrix_b)
12     d = numpy.array([(c[i + 1] - c[i]) / (3 * h[i]) for i in range(n - 1)])
13     b = numpy.array([(a[i + 1] - a[i]) / h[i] - h[i] * (c[i + 1] + 2 * c[i]) / 3 for i in range(n
14                     - 1)])
15     qs_coeff = np.concatenate([a, b, c, d])
16     return qs_coeff
```

Листинг 4. Реализация функции `qubic_spline`

```
1 def qubic_spline(x, x_nodes, qs_coeff, n):
2     index = binary_search(x_nodes, x)
3     s = qs_coeff[index] + qs_coeff[index + n] * (x - x_nodes[index]) + qs_coeff[index + 2
4         * n - 1] * (
5         x - x_nodes[index]) ** 2 + \
6         qs_coeff[index + 3 * n - 1] * (x - x_nodes[index]) ** 3
7     return s
```

Листинг 5. Реализация функции `d_qubic_spline`

```
1 def d_qubic_spline(x, x_nodes, qs_coeff, n):
2     index = binary_search(x_nodes, x)
3     d_s = qs_coeff[index + n] + 2 * qs_coeff[index + 2 * n - 1] * (x - x_nodes[index])
4         + 3 * qs_coeff[
5         index + 3 * n - 1] * (x - x_nodes[index]) ** 2
6     return d_s
```

В функцию `cubic_spline_coeff` передаются узлы $x(t)$, $y(t)$ (оба вычисленные по формуле (2)) и количество узлов (переменная `n_s`). Функция возвращает коэффициенты a_i, b_i, c_i, d_i в массив `qs_coeff`.

В функцию `f` (Листинг 6) передаётся массив `qs_coeff`. В ней вычисляется значение $f(y(x))$ из формулы

$$F(y) = \sqrt{\frac{1 + (y'(x))^2}{2gy(x)}}, \quad (5)$$

что является подынтегральным выражением формулы (1).

Листинг 6. Реализация функции `d_cubic_spline`

```

1 def d_cubic_spline(x, x_nodes, qs_coeff, n):
2     index = binary_search(x_nodes, x)
3     d_s = qs_coeff[index + n] + 2 * qs_coeff[index + 2 * n - 1] * (x - x_nodes[index])
4         + 3 * qs_coeff[index + 3 * n - 1] * (x - x_nodes[index]) ** 2
5     return d_s

```

Расчёт интеграла (5) для функции $y(x)$ с помощью функций, реализующих составные формулы Симпсона и трапеций, для множества значений $n \in [3; 9999]$ узлов - листинг 7. Полученные значения интеграла используются для нахождения абсолютной погрешности численного интегрирования для обеих формул. В качестве абсолютного значения взято значение составной формулы Симпсона для интеграла (1), где $n = 20000$.

Листинг 7. Вычисление абсолютной погрешности для формул составных Симпсона и трапеций

```

1 a = 0.01
2 b = 2
3 int_simp = [0] * 5000
4 int_trap = [0] * 5000
5 h_integral = [0] * 5000
6 i = 0
7 true_integral = composite_simpson(a, b, 20000, f)
8 for n in range(3, 10000, 2):
9     h_integral[i] = (b - a) / (n - 1)
10    int_simp[i] = abs(true_integral - composite_simpson(a, b, n, f))
11    int_trap[i] = abs(true_integral - composite_trapezoid(a, b, n, f))
12    i += 1

```

Был построен log-log график зависимости абсолютной погрешности численного интегрирования от шага интегрирования для обеих формул (Рисунок 1).

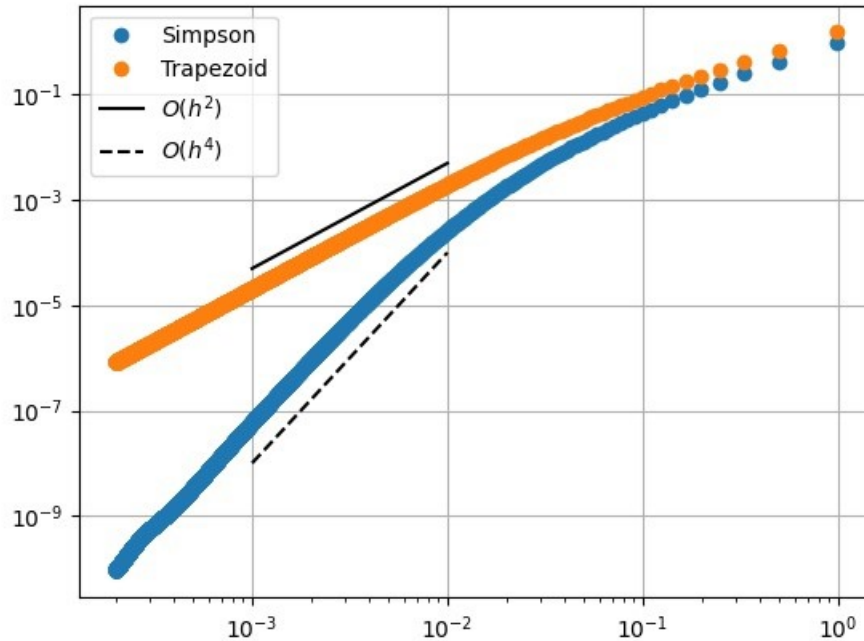


Рис. 1. График зависимости абсолютной погрешности E численного интегрирования для составной формулы Симпсона и составной формулы трапеций.

8 Определение порядка точности формулы по полученному графику.

Порядок точности численного метода – это наибольшая степень полинома, для которой метод даёт точное решение задачи. Для определения порядка точности численных методов интегрирования по графику. Для каждого из графиков определить тангенс угла наклона (он показывает зависимость абсолютных погрешностей, использованных в численных методах интегрирования). Это справедливо, так как график построен по шкале с логарифмической зависимостью.

Вычисление порядка точности для формулы Симпсона:

$$\frac{\lg 10^{-4} - \lg 10^{-8}}{\lg 10^{-2} - \lg 10^{-3}} = 4,$$

Вычисление порядка точности для формулы трапеций:

$$\frac{\lg 10^{-2} - \lg 10^{-4}}{\lg 10^{-2} - \lg 10^{-3}} = 2.$$

9 Сравнение порядка для обеих формул, полученного с помощью графика, с аналитическим порядком точности.

Аналитический порядок точности для составной формулы Симпсона пропорционален $O(h^4)$:

$$\frac{(b-a)h^4}{180} f^{(4)}(\varepsilon).$$

Аналитический порядок точности для составной формулы трапеций пропорционален $O(h^2)$:

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

По рисунку 1 можно сделать вывод, что абсолютная погрешность численного интегрирования от шага интегрирования для составной формулы Симпсона и формулы составной трапеций пропорциональная их аналитическому порядку точности. То есть, для составной формулы Симпсона это $O(h^4)$, а для составной формулы трапеций - $O(h^2)$.

10 Существование оптимального шага интегрирования для данной формулы, минимизирующий достижимую погрешность

Пусть значение $f(x)$ в точке x_i вычисляется с погрешностью округления e_i :

$$f(x_i) = f(x_i) + e_i, i = 1, \dots, n+1.$$

Следовательно, полная погрешность округления, накапливаемая составной формулой Симпсона, оценивается следующим образом:

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

Пусть погрешность округления ограничена машинным эпсилон:

$$|e_i| \leq \epsilon, i = 1, \dots, n+1.$$

Тогда полная погрешность оценивается, как:

$$(e_h) \leq \frac{h\epsilon}{3} (1 + 2(\frac{n}{2} - 1) + 4\frac{n}{2} + 1) = nh\epsilon = (b-a)\epsilon$$

Верхняя грань для накопленной погрешности округления не зависит от n и h . Это означает, что увеличение числа под-отрезков не приводит к дестабилизации полной погрешности. Минимальная разность двух значений в компьютере - это машинное эпсилон, то есть минимизацию значения шага интегрирования следует производить до тех пор, пока абсолютная погрешность функции не станет равна машинному эпсилон.

11 Заключение

В процессе выполнения лабораторной работы было изучено и применено:

1. Численное интегрирование на примере составных формул Симпсона и трапеции.
2. Была оценена зависимость погрешности вычисления интеграла численным интегрированием от шага интегрирования.
3. Был сделан вывод от том, что недостаточность гладкости функции приводит к изменению порядка точности на графике.
4. Было показано, что оптимальный шаг интегрирования, зависит от машинного эпсилон

Список использованных источников

1. Першин А.Ю. Лекции по курсу «Вычислительная математика». Москва, 2018-2021. С. 140.

Выходные данные

Журавлев Н.В.. Отчет о выполнении лабораторной работы по дисциплине «Вычислительная математика». [Электронный ресурс] — Москва: 2021. — 10 с. URL: <https://sa2systems.ru:88> (система контроля версий кафедры РК6)

Постановка:



ассистент кафедры РК-6, PhD А.Ю. Першин

Решение и вёрстка:



студент группы РК6-52Б, Журавлев Н.В.

2021, осенний семестр