



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:	Горелкина Елизавета Егоровна
Группа:	РК6-52Б
Тип задания:	лабораторная работа
Тема:	Интерполяция в условиях измерений с неопределенностью

Студент

\_\_\_\_\_  
подпись, дата

Горелкина Е.Е.  
\_\_\_\_\_  
Фамилия, И.О.

Преподаватель

\_\_\_\_\_  
подпись, дата

\_\_\_\_\_  
Фамилия, И.О.

Москва, 2021

# Содержание

<b>Интерполяция в условиях измерений с неопределенностью</b>	<b>3</b>
1    Задание . . . . .	3
2    Цель выполнения лабораторной работы . . . . .	5
3    Выполненные задачи . . . . .	5
4    Разработка функции вычисления коэффициентов кубического сплайна .	6
5    Разработка функций вычисления значения кубического сплайна и его производной в заданной точке . . . . .	9
6    Построение аппроксимации по заданным узлам с помощью кубического сплайна . . . . .	11
7    Разработка функции вычисления значения $i$ -го базисного полинома Лагран- жа . . . . .	12
8    Разработка функции вычисления значения интерполяционного полино- ма Лагранжа . . . . .	13
9    Анализ влияния погрешности измерений . . . . .	15
Часть 1. Погрешность измерений. Метод Монте-Карло . . . . .	15
Часть 2. Построение доверительной полосы и усредненного интерполянта	18
Часть 3. Анализ влияния погрешностей на результаты интерполяции . .	24
10   Заключение . . . . .	29

# Интерполяция в условиях измерений с неопределенностью

## 1 Задание

Базовая часть:

1. Разработать функцию `qubic_spline_coeff(x_nodes, y_nodes)`, которая посредством решения матричного уравнения вычисляет коэффициенты естественного кубического сплайна.

2. Написать функции `qubic_spline(x, x_nodes, qs_coeff)` и `d_qubic_spline(x, x_nodes, qs_coeff)`, которые вычисляют соответственно значение кубического сплайна и его производной в точке  $x$  (`qs_coeff` обозначает матрицу коэффициентов).

3. Используя данные в таблице 1, требуется построить аппроксимацию зависимости уровня поверхности жидкости  $h(x)$  от координаты  $x$  (рис. 1) с помощью кубического сплайна и продемонстрировать ее на графике вместе с исходными узлами.

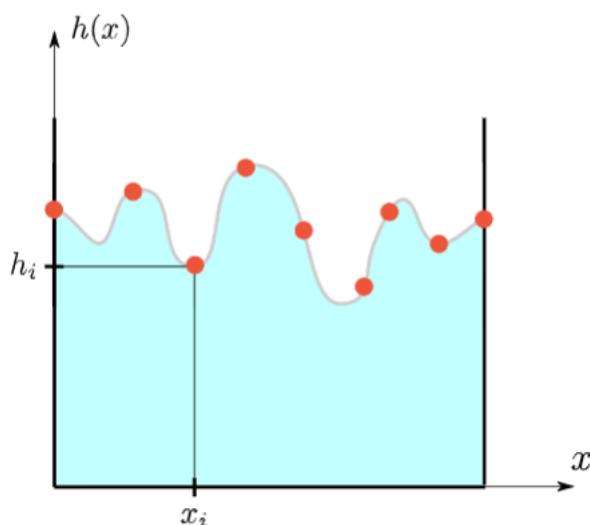


Рисунок 1. Поверхность вязкой жидкости (серая кривая), движущейся сквозь некоторую среду (например, пористую). Её значения известны только в нескольких точках (красные узлы)

Таблица 1: Значения уровня поверхности вязкой жидкости

$x_i$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$h_i$	3.37	3.95	3.73	3.59	3.15	3.15	3.05	3.86	3.60	3.70	3.02

Продвинутая часть:

1. Разработать функцию  $l\_i(i, x, x\_nodes)$ , которая возвращает значение  $i$ -го базисного полинома Лагранжа, заданного на узлах с абсциссами  $x\_nodes$ , в точке  $x$ .

2. Написать функцию  $L(x, x\_nodes, y\_nodes)$ , которая возвращает значение интерполяционного полинома Лагранжа, заданного на узлах с абсциссами  $x\_nodes$  и ординатами  $y\_nodes$ , в точке  $x$ .

3. Известно, что при измерении координаты  $x$  всегда возникает погрешность, которая моделируется случайной величиной с нормальным распределением с нулевым математическим ожиданием и стандартным отклонением  $10^{-2}$ . Требуется провести следующий анализ, позволяющий выявить влияние этой погрешности на интерполяцию:

а) Сгенерировать 1000 векторов значений  $[\tilde{x}_1, \dots, \tilde{x}_{11}]^T$  предполагая, что  $\tilde{x}_i = x_i + Z$ , где  $x_i$  соответствует значению в таблице 1 и  $Z$  является случайной величиной с нормальным распределением с нулевым математическим ожиданием и стандартным отклонением  $10^{-2}$ .

б) Для каждого из полученных векторов построить интерполянт Лагранжа, предполагая, что в качестве абсцисс узлов используются значения  $\tilde{x}_i$ , а ординат —  $h_i$  из таблицы 1. В результате вы должны иметь 1000 различных интерполянтов.

в) Предполагая, что все интерполянты представляют собой равновероятные события, построить такие функции  $\tilde{h}_l(x)$  и  $\tilde{h}_u(x)$ , где  $\tilde{h}_l(x) < \tilde{h}_u(x) \forall x \in [0, 1]$ , что вероятность того, что значение интерполянта в точке будет лежать в интервале  $[\tilde{h}_l(x), \tilde{h}_u(x)]$ , равна 0.9.

г) Отобразить на едином графике функции  $h_l(x)$ ,  $h_u(x)$ , усредненный интерполянт и узлы из таблицы 1.

е) Какие участки интерполянта и почему являются наиболее чувствительными к погрешностям?

4. Повторить анализ, описанный в предыдущем пункте, в предположении, что координаты  $x_i$  вам известны точно, в то время как измерения уровня поверхности  $h_i$  имеют ту же погрешность, что и в предыдущем пункте. Изменились ли выводы вашего анализа?

5. Повторить два предыдущих пункта для случая интерполяции кубическим сплайном. Какие выводы вы можете сделать, сравнив результаты анализа для интерполяции Лагранжа и интерполяции кубическим сплайном?

## 2 Цель выполнения лабораторной работы

В базовой части данной работы предлагается познакомиться с интерполяцией в целом на примере интерполяции кубическим сплайном.

В продвинутой части предлагается проанализировать, как неопределенности влияют на результат интерполяции, используя интерполяцию сплайном и интерполяцию полиномом Лагранжа, и сравнить результаты анализа для двух указанных методов.

## 3 Выполненные задачи

1. На языке программирования Python была написана функция, вычисляющая коэффициенты естественного кубического сплайна.

2. На языке программирования Python были написаны две функции: первая вычисляет значение кубического сплайна в заданной точке, вторая вычисляет производную кубического сплайна в заданной точке.

3. Была проведена и продемонстрирована на графике аппроксимация неизвестной зависимости с помощью кубического сплайна по узлам, представленным в таблице 1.

4. На языке программирования Python была написана функция, возвращающая значение  $i$ -го базисного полинома Лагранжа.

5. На языке программирования Python была написана функция, возвращающая значение в точке  $x$  интерполяционного полинома Лагранжа, заданного в узлах, представленных в таблице 1.

6. Был проведен следующий анализ влияния погрешности измерений на интерполяцию:

- В первой части анализа были подготовлены данные: сгенерированы 1000 векторов узлов с зашумленными данными для всех четырех случаев, предложенных в задании (для интерполянта Лагранжа зашумление данных  $x_i$  или  $h_i$ , аналогично для кубического сплайна). Также были построены семейства интерполянтов для всех четырех предложенных в задании случаев и продемонстрированы на графиках.
- Во второй части была построена доверительная полоса с уровнем доверия 0.9 для всех четырех случаев. Был подробно описан и обоснован метод построения такой полосы. Тут же был построен усредненный интерполянт и представлены четыре графика, на которых изображены сразу доверительная полоса и усредненный интерполянт.
- В третьей части был проведен анализ влияния погрешностей на результаты интерполяции, было выяснено, как разные погрешности влияют на разные методы интерполяции.

## 4 Разработка функции вычисления коэффициентов кубического сплайна

Кубический сплайн является примером локальной интерполяции, что будет видно далее из его определения, которое понадобится нам для вычисления его коэффициентов.

**Опр.** Пусть задана функция  $f(x)$  в  $n$  интерполяционных узлах  $a = x_1, x_2, \dots, x_n = b$  на отрезке  $[a, b]$ . Кубическим сплайном для  $f(x)$  называется функция  $S(x)$ , для которой выполнено следующее:

- $S(x)$  кусочно задана кубическими многочленами  $S_i(x) \forall [x_i, x_{i+1}]$ ,  $i \in [1, n-1]$ ;
- $S_i(x_i) = f(x_i)$ ,  $S_i(x_{i+1}) = f(x_{i+1})$ ,  $i \in [1, n-1]$ ;
- $S_i(x_{i+1}) = S_{i+1}(x_{i+1})$ ,  $i \in [1, n-2]$ ;
- $S'_i(x_{i+1}) = S'_{i+1}(x_{i+1})$ ,  $i \in [1, n-2]$ ;
- $S''_i(x_{i+1}) = S''_{i+1}(x_{i+1})$ ,  $i \in [1, n-2]$ ;
- граничные условия  $S''(x_1) = S''(x_n) = 0$ ,  $S'(x_1) = f'(x_1)$ ,  $S'(x_n) = f'(x_n)$ .

Кубический полином задается 4 константами, поэтому для кубического сплайна нам необходимо  $4(n-1)$  коэффициентов:

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3. \quad (1)$$

На данном этапе можно заметить, что подход локальной интерполяции, в частности кубическим сплайном, имеет важное преимущество перед глобальной: увеличение количества узлов не ведет к увеличению степени полинома – он всегда остается полиномом третьей степени.

Из формулы (1) видно, что  $S_i(x_i) = a_i = f(x_i)$ . Далее, как было показано в лекциях [1], можно получить следующие рекуррентные формулы для  $b_i$  и  $d_i$ , исходя из определения кубического сплайна.

$$b_i = \frac{1}{h_i}(a_{i+1} - a_i) - \frac{h_i}{3}(c_{i+1} + 2c_i), \quad (2)$$

$$d_i = \frac{c_{i+1} - c_i}{3h_i}, \quad (3)$$

где  $h_i = x_{i+1} - x_i$ .

В итоге для нахождения векторов коэффициентов  $b_i$  и  $d_i$  необходимо найти вектор коэффициентов  $c_i$ . Используя формулы из лекций, запишем рекуррентное уравнение относительно  $c_{i-1}$ ,  $c_i$  и  $c_{i+1}$ :

$$h_{i-1}c_{i-1} + 2(h_i + h_{i-1})c_i + h_ic_{i+1} = \frac{3}{h_{i-1}}(a_i - a_{i-1}) - \frac{3}{h_{i-2}}(a_{i-1} - a_{i-2}). \quad (4)$$

Тогда коэффициенты кубического сплайна можно получить, решая следующее матричное уравнение  $\mathbf{A}\mathbf{c} = \mathbf{b}$ ,  $\mathbf{c} = [c_1, \dots, c_n]^T$ :

$$\begin{pmatrix} 1 & 0 & \dots & \dots & \dots & 0 \\ h_1 & 2(h_2 + h_1) & h_2 & 0 & \dots & 0 \\ 0 & h_2 & 2(h_3 + h_2) & h_3 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & h_{n-2} & 2(h_{n-2} - h_{n-1}) & h_{n-1} \\ 0 & \dots & \dots & \dots & \dots & 1 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_{n-1} \\ c_n \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{3}{h_2}(a_3 - a_2) - \frac{3}{h_1}(a_2 - a_1) \\ \frac{3}{h_3}(a_4 - a_3) - \frac{3}{h_2}(a_3 - a_2) \\ \vdots \\ \frac{3}{h_{n-1}}(a_n - a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) \\ 0 \end{pmatrix},$$

где

$$h_i = x_{i+1} - x_i. \quad (5)$$

Решим данную систему методом прогонки [2], полагая что  $c_i = z_i - \mu_i c_{i+1}$  и  $\frac{3}{h_i}(a_{i+1} - a_i) - \frac{3}{h_{i-1}}(a_i - a_{i-1}) = \alpha_i$ . Тогда выразим  $c_{i-1}$  через  $c_{i+1}$ :

$$c_{i-1} = z_{i-1} - \mu_{i-1} c_i = z_{i-1} - \mu_{i-1} z_i + \mu_{i-1} \mu_i c_{i+1},$$

подставим  $a_{i-1}, c_{i-1}, c_i$  в (4):

$$h_{i-1}(z_{i-1} - \mu_{i-1} z_i + \mu_{i-1} \mu_i c_{i+1}) + 2(h_i + h_{i-1})(z_i - \mu_i c_{i+1}) + h_i c_{i+1} = \alpha_{i-1},$$

$$[h_{i-1} \mu_{i-1} \mu_i - 2(h_i + h_{i-1}) \mu_i + h_i] c_{i+1} + h_{i-1} z_{i-1} - h_{i-1} \mu_{i-1} z_i + 2(h_i + h_{i-1}) z_i - \alpha_{i-1} = 0,$$

данное соотношение будет выполнено, независимо от  $c_{i+1}$ , если будут выполнены два следующих условия:

$$h_{i-1} \mu_{i-1} \mu_i - 2(h_i + h_{i-1}) \mu_i + h_i = 0,$$

$$h_{i-1} z_{i-1} - h_{i-1} \mu_{i-1} z_i + 2(h_i + h_{i-1}) z_i - \alpha_{i-1} = 0,$$

откуда получаем

$$\mu_i = \frac{h_i}{2(h_i + h_{i-1}) - h_{i-1} \mu_{i-1}}$$

$$z_i = \frac{\alpha_{i-1} - h_{i-1} z_{i-1}}{2(h_i + h_{i-1}) - h_{i-1} \mu_{i-1}},$$

и, используя следующее соотношение

$$h_i + h_{i-1} = x_{i+1} - x_{i-1},$$

введем еще один коэффициент

$$l_i = 2(x_{i+1} - x_{i-1}) - h_{i-1} \mu_{i-1}, \quad (6)$$

окончательно получая

$$\mu_i = \frac{h_i}{l_i}, \quad z_i = \frac{\alpha_{i-1} - h_{i-1}z_{i-1}}{l_i}. \quad (7)$$

Теперь можем реализовать функцию для вычисления коэффициентов кубического сплайна в заданных узлах, используя формулы 2-7 и полагая, что  $l_1 = l_n = 1$ ,  $\mu_1 = \mu_n = 0$  и  $z_1 = z_n = 0$ . Листинг кода представлен ниже, все обозначения, приведенные выше, сохранены.

**Функция** `qubic_spline_coeff(x_nodes, y_nodes)` **получает** на вход два параметра: списки с узлами `x_nodes` и значениями  $f(x)$  в узлах `y_nodes`, – и **возвращает** матрицу  $(n - 1) \times 4$ , столбцами которой являются соответствующие векторы коэффициентов кубического сплайна. Матрица получается путем применения метода `np.c_[a, b, c, d]` из модуля `numpy`. Данный метод осуществляет конкатенацию списков по столбцам, в результате чего в возвращаемой матрице в  $i$ -ой строке расположены коэффициенты  $a_i, b_i, c_i, d_i$ .

Листинг 1. Коэффициенты сплайна

---

```
def qubic_spline_coeff(x_nodes, y_nodes):
    n = len(x_nodes)
    a = list(y_nodes)
    h = [x_nodes[i + 1] - x_nodes[i] for i in range(n - 1)]

    alph = [3 / h[i] * (a[i + 1] - a[i])
             - 3 / h[i - 1] * (a[i] - a[i - 1]) for i in range(1, n - 1)]

    l = [1 for i in range(n)]
    mu = [0 for i in range(n - 1)]
    z = [0 for i in range(n)]

    for i in range(1, n - 1):
        l[i] = 2 * (x_nodes[i + 1] - x_nodes[i - 1]) - h[i - 1] * mu[i - 1]
        mu[i] = h[i] / l[i]
        z[i] = (alph[i - 1] - h[i - 1] * z[i - 1]) / l[i]

    c = [0 for i in range(n)]
    b = [0 for i in range(n - 1)]
    d = [0 for i in range(n - 1)]

    for j in range(n - 1, 0, -1):
        c[j - 1] = z[j - 1] - mu[j - 1] * c[j]
        b[j - 1] = ((a[j] - a[j - 1]) / h[j - 1]
                    - h[j - 1] * (c[j] + 2 * c[j - 1]) / 3)
        d[j - 1] = (c[j] - c[j - 1]) / 3 / h[j - 1]

    c.pop(-1)
    a.pop(-1)

    matr = np.c_[a, b, c, d]

    return matr
```

---



## 5 Разработка функций вычисления значения кубического сплайна и его производной в заданной точке

Получив матрицу с коэффициентами кубического сплайна из предыдущей функции, реализуем функцию вычисления значения кубического сплайна. По определению кубический сплайн существует только на промежутке  $[x_0, x_n]$ , где  $x_0$  и  $x_n$  – граничные узлы интерполяции. Однако, как мы увидим далее при моделировании зашумленных данных и построении доверительной полосы, нам потребуется экстраполировать сплайн за границы указанного выше промежутка.

Так как в задании не оговаривается выбор метода экстраполяции, то будем просто доопределять значения, используя коэффициенты для двух крайних промежутков. В итоге получим следующее условие на узлы, выходящие за границы интерполяции:

$$S(x) = \begin{cases} a_1 + b_1(x - x_1) + c_1(x - x_1)^2 + d_1(x - x_1)^3, & x < x_1 \\ a_{n-1} + b_{n-1}(x - x_1) + c_{n-1}(x - x_1)^2 + d_{n-1}(x - x_1)^3, & x > x_n \end{cases}. \quad (8)$$

Листинг кода функции, вычисляющей значение сплайна в данной точке, приведен ниже. **Функция** `qubic_spline(x, x_nodes, qs_coeff)` **получает** на вход: точку  $x$ , в которой нужно найти значение сплайна, список значений `x_nodes` в заданных узлах и матрицу коэффициентами сплайна `qs_coeff`, – **возвращает** значение сплайна в точке  $x$ .

Листинг 2. Значение сплайна в заданной точке

---

```
def qubic_spline(x, x_nodes, qs_coeff):  
    for i in range(len(x_nodes) - 1):  
        if x <= x_nodes[i + 1]:  
            break  
  
    return (qs_coeff[i][0] + qs_coeff[i][1] * (x - x_nodes[i])  
            + qs_coeff[i][2] * (x - x_nodes[i]) ** 2  
            + qs_coeff[i][3] * (x - x_nodes[i]) ** 3)
```

---

В данной функции в цикле `for` происходит перебор всех промежутков, на которых определен кубический сплайн. Если значение выходит за пределы интерполируемого отрезка, то функция прерывается либо на самой первой итерации, либо доходит до конца, тем самым реализуется экстраполяция 8.

Производная кубического сплайна в точке  $x$ :  $S'(x) = b_i + 2c_i(x - x_i) + 3d_i(x - x_i)^2$ .

Значения производной нет необходимости экстраполировать для дальнейшей работы.

Листинг кода функции, вычисляющей значение производной кубического сплайна в заданной точке, представлен ниже. **Функция** `d_qubic_spline(x, x_nodes, qs_coeff)` **получает** на вход все то же самое, что и функция `qubic_spline(x, x_nodes, qs_coeff)`, и **возвращает** значение производной сплайна в точке  $x$ .

### Листинг 3. Производная сплайна в заданной точке

---

```
def d_qubic_spline(x, x_nodes, qs_coeff):  
    for i in range(len(x_nodes) - 1):  
        if x <= x_nodes[i+1]:  
            return (qs_coeff[i][1] + 2 * qs_coeff[i][2] * (x - x_nodes[i])  
                    + 3 * qs_coeff[i][3] * (x - x_nodes[i]) ** 2)  
  
    return None
```

---

В случае непопадания в промежуток функция возвращает None.

## 6 Построение аппроксимации по заданным узлам с помощью кубического сплайна

У нас уже имеются все необходимые функции для построения графика сплайна по заданным узлам. Вычислим матрицу с коэффициентами, сгенерируем список из  $n$  (в нашем случае возьмем  $n = 201$ ) точек, равномерно расположенных на отрезке  $[0, 1]$ , и по этому списку вычислим  $n$  значений сплайна. Воспользуемся модулем `matplotlib.pyplot` и построим аппроксимацию по заданным узлам из таблицы 1 в виде графика, на котором отобразим также касательные в узлах  $[x_2, x_{n-1}]$ , демонстрируя работоспособность функции, вычисляющей производную сплайна (рис. 2).

Листинг 4. Построение сплайна

```
n = 200
matr = cubic_spline_coeff(x_n, y_n)
x = [i / n for i in np.arange(0, n + 1)]
q = [cubic_spline(x[i], x_n, matr) for i in range(0, n + 1)]
```

Стоит отметить, что на графике отображены только  $n - 2$  касательных к сплайну, так как, вообще говоря, для существования производной  $S(x)$  в точке  $x$  необходимо и достаточно наличие у  $S(x)$  равных друг другу левой и правой производной в этой точке, то есть  $S'(x) = S'_+(x) = S'_-(x)$ . Но в силу того, что сплайн за границами интерполируемого отрезка не определен, то сказать что-то про производную слева для первого узла и производную справа для последнего узла мы точно не можем.

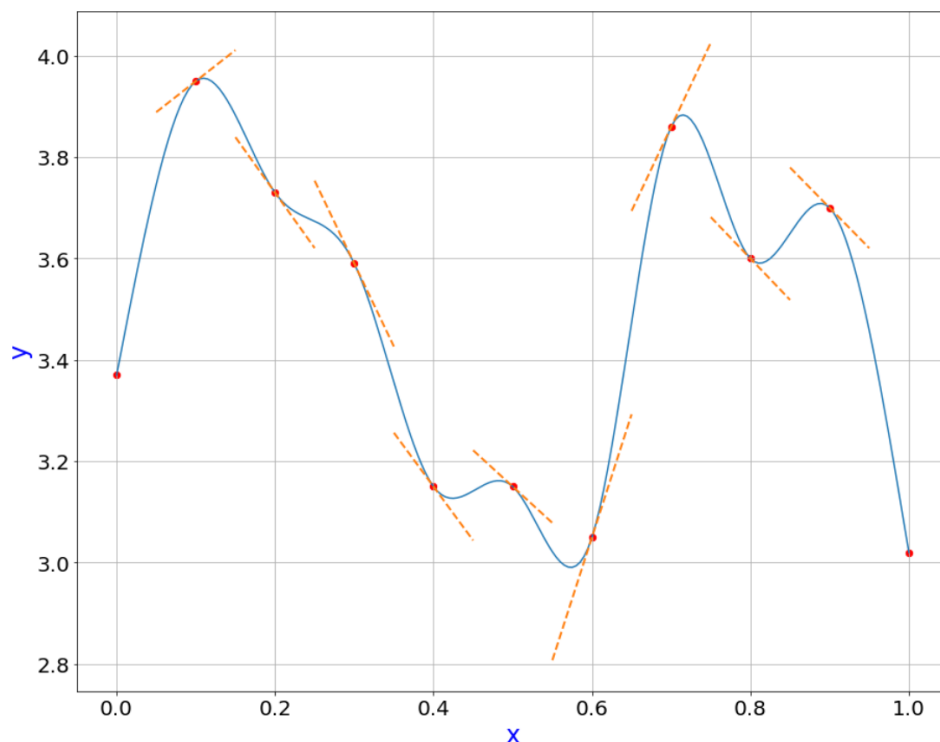


Рисунок 2. Кубический сплайн, узлы и касательные в узлах

## 7 Разработка функции вычисления значения $i$ -го базисного полинома Лагранжа

Приступим к реализации метода глобальной интерполяции на примере интерполяции полиномом Лагранжа. Базисный полином Лагранжа  $n - 1$  степени:

$$l_i(x) = \prod_{i=1, i \neq j}^n \frac{x - x_j}{x_i - x_j}. \quad (9)$$

Листинг функции представлен ниже. **Функция** `l_i(i, x, x_nodes)` **получает** на вход: номер полинома  $i$ , точку  $x$ , в которой нужно вычислить значение базисного полинома, и список `x_nodes` со значениями узлов, – и **возвращает** значение  $i$ -го базисного полинома Лагранжа в заданной точке. Вычисление значения происходит в цикле `for`, где проверяется условие неравенства индексов  $i$  и  $j$  согласно формуле 9.

Листинг 5. Базисный полином Лагранжа

---

```
def l_i(i, x, x_nodes):
    n = len(x_nodes)
    p = 1

    for j in range(n):
        if j != i:
            p *= (x - x_nodes[j]) / (x_nodes[i] - x_nodes[j])

    return p
```

---

## 8 Разработка функции вычисления значения интерполяционного полинома Лагранжа

**Опр.** Пусть функция  $f(x)$  задана в  $n$  интерполяционных узлах  $x_1, x_2, \dots, x_n$  на отрезке  $[a, b]$ , т.е.  $x_1 = a$  и  $x_n = b$ . Тогда интерполяционным полиномом для функции  $f(x)$  и соответствующих узлов интерполяции называется функция

$$L_{n-1}(x) = \sum_{i=1}^n f(x_i)l_i(x),$$

где  $l_i(x)$  - базисный полином Лагранжа.

Листинг кода представлен ниже. **Функция**  $L(x, x\_nodes, y\_nodes)$  **получает** на вход: значение  $x$ , в котором нужно вычислить значение полинома Лагранжа, и списки с узлами  $x\_nodes$  и значениями  $f(x)$   $y\_nodes$ , – **возвращает** значение полинома Лагранжа в точке  $x$ .

Листинг 6. Значение полинома Лагранжа в заданной точке

---

```
def L(x, x_nodes, y_nodes):
    n = len(x_nodes)
    s = 0

    for i in range(n):
        s += y_nodes[i] * l_i(i, x, x_nodes)

    return s
```

---

Изобразим на одной плоскости получившийся полином для заданных узлов вместе с кубическим сплайном (рис. 3).

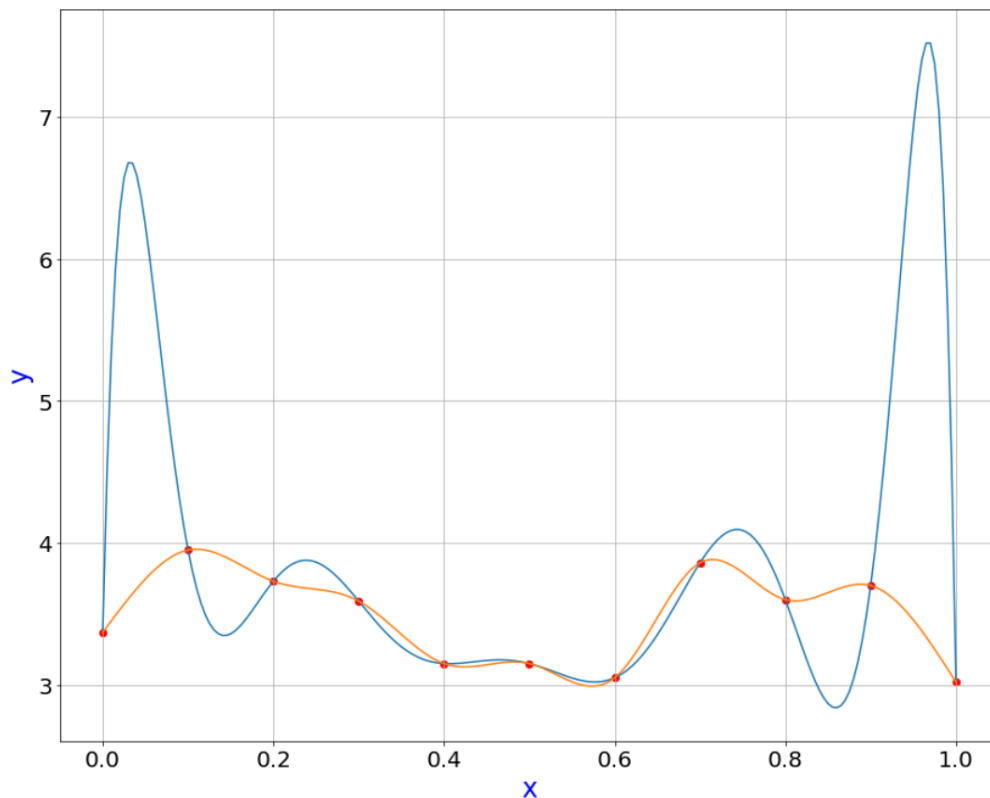


Рисунок 3. Полином Лагранжа (синяя кривая), кубический сплайн(рыжая кривая) и узлы

На рисунке отчетливо видно проявление так называемого феномена Рунге – нежелательные осцилляции, возникающие при интерполировании полиномами высоких степеней. Это происходит из-за структуры базисных полиномов Лагранжа: на концах происходит скопление их экстремумов. В данном случае узлы, выбранные нами, расположены равномерно, а, как известно из лекций, для минимизации ошибки интерполяции Лагранжа стоит выбирать чебышевские узлы. Иными словами, при данных узлах из таблицы 1 мы не можем провести оптимальную интерполяцию полиномом Лагранжа, ошибка интерполяции будет неограниченно возрастать при увеличении числа узлов. Также отметим, что для 11 узлов полином Лагранжа будет иметь 10 степень, то есть выше, более чем в 3 раза, степени кубического сплайна.

## 9 Анализ влияния погрешности измерений

### Часть 1. Погрешность измерений. Метод Монте-Карло

Шумом будем называть погрешность, возникающую при измерениях в реальности. Для каждого метода интерполяции рассмотрим два случая. В первом случае зашумленными данными будут являться значения  $x_i$ , во втором – значения  $h_i$ , представленные в таблице 1.

Для каждого случая проведем 1000 экспериментов, в результате получим четыре семейства размером 1000, по два для каждого метода интерполяции. Описанные действия представляют собой метод Монте-Карло, или метод статистического моделирования, заключающийся в многократном повторении моделирования зашумленных данных для получения интервальной оценки интерполянтов, построенных на предположении о том, что в реальности при измерении значений в узлах существует некоторая неопределенность.

Программно это будет реализовано следующим образом. Воспользуемся функцией `np.random.normal(mu, sigma)`, в которую передадим указанные в задании параметры распределения случайной величины. Данная функция возвращает реализацию случайной величины, имеющей нормальное распределение. Сгенерируем два списка: зашумление по  $x_i$  и зашумление по  $h_i$ . Листинг кода представлен ниже, в нем `x_n` и `y_n` – соответствующие списки, хранящие значения из таблицы 1.

Листинг 7. Генерация списков с зашумленными данными

---

```
mu, sigma = 0, 0.01
m, k = len(x_n), len(y_n)

x_generate = [[x_n[i] + np.random.normal(mu, sigma)
               for i in range(m)] for j in range(1000)]
h_generate = [[y_n[i] + np.random.normal(mu, sigma)
               for i in range(k)] for j in range(1000)]
```

---

Затем сгенерируем еще четыре списка: `lx_generate`, `ly_generate`, `qx_generate` и `qy_generate`. В них хранятся значения соответственно семейства полиномов Лагранжа с зашумлением  $x_i$ , полиномов Лагранжа с зашумлением  $h_i$ , кубических сплайнов с зашумлением  $x_i$  и кубических сплайнов с зашумлением  $h_i$ .

Листинг 8. Генерация интерполянтов на основе зашумленных данных

---

```
lx_generate = [[L(x[i], x_generate[j], y_n)
               for i in range(0, n + 1)] for j in range(1000)]
ly_generate = [[L(x[i], x_n, h_generate[j])
               for i in range(0, n + 1)] for j in range(1000)]
qx_generate = [[qubic_spline(x[i], x_generate[j], qubic_spline_coeff(x_generate[j], y_n))
               for i in range(0, n + 1)] for j in range(1000)]
qy_generate = [[qubic_spline(x[i], x_n, qubic_spline_coeff(x_n, h_generate[j]))
               for i in range(0, n + 1)] for j in range(1000)]
```

---

Построим полученные семейства интерполянтов (рис. 4 - 7).

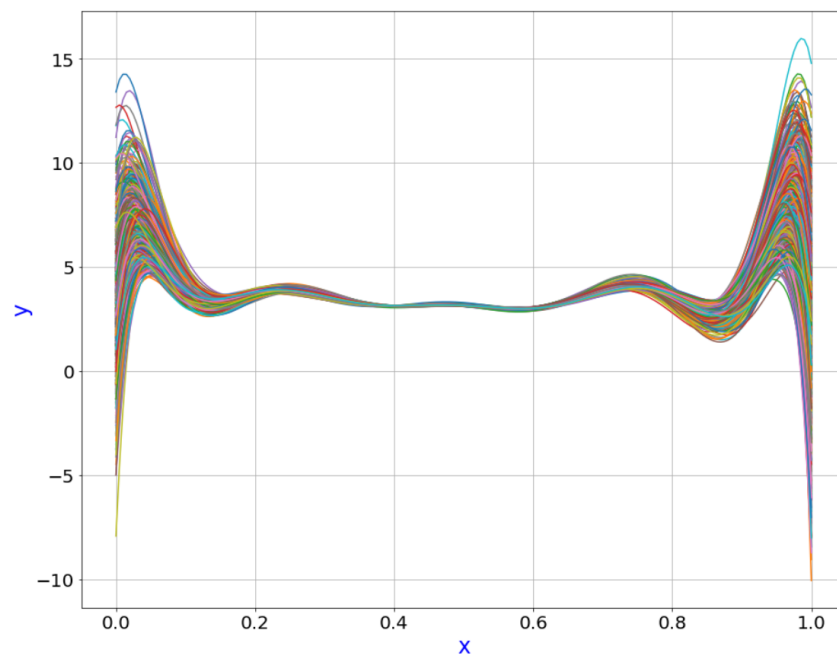


Рисунок 4. Интерполяционные полиномы Лагранжа: семейство размером 1000, построенное на основе зашумленных данных (зашумлены значения  $x_i$ )

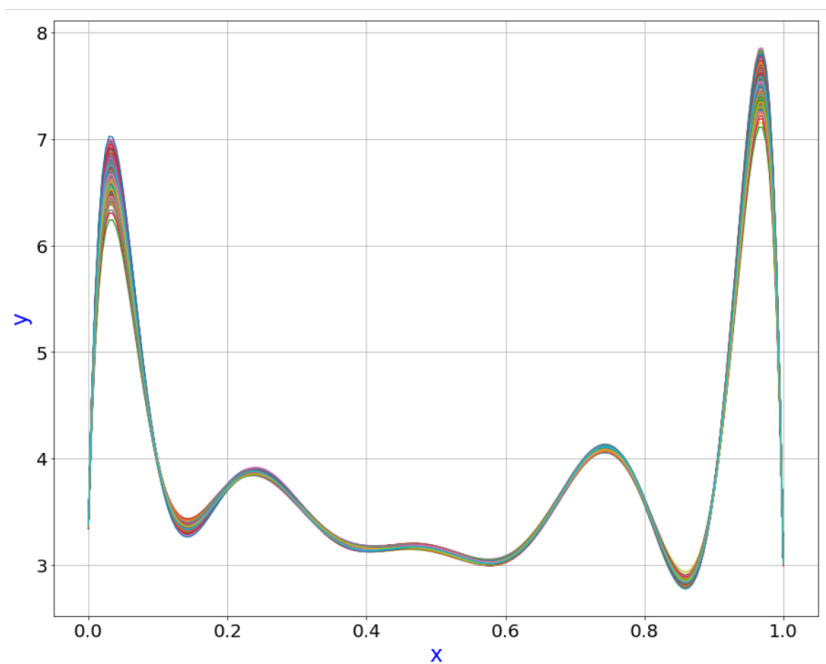


Рисунок 5. Интерполяционные полиномы Лагранжа: семейство размером 1000, построенное на основе зашумленных данных (зашумлены значения  $h_i$ )



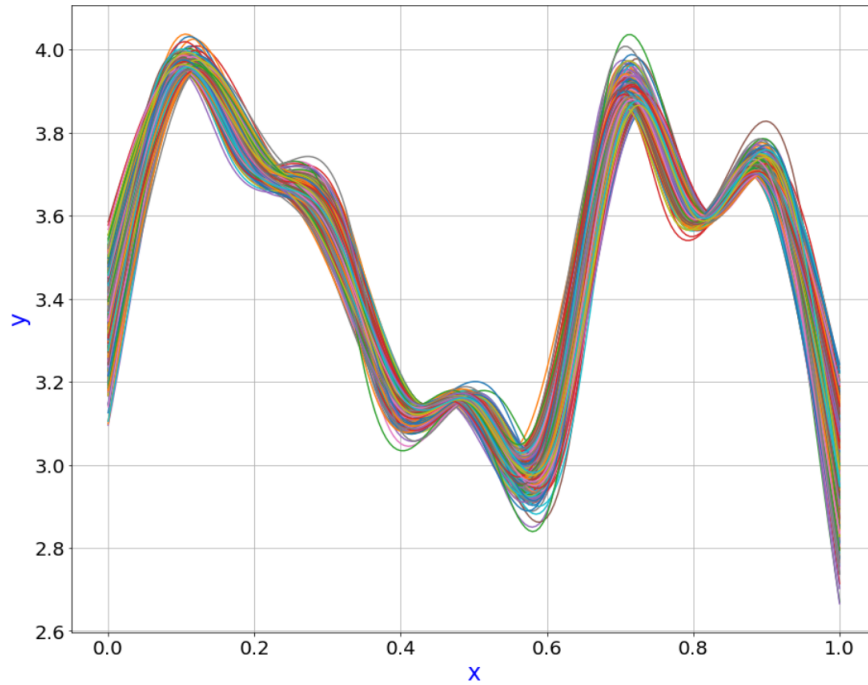


Рисунок 6. Кубические сплайны: семейство размером 1000, построенное на основе зашумленных данных (зашумлены значения  $x_i$ )

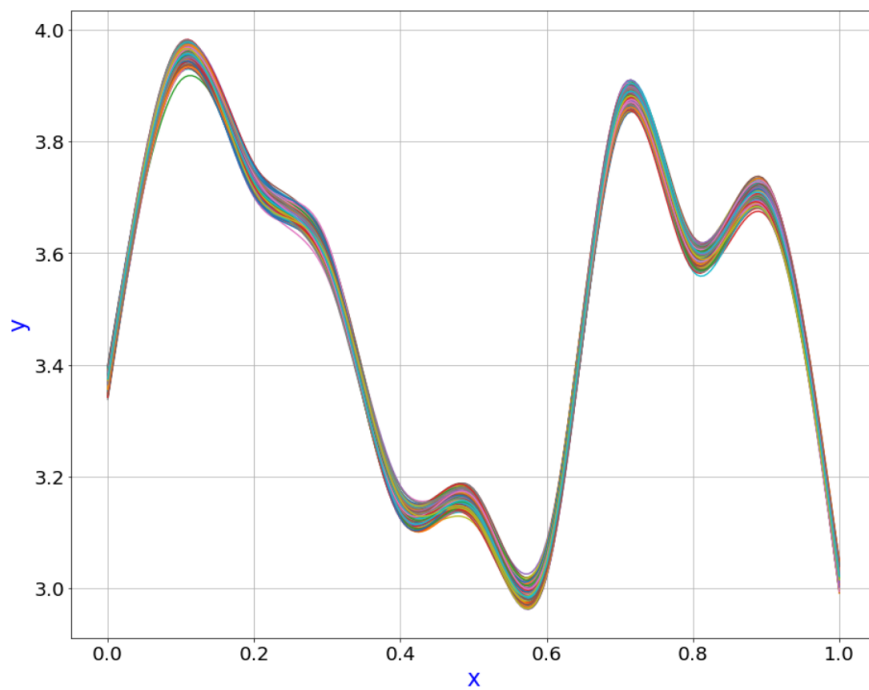


Рисунок 7. Кубические сплайны: семейство размером 1000, построенное на основе зашумленных данных (зашумлены значения  $h_i$ )

## Часть 2. Построение доверительной полосы и усредненного интерполянта

Перейдем к построению доверительной полосы. Для начала определим, для чего вообще используется доверительная полоса, а затем рассмотрим один из способов ее построения.

Доверительные полосы широко используются в статистическом анализе для демонстрации неопределенности в оценке какой-либо функции, построенной на основе зашумленных данных. Как описано ранее, для проведения подобного анализа было смоделировано 1000 векторов, включающих в себя определенную погрешность измерения.

Отличие доверительной полосы от доверительного интервала состоит в том, что доверительные интервалы относятся к какой-то конкретной точке, в то время как доверительные полосы являются обобщением доверительного интервала и относятся уже к совокупности точек. Иными словами доверительная полоса – ряд доверительных интервалов для какой-то последовательности выборок случайных величин из определенного распределения.

В нашем случае доверительной полосой будет область, ограниченная функциями  $h_l$  и  $h_u$  снизу и сверху соответственно.

Для того, чтобы проводить дальнейшие рассуждения, напомним некоторые понятия из статистики и теории вероятностей.

Во всех определениях будем считать, что дано вероятностное пространство  $(\Omega, \mathcal{F}, \mathbb{P})$ , и на нем определена случайная величина  $X$  с распределением  $\mathbb{P}^X$  [3].

**Опр.** Функция  $F_X(x)$  называется функцией распределения случайной величины  $X$  и задается следующей формулой

$$F_X(x) = \mathbb{P}(X \leq x) = \mathbb{P}^X((-\infty, x]), \quad F_X(x): \mathbb{R} \rightarrow [0, 1].$$

**Опр.** Выборка  $\mathbf{X} = \{X_1, \dots, X_n\}$  объема  $n$  из закона распределения  $\mathbb{P}^X$  – набор из  $n$  независимых случайных величин, имеющих распределение  $\mathbb{P}^X$ .

**Опр.** Статистика – функция от выборки  $\mathbf{X}$ .

**Опр.** Набор  $X_{(1)} \leq X_{(2)} \leq \dots \leq X_{(n)}$  упорядоченных по возрастанию компонент выборки  $(X_1, \dots, X_n)$  называется вариационным рядом, а сами случайные величины  $X_{(k)}$  –  $k$ -ми порядковыми статистиками.

$$X_{(1)} = \min\{X_1, \dots, X_n\};$$

$$X_{(2)} = \max\{\min\{X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_n\}\};$$

...

$$X_{(n)} = \max\{X_1, \dots, X_n\}.$$

**Опр.** Две статистики  $\hat{\theta}_1$  и  $\hat{\theta}_2$  определяют границы доверительного интервала параметра  $\theta$  с коэффициентом доверия  $1-p$ ,  $p \in (0, 1)$ , если  $\forall \theta \in \Theta$  для выборки  $\mathbf{X} = \{X_1, \dots, X_n\}$  из закона распределения  $\mathbb{P}^X$  справедливо равенство

$$\mathbb{P}(\hat{\theta}_1(\mathbf{X}) \leq \theta \leq \hat{\theta}_2(\mathbf{X})) = 1 - p.$$

**Опр.** Пусть фиксировано  $\alpha \in (0, 1)$ , тогда  $\alpha$ -квантилем распределения  $\mathbb{P}^X$  называется число  $x_\alpha \in \mathbb{R}$ , такое что

$$\mathbb{P}(X \leq x_\alpha) \geq \alpha.$$

**Опр.** Пусть дана выборка  $(X_1, \dots, X_n)$  объема  $n$ , порожденная случайной величиной  $X$  с функцией распределения  $F(x)$ , тогда случайная величина  $F_n(x)$  называется эмпирической функцией распределения и задается следующим образом

$$F_n(x) = \sum_{i=1}^n \mathbf{1}_{\{X_i \leq x\}}, \quad \mathbf{1}_{\{X_i \leq x\}} = \begin{cases} 1, & X_i \leq x \\ 0, & X_i > x \end{cases}.$$

В нашем случае каждый интерполянт был построен по 201 точке. Соответственно, имеем 201 выборку  $X^1, \dots, X^{201}$  размера 1000,  $X^i = (X_1^i, \dots, X_{1000}^i)$ . Согласно условию задачи необходимо найти такой интервал, что вероятность попадания в него значения интерполянта в точке  $x$  равна 0.9. Иначе говоря, для каждой выборки ставится следующая задача:

$$\mathbb{P}(\tilde{h}_l^i(x) \leq X^i \leq \tilde{h}_u^i(x)) = 0.9.$$

Убедимся, что вся 201 выборка является выборкой из непрерывного распределения. Для обоснования этого утверждения будем рассматривать полином Лагранжа и кубический сплайн для какого-то фиксированного  $x \in [0, 1]$ . Отсюда имеем для интерполяции полиномом Лагранжа  $L_{10}(\xi_1, \dots, \xi_{11})$  – функцию от вектора непрерывных случайных величин и для кубического сплайна  $S_i(\xi_i)$ ,  $i \in [1, 10]$  – функцию от непрерывной случайной величины (для каждого промежутка сплайна своя случайная величина), при зашумлении  $h_i$ , и функцию от вектора непрерывных случайных величин  $S(\xi_1, \dots, \xi_n)$ , при зашумлении  $x_i$ , где  $\xi_i \sim N(0, \frac{1}{100})$ . Далее как полином Лагранжа, так и кубический сплайн (учитывая экстраполяцию 8) являются непрерывными функциями, значит, по *теореме о непрерывности сложной функции* интерполянты Лагранжа и кубические сплайны, смоделированные при помощи зашумления исходных данных непрерывной случайной величиной, также будут непрерывными случайными величинами (при фиксированном  $x$ ). Тогда вся 201 выборка для обоих интерполяционных методов будет выборкой из абсолютно непрерывного распределения.

Исходя из рассуждений выше для абсолютно непрерывного распределения можем пользоваться тем, что  $F_X(x_\alpha) = \alpha$ ,  $x_\alpha$  –  $\alpha$ -квантиль (по определениям квантиля и функции распределения).

Теперь произведем некоторые алгебраические действия:

$$\alpha = \frac{1 + \alpha}{2} - \frac{1 - \alpha}{2} = F\left(x_{\frac{1+\alpha}{2}}\right) - F\left(x_{\frac{1-\alpha}{2}}\right) = \mathbb{P}\left(x_{\frac{1+\alpha}{2}} \leq X^i \leq x_{\frac{1+\alpha}{2}}\right).$$

Для каждой выборки имеем свою эмпирическую функцию распределения. Оценим  $\alpha$ -квантиль методом подстановки, полагая, что если  $\phi(F_X) = x_\alpha$ , то  $\phi(F_n) = \hat{x}_\alpha$ . Из этих соображений несложно построить оценку квантиля следующим образом.

$$k = \lfloor \alpha \cdot (n - 1) \rfloor, \quad x_\alpha = \begin{cases} X_{(k+1)}^i, & k + 1 < \alpha n \\ \frac{X_{(k)}^i + X_{(k+1)}^i}{2}, & k + 1 = \alpha n \\ X_{(k)}^i, & k + 1 > \alpha n \end{cases}$$

Нам необходимы квантили  $x_{\frac{1-\alpha}{2}}$  и  $x_{\frac{1+\alpha}{2}}$ . Рассчитаем, каким  $k$ -порядковым статистикам они равны.

$$k_1 = \left\lfloor \frac{1-\alpha}{2} \cdot (n-1) \right\rfloor = \left\lfloor \frac{1-0.9}{2} \cdot (1000-1) \right\rfloor = \lfloor 0.05 \cdot (999) \rfloor = \lfloor 49.95 \rfloor = 49,$$

$$k_1 + 1 = 50 = \frac{1-0.9}{2} \cdot 1000 \Rightarrow x_{\frac{1-\alpha}{2}} = \frac{X_{(49)}^i + X_{(50)}^i}{2},$$

$$k_2 = \left\lfloor \frac{1+\alpha}{2} \cdot (n-1) \right\rfloor = \left\lfloor \frac{1+0.9}{2} \cdot (1000-1) \right\rfloor = \lfloor 0.95 \cdot (999) \rfloor = \lfloor 949.05 \rfloor = 949,$$

$$k_2 + 1 = 950 = \frac{1+0.9}{2} \cdot 1000 \Rightarrow x_{\frac{1+\alpha}{2}} = \frac{X_{(949)}^i + X_{(950)}^i}{2}.$$

Программно поиск соответствующих порядковых статистик реализован с помощью сортировки списков, в каждом из которых хранится 1000 значений для всей 201 точки, по которым происходили построения графиков. Затем для каждого списка длины 1000 берем значения с индексами 48 и 49, суммируем и делим на 2. Получаем квантиль, по которому будет построена кривая  $h_l$ . Затем берем значения с индексами 948 и 949 и также делим их сумму на 2. Получаем квантиль, по которому будет построена кривая  $h_u$ . Усредненный интерполянт будем искать, находя выборочное среднее.

Листинг 9. Построение доверительной полосы и поиск усредненного интерполянта

---

```
def conf_band(y, n):
    y_ = y.transpose()
    for i in range(n + 1):
        y_[i].sort()

    h_l = [(y_[i][48] + y_[i][49]) / 2 for i in range(n + 1)]
    h_m = [sum(y_[i]) / 1000 for i in range(n + 1)]
    h_u = [(y_[i][948] + y_[i][949]) / 2 for i in range(n + 1)]

    return [h_l, h_m, h_u]
```

---

**Функция** `conf_band(y, n)` **принимает** на вход матрицу вида  $\begin{pmatrix} x_1^1 & \dots & x_1^m \\ \vdots & \dots & \vdots \\ x_n^1 & \dots & x_n^m \end{pmatrix}$ , где  $m$

- число выборок, а  $n$  - количество элементов выборки. В нашем случае  $m = 201, n = 1000$ .

Для построения доверительной полосы нужно брать столбцы матрицы и производить построение вариационного ряда, программно – сортировку списка, в котором содержатся значения  $i$ -го столбца,  $i \in [1, 201]$ . Как видно, нам было бы удобнее, чтобы столбцы матрицы были ее строками, чтобы осуществлять перебор по выборкам в цикле. Поэтому мы транспонируем переданную в функцию матрицу, чтобы она приняла

вид  $\begin{pmatrix} x_1^1 & \dots & x_n^1 \\ \vdots & \dots & \vdots \\ x_1^m & \dots & x_n^m \end{pmatrix}$  и затем уже осуществляем проход по ее строкам с сортировкой

каждого списка из строки. Функция **возвращает** матрицу  $3 \times (n - 1)$  в виде списка. В результате получаем графики, приведенные ниже (рис. 8, 9, 10, 11).

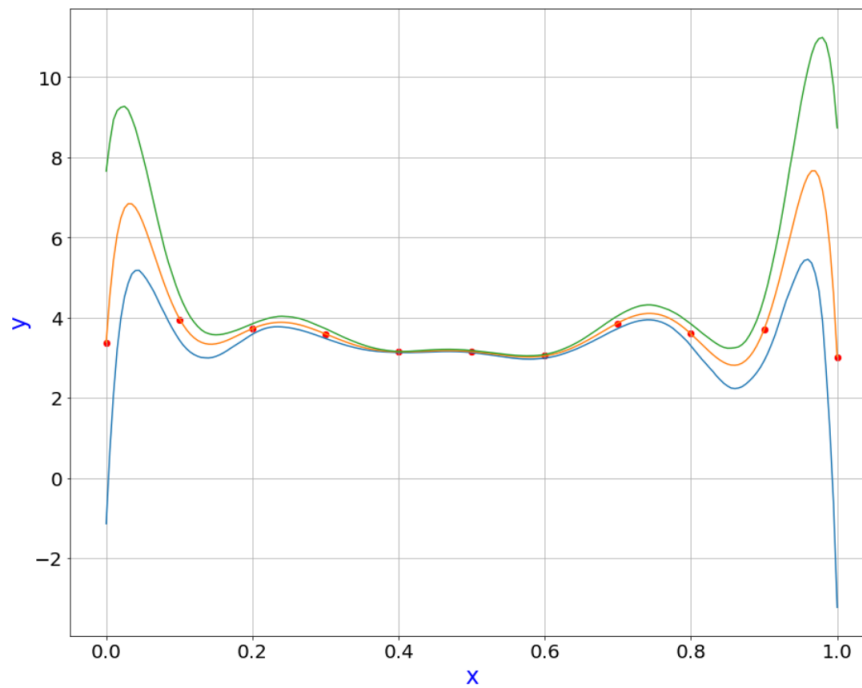


Рисунок 8. Доверительная полоса для выборки полиномов Лагранжа, построенной на основе зашумленных данных (зашумлены значения  $x_i$ ). Зеленая кривая –  $h_u$ , синяя кривая –  $h_l$ , рыжая кривая - усредненный интерполянт

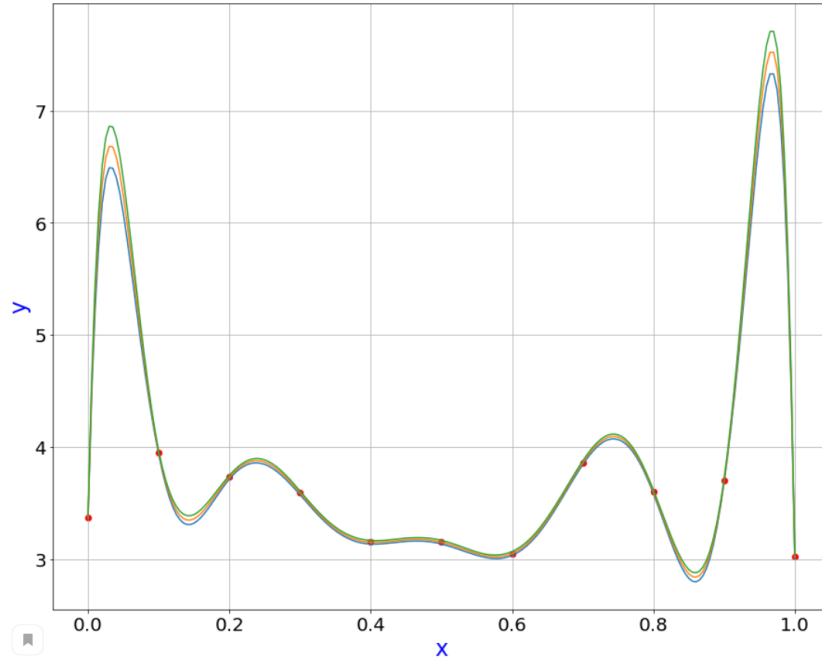


Рисунок 9. Доверительная полоса для выборки полиномов Лагранжа, построенной на основе зашумленных данных (зашумлены значения  $h_i$ ). Зеленая кривая –  $h_u$ , синяя кривая –  $h_l$ , рыжая кривая - усредненный интерполянт

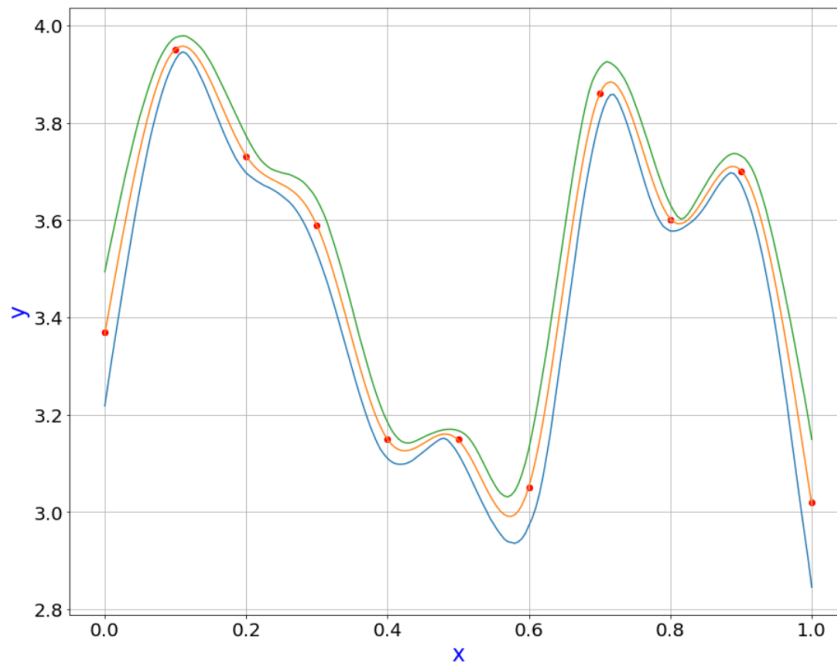


Рисунок 10. Доверительная полоса для выборки кубических сплайнов, построенной на основе зашумленных данных (зашумлены значения  $x$ ). Зеленая кривая –  $h_u$ , синяя кривая –  $h_l$ , рыжая кривая - усредненный интерполянт

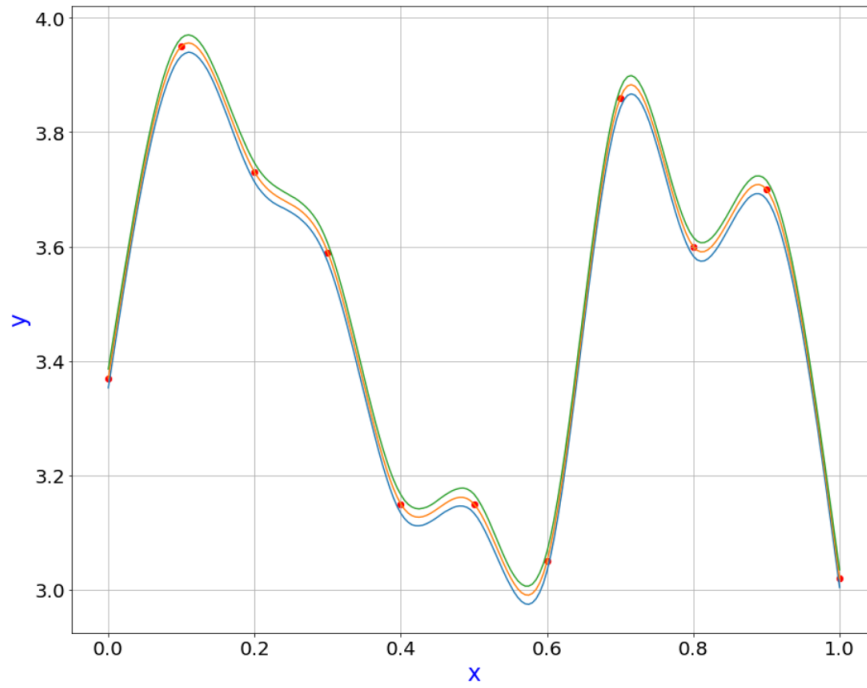


Рисунок 11. Доверительная полоса для выборки кубических сплайнов, построенной на основе зашумленных данных (зашумлены значения  $h$ ). Зеленая кривая –  $h_u$ , синяя кривая –  $h_l$ , рыжая кривая - усредненный интерполянт

### Часть 3. Анализ влияния погрешностей на результаты интерполяции

Для проведения анализа влияния погрешностей на интерполяцию воспользуемся модулем pandas. С помощью него создадим таблицы, в которых будут храниться значения полученных кривых для доверительной полосы, значения усредненных интерполянтов и значения интерполянтов, вычисляемых при условии, что данные не были зашумлены.

Проанализируем, какие участки каждого интерполянта являются наиболее чувствительны к погрешностям.

Начнем с полинома Лагранжа.

Таблица 2: Доверительные полосы для полинома Лагранжа, первая 21 точка

	Лагранж, x, hl	Лагранж, h, hl	Лагранж, x, hu	Лагранж, h, hu	x
0	-1.141752	3.353102	7.659965	3.386143	0.000
1	0.678443	4.427928	8.385957	4.559554	0.005
2	2.096533	5.221001	8.941752	5.448480	0.010
3	3.209327	5.787035	9.169519	6.084063	0.015
4	3.988192	6.168514	9.247275	6.507414	0.020
5	4.521292	6.393403	9.274889	6.755571	0.025
6	4.858994	6.491215	9.186320	6.861928	0.030
7	5.089203	6.487279	8.974797	6.855630	0.035
8	5.176515	6.406320	8.722524	6.762216	0.040
9	5.180573	6.266809	8.390826	6.600756	0.045
10	5.084935	6.080485	8.038450	6.389514	0.050
11	4.963447	5.867272	7.675037	6.145058	0.055
12	4.812942	5.636265	7.271136	5.879650	0.060
13	4.655902	5.395433	6.868160	5.604349	0.065
14	4.467876	5.152463	6.465268	5.328162	0.070
15	4.273559	4.916230	6.087566	5.058768	0.075
16	4.098443	4.688970	5.707882	4.799629	0.080
17	3.912608	4.475944	5.382180	4.558528	0.085
18	3.744756	4.278302	5.105244	4.336945	0.090
19	3.590341	4.097159	4.825049	4.137355	0.095
20	3.445077	3.933560	4.578829	3.964819	0.100

В таблице 2 представлена следующая информация: в столбце “Лагранж, x, hl” – нижняя граница доверительной полосы при зашумленных  $x_i$ , “Лагранж, h, hl” – нижняя граница доверительной полосы при зашумленных  $h_i$ , “Лагранж, x, hu” и “Лагранж, h, hu” – верхние границы, “x” – данные  $x$ .

Оценим верхнюю границу погрешности, привносимой шумом. Возьмем максимальную по модулю  $Z$  из всех, что были прибавлены к  $x_i$ :

$$\tilde{L}_{10} = \sum_{i=1}^{10} f(x_i) \prod_{i=1, i \neq j}^n \frac{x - x_j + |Z|}{x_i - x_j}, \quad |L_{10} - \tilde{L}_{10}| < C|Z|^{10}.$$

То есть при зашумлении  $x_i$  максимальная абсолютная погрешность для полинома Лагранжа имеет верхнюю оценку, пропорциональную  $|Z|^{10}$  ( $C$  – какая-то константа).



Теперь оценим абсолютную погрешность при зашумлении  $h_i = f(x_i)$ :

$$\tilde{L}_{10} = \sum_{i=1}^{10} (f(x_i) + |Z|) \prod_{i=1, i \neq j}^n \frac{x - x_j}{x_i - x_j}, \quad |L_{10} - \tilde{L}_{10}| < C|Z|.$$

Здесь уже верхняя оценка абсолютной погрешности гораздо меньше, так как мы просто изменяем линейный параметр. Посмотрим, что происходит с кубическим сплайном.

Таблица 3: Доверительные полосы для кубического сплайна, первая 21 точка

	Сплайн, x, hl	Сплайн, h, hl	Сплайн, x, hu	Сплайн, h, hu	x
0	3.218215	3.353102	3.493812	3.386143	0.000
1	3.267740	3.394641	3.529519	3.425633	0.005
2	3.315591	3.435736	3.563875	3.465046	0.010
3	3.361758	3.476101	3.598130	3.504328	0.015
4	3.407116	3.515837	3.631560	3.543010	0.020
5	3.453173	3.555344	3.664231	3.580878	0.025
6	3.498116	3.593418	3.698284	3.618745	0.030
7	3.541904	3.630317	3.729157	3.655980	0.035
8	3.585223	3.666004	3.759170	3.691314	0.040
9	3.626589	3.699729	3.788585	3.725871	0.045
10	3.666091	3.732295	3.815710	3.759023	0.050
11	3.703741	3.762694	3.839833	3.790188	0.055
12	3.739739	3.791541	3.863554	3.819575	0.060
13	3.773735	3.818187	3.885303	3.846574	0.065
14	3.803352	3.842684	3.905396	3.871682	0.070
15	3.831294	3.864444	3.923854	3.894508	0.075
16	3.856809	3.883774	3.938686	3.914502	0.080
17	3.879627	3.900495	3.951551	3.931620	0.085
18	3.899249	3.914978	3.962089	3.945820	0.090
19	3.915979	3.925845	3.970135	3.956996	0.095
20	3.929992	3.933560	3.975065	3.964819	0.100

В таблице 3 представлена следующая информация: в столбце “Сплайн, x, hl” – нижняя граница доверительной полосы при зашумленных  $x_i$ , “Сплайн, h, hl” – нижняя граница доверительной полосы при зашумленных  $h_i$ , “Сплайн, x, hu” и “Сплайн, h, hu” – верхние границы, “x” – данные  $x$ .

Будем полагать  $|Z|$  такой же как и в случае выше с полиномом Лагранжа:

$$x_i - |Z|, \quad x_{i+1} + |Z|, \quad h_i = x_{i+1} - x_i + 2|Z|,$$

$$(h_{i-1} + 2|Z|)c_{i-1} + 2(h_i + h_{i-1} + 4|Z|)c_i + (h_i + 2|Z|)c_{i+1} = \frac{3}{h_{i-1} + 2|Z|}(a_i - a_{i-1}) - \frac{3}{h_{i-2} + 2|Z|}(a_{i-1} - a_{i-2}),$$

раз для нашего случая узлы расположены равномерно, то  $h_{i-2} = h_{i-1} = h_i \forall i \in [1, n]$  и пусть  $h_i + 2|Z| = \Delta$ ,

$$\Delta c_{i-1} + 4\Delta c_i + \Delta c_{i+1} = \frac{3}{\Delta}(a_i - a_{i-1}) - \frac{3}{\Delta}(a_{i-1} - a_{i-2}),$$

$$\begin{aligned}
c_{i-1} + 4c_i + c_{i+1} &= \frac{3}{\Delta^2}(a_i - a_{i-1}) - \frac{3}{\Delta^2}(a_{i-1} - a_{i-2}), \\
\tilde{c}_i &= -4c_i - c_{i-1} + \frac{3}{\Delta^2}(a_i - a_{i-1}) - \frac{3}{\Delta^2}(a_{i-1} - a_{i-2}), \\
\left| \frac{1}{\Delta^2} - \frac{1}{h_i^2} \right| &= \left| \frac{4|Z|^2 + 4|Z| + h_i^2 - h_i^2}{4|Z|^2 h_i^2 + 4|Z| h_i^2 + h_i^4} \right| = \left| \frac{1}{h_i^2} \cdot \left( 1 - \frac{h_i^2}{4|Z|^2 + 4|Z| + h_i^2} \right) \right| < \frac{C}{|Z|^2} \Rightarrow |c_i - \tilde{c}_i| < \frac{C_1}{|Z|^2}, \\
\tilde{d}_i &= \frac{\tilde{c}_{i+1} - \tilde{c}_i}{3\Delta} \Rightarrow |d_i - \tilde{d}_i| < \frac{C_2}{|Z|^3} \\
\tilde{b}_i &= \frac{1}{\Delta}(a_{i+1} - a_i) - \frac{\Delta}{3}(c_{i+1} + 2c_i) \Rightarrow |b_i - \tilde{b}_i| < \frac{C_3}{|Z|}, \quad C, C_1, C_2, C_3 = \text{const.}
\end{aligned}$$

Мы рассмотрели оценки коэффициентов сплайна, теперь рассмотрим абсолютную погрешность для значения сплайна при зашумлении  $x_i$ :

$$|S_i(x) - \tilde{S}_i(x)| < \left| \frac{C_3}{|Z|}(x - x_i) \right| + |\tilde{b}_i| |Z| + \left| \frac{C_1}{|Z|^2}(x - x_i)^2 \right| (x - x_i)^2 + |\tilde{c}_i| |Z|^2 + \left| \frac{C_2}{|Z|^3}(x - x_i)^3 \right| + |\tilde{d}_i| |Z|^3 < \frac{C_4}{|Z|}.$$

Переобозначим  $h_i$  – зашумляемые значения, через  $f(x_i) = a_i$  и посмотрим, что будет происходить со сплайном в этом случае.

$$h_{i-1}c_{i-1} + 2(h_i + h_{i-1})c_i + h_i c_{i+1} = \frac{3}{h_{i-1}}(a_i - a_{i-1} + 2|Z|) - \frac{3}{h_{i-2}}(a_{i-1} - a_{i-2} + 2|Z|),$$

$$|c_i - \tilde{c}_i| < C_1|Z|,$$

$$|b_i - \tilde{b}_i| < C_2|Z|,$$

$$|d_i - \tilde{d}_i| < C_3|Z|,$$

$$|S_i(x) - \tilde{S}_i(x)| < C_4|Z|, \quad C_1, C_2, C_3, C_4 = \text{const.}$$

Как видим, верхняя оценка абсолютной погрешности для сплайна при зашумлении  $x_i$  пропорциональна  $\frac{1}{|Z|}$ , а при зашумлении  $h_i$  пропорциональна  $|Z|$ . Учитывая, что  $|Z|$  это случайная величина  $\sim N(0, \frac{1}{100})$ , по *закону трех сигм* вероятность того, что случайная величина отклонится от своего математического ожидания более чем на три среднеквадратических отклонения, больше, чем 0.99. В нашем случае  $\mathbb{P}(-\frac{3}{100} \leq Z \leq \frac{3}{100}) > 0.99$ . Поэтому на графике для зашумленных  $x_i$  наблюдаем более широкую доверительную полосу, чем на графике для зашумленных  $h_i - \frac{1}{|Z|}$  в нашем случае получается в среднем больше, чем  $|Z|$ .

Обратимся к графикам (рис. 10, 11): доверительные полосы для кубического сплайна имеют примерно одинаковую ширину на всем промежутке, гораздо более стабильную, чем у полиномов Лагранжа. Это можно объяснить тем, что в случае локальной интерполяции шумовая погрешность влияет на все промежутки независимо и примерно в равной степени. В то время как при глобальной интерполяции (рис. 8, 9) промежутков всего один и ближе к концам наблюдается эффект Рунге, поэтому шум влияет

на глобальный интерполянт не равномерно. То же самое можем увидеть, обратясь к таблицам 2 и 3. При приближении к центральным узлам доверительная полоса для глобальной интерполяции становится все уже, в то время как для локальной остается примерно одинаковой.

Теперь сравним усредненные интерполянты и "истинные". Посмотрим первые 15 точек.

Таблица 4: Усредненные интерполянты, первые 15 точек

	Лагранж, x	Лагранж, h	Лагранж	Сплайн, x	Сплайн, h	Сплайн
0	3.393361	3.370210	3.370000	3.366870	3.370210	3.370000
1	4.566258	4.496086	4.494958	3.407440	3.410642	3.410408
2	5.442847	5.337397	5.335577	3.447899	3.450902	3.450645
3	6.069885	5.939111	5.936792	3.488065	3.490819	3.490539
4	6.488754	6.341021	6.338364	3.527758	3.530220	3.529918
5	6.735925	6.578189	6.575329	3.566796	3.568935	3.568611
6	6.843399	6.681375	6.678423	3.604998	3.606791	3.606446
7	6.839115	6.677428	6.674476	3.642184	3.643617	3.643251
8	6.747330	6.589653	6.586772	3.678173	3.679240	3.678856
9	6.588973	6.438152	6.435398	3.712783	3.713490	3.713088
10	6.381973	6.240139	6.237555	3.745834	3.746195	3.745777
11	6.141564	6.010237	6.007853	3.777144	3.777182	3.776750
12	5.880564	5.760745	5.758581	3.806534	3.806280	3.805835
13	5.609640	5.501890	5.499958	3.833821	3.833318	3.832863
14	5.337543	5.242061	5.240364	3.858825	3.858124	3.857660

В таблице 2 представлена следующая информация: в столбце “Лагранж, x” находятся значения усредненного интерполянта при зашумленных данных  $x_i$ , в столбце “Лагранж, h” – значения усредненного интерполянта при зашумленных  $h_i$ , в столбце “Лагранж” – значения синтерполянта без зашумления. Аналогично для сплайна.

Так как моделируемая погрешность является случайной величиной с нулевым математическим ожиданием, то в среднем абсолютное отклонение истинного значения интерполянта от усредненного должно быть почти наверное равно нулю (при размере выборки, стремящимся к бесконечности). Как видно из таблицы 2 для полинома Лагранжа при зашумленных данных  $x_i$  отклонение среднего выборочного от матожидания больше, чем при зашумленных данных  $h_i$ .

Посчитаем среднее значение абсолютного отклонения среднего выборочного от матожидания для всей совокупности из 201 выборки для каждого из четырех случаев.

$$\bar{\Delta} = \frac{1}{m} \sum_{i=1}^m |\bar{f}(x_i) - f(x_i)|.$$

Для нашего случая  $m = 201$ . В результате для четырех столбцов из таблицы получаем

Таблица 5: Значения средних абсолютных отклонений среднего выборочного от математического ожидания

функция	$L_x$	$L_h$	$S_x$	$S_h$
погрешность	0.0248402	0.0013058	0.00136555	0.0001726

В таблице 5 нижние индексы у  $L$  и  $S$  означают зашумляемые данные. Можем видеть, что наибольшая погрешность наблюдается при зашумлении  $x_i$  для полинома Лагранжа, а наименьшая при зашумлении  $h_i$  для кубического сплайна. Такие отклонения можно объяснить все теми же рассуждениями, приведенными выше, в которых рассматривались верхние оценки абсолютной погрешности. Однако, стоит отметить, что среднее абсолютное отклонение выборочного среднего от математического ожидания все же невелико. Такой эффект наблюдается за счет того, что случайная величина имеет сравнительно малую меру разброса, как уже отмечалось выше, поэтому все усредненные интерполлянты сосредоточены в достаточно малой окрестности своих истинных значений.

## 10 Заключение

В результате проведенного анализа и проделанной работы можем сделать следующие выводы.

1. Неопределенность исходных данных влияет на глобальную и локальную интерполяцию по-разному. В первом случае в результате эффекта Рунге происходят сильные отклонения от среднего на краях отрезка интерполяции, но ближе к середине доверительная полоса сужается. Во втором случае неопределенность относительно равномерно влияет на интерполяцию и доверительная полоса сохраняет примерно одинаковую ширину на всем отрезке интерполяции.
2. Наиболее сильные отклонения от среднего при глобальной интерполяции наблюдаются при зашумлении данных  $x_i$ , это можно объяснить высокой степенью (пропорциональна степени интерполяционного полинома) верхней оценки абсолютной погрешности интерполяции. В то время как отклонения при зашумлении  $h_i$  являются небольшими, что опять же объясняется малой степенью верхней оценки абсолютной погрешности.
3. Для локальной интерполяции ситуация складывается аналогично, однако там уже подобное явление связано с тем, какие значения может принимать моделируемая случайная величина, что было показано во 3 части 9 пункта данной работы.
4. Для малого количества узлов лучше использовать интерполяцию полиномом Лагранжа, так как она требует меньше затрат на вычисления. Однако локальная интерполяция кубическими сплайнами более устойчива к погрешностям, при большом количестве узлов (а особенно равномерно расположенных) лучше использовать ее.

## Список использованных источников

1. Першин А.Ю. Лекции по курсу «Вычислительная математика». Москва, 2018-2021. С. 140.
2. Richard L. Burden, J. Douglas Faires, Annette M. Burden. Numerical Analysis. Cengage Learning, 2015. P. 912.
3. М. Б. Лагутин. Наглядная математическая статистика. Москва. БИНОМ. Лаборатория знаний, 2009. С. 472.

## Выходные данные

Горелкина Е.Е.. Отчет о выполнении лабораторной работы по дисциплине «Вычислительная математика». [Электронный ресурс] — Москва: 2021. — 30 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, Горелкина Е.Е.

2021, осенний семестр