Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего 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
Группа	РК6-52Б
Тип задания	Лабораторная работа №1
Тема лабораторной работы	Интерполяция в условиях измерений с неопределенностью

Студент _____ **Журавлев Н.В.**
подпись, дата *фамилия, и.о.*

Преподаватель _____ **Першин А. Ю**
подпись, дата *фамилия, и.о.*

Москва, 2021 г.

Оглавление

Задание на лабораторную работу

Интерполяция, вероятно, является самым простым способом определения недостающих значений некоторой функции при условии, что известны соседние значения. Однако, за кадром зачастую остается вопрос о том, насколько точно мы знаем исходные данные для проведения интерполяции или любой другой аппроксимации. К примеру, исходные данные могут быть получены путем снятия показаний с датчиков, которые всегда обладают определенной погрешностью. В этом случае всегда возникает желание оценить влияние подобных погрешностей и неопределенностей на аппроксимацию. В этом задании на простейшем примере мы познакомимся с интерполяцией в целом (базовая часть) и проанализируем, как неопределенности влияют на ее предсказания (продвинутая часть).

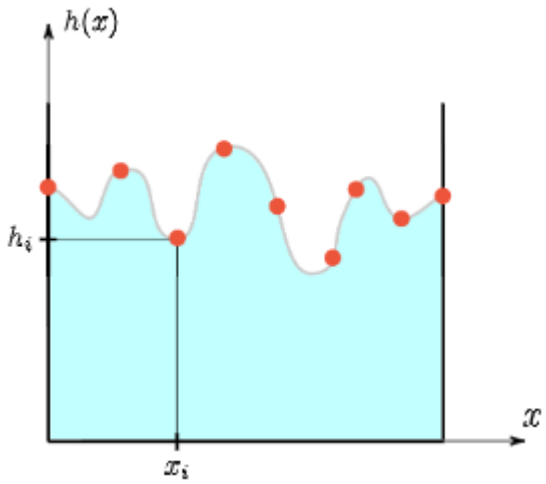

Рисунок 1

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

Базовая часть

1. Разработать функцию $qubic\_spline\_coeff(x\_nodes, y\_nodes)$, которая посредством решения матричного уравнения вычисляет коэффициенты естественного кубического сплайна. Для простоты,

решение матричного уравнения можно производить с помощью вычисления обратной матрицы с использованием функции $numpy.linalg.inv()$.

2. Написать функции $qubic\_spline(x, qs\_coeff)$ и $d\_qubic\_spline(x, qs\_coeff)$, 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 (см. рисунок 1) с помощью кубического сплайна и продемонстрировать ее на графике вместе с исходными узлами.

Таблица 1: Значения уровня поверхности вязкой жидкости (рис. 1)

i	1	2	3	4	5	6	7	8	9	10	11
x_i	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
h_i	3.37	3.95	3.73	3.59	3.15	3.05	3.05	3.86	3.60	3.70	3.03

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, вточке 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_i всегда возникает погрешность, которая моделируется случайной величиной с нормальным распределением с нулевым математическим ожиданием и

стандартным отклонением10^{-2}. Требуется провести следующий анализ, позволяющий выявить влияние этой погрешности на интерполяцию:

a. Сгенерировать 1000 векторов значений $[\tilde{x}_1, \ldots, \tilde{x}_{11}]^T$, 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}.

b. 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.

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

d. Отобразить на едином графике функции $\tilde{h}_l(x), \tilde{h}_u(x)$, усредненный интерполянт и узлы из таблицы 1.

e. 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 предыдущие пункта для случая интерполяции кубическим сплайном. Какие выводы вы можете сделать, сравнив результаты анализа для интерполяции Лагранжа и интерполяции кубическим сплайном?

Цель выполнения лабораторной работы

Изучить общие принципы интерполирования на примере построения кубических сплайнов и полиномов.

Выполненные задачи

Базовая часть

1. Вычисление коэффициентов кубического сплайна.
2. Вычисление кубического сплайна и его производной.
3. Аппроксимация узлами в таблице 1 кубических сплайнов.

Продвинутая часть:

1. Возвращение значения i−го базисного полинома Лагранжа.
2. Возвращение значения интерполяционного полинома Лагранжа, заданного на узлах.
3. Анализ, позволяющий выявить влияние погрешности при измерении координаты xiна интерполяцию.
    a. Генерация векторов.
    b. Построение интерполянтов Лагранжа для каждого из полученных векторов.
    c. Построение таких функций $\tilde{h}l(x)$, $\tilde{h}u(x)$, что вероятность того, что значение интерполянта в точке будет лежать в интервале [$\tilde{h}l(x)$, $\tilde{h}u(x)$] равна 0.9.
    d. Отображение на едином графике функций $\tilde{h}l(x)$, $\tilde{h}u(x)$, усреднённого интерполянта и узлов из таблицы 1.
    e. Выявление чувствительных участков интерполянта.
4. Повторение анализа, описанного в предыдущем пункте, предполагая, что координаты xi известны точно, в то время как

измерения уровня поверхности hi имеют ту же погрешность, что и в предыдущем пункте.

5. Повторение двух предыдущих пункта для случая интерполяции кубическим сплайном.

Вычисление коэффициенты естественного кубического сплайна

Листинг 1:

```
def qubic_spline_coef(x_nodes, y_nodes):
    n = len(x_nodes)
    h = [x_nodes[i + 1] - x_nodes[i] for i in range(n - 1)]
    matrix_a = numpy.diag(numpy.r_[[1], [2 * (h[i + 1] + h[i]) for i in
range(len(h) - 1)], [1]])
    matrix_a = matrix_a + numpy.diag(numpy.r_[[0], [h[i] for i in
range(1, n - 1)]], 1)
    matrix_a = matrix_a + numpy.diag(numpy.r_[[h[i] for i in range(n -
2)], [0]], -1)

    a = numpy.array(y_nodes)

    matrix_b = numpy.r_[[0], [3 * (a[i + 2] - a[i + 1]) / h[i + 1] - 3
* (a[i + 1] - a[i]) / h[i] for i in range(n - 2)], [0]]

    c = numpy.linalg.solve(matrix_a, matrix_b)
    d = numpy.array([(c[i + 1] - c[i]) / (3 * h[i]) for i in range(n -
1)])
    b = numpy.array([(a[i + 1] - a[i]) / h[i] - h[i] * (c[i + 1] + 2 *
c[i]) / 3 for i in range(n - 1)])

    result = numpy.zeros((5, n - 1))
    for i in range(n - 1):
        result[0][i] = a[i]
        result[1][i] = b[i]
        result[2][i] = c[i]
        result[3][i] = d[i]
        result[4][i] = x_nodes[i]
    return result
```

Вывод:

Запрограммируем формулы для вычисления коэффициентов a_i, b_i, c_i, d_i:

$$a_i = f(x_i) \quad (1.1)$$

$$\begin{bmatrix} 1 & 0 & \cdots & \cdots & \cdots & 0 \\ h_1 & 2(h_2+h_1) & h_2 & 0 & \cdots & 0 \\ 0 & h_2 & 2(h_3+h_2) & 2 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & h_{n-2} & 2(h_{n-2}+h_{n-1}) & h_{n-1} \\ 0 & \cdots & \cdots & \cdots & \cdots & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_{n-1} \\ c_{n-2} \end{bmatrix} =$$

$$\begin{bmatrix} 0 \\ \frac{3}{h_2}(a_3-a_2) - \frac{3}{h_1}(a_2-a_1) \\ \frac{3}{h_3}(a_4-a_3) - \frac{3}{h_2}(a_3-a_2) \\ \vdots \\ \frac{3}{h_{n-1}}(a_n-a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1}-a_{n-2}) \\ 0 \end{bmatrix} \quad (1.2)$$

$$d_i = \frac{c_{i+1}-c_i}{3h_i} \quad (1.3)$$

$$b_i = \frac{1}{h_i}(a_{i+1}-a_i) - \frac{h_i}{3}(c_{i+1}-2c_i) \quad (1.4)$$

При нахождении коэффициента c, т.к. используются естественные граничные условия, то $c_1 = c_n = 0$.

Вычисление значения и производной кубического сплайна

Листинг 2:

```
def qubic_spline(x, qs_coeff):

    a = qs_coeff[0]
    b = qs_coeff[1]
    c = qs_coeff[2]
    d = qs_coeff[3]
    xi = qs_coeff[4]

    if x <= xi[0]:
        i = 0
        return a[i] + b[i] * (x - xi[i]) + c[i] * (x - xi[i]) ** 2 +
d[i] * (x - xi[i]) ** 3

    for i in range(1, len(xi)):
        if xi[i - 1] <= x <= xi[i]:
            j = i - 1
            return a[j] + b[j] * (x - xi[j]) + c[j] * (x - xi[j]) ** 2
+ d[j] * (x - xi[j]) ** 3

    if xi[-1] <= x:
        return a[-1] + b[-1] * (x - xi[-1]) + c[-1] * (x - xi[-1]) **
2 + d[-1] * (x - xi[-1]) ** 3
```

Функция d_qubic_spline:

```
def d_qubic_spline(x, qs_coeff):
    b = qs_coeff[1]
    c = qs_coeff[2]
    d = qs_coeff[3]
    xi = qs_coeff[4]

    if x < - xi[0]:
        i = 0
        return b[i] + 2 * c[i] * (x - xi[i]) + 3 * d[i] * (x - xi[i])
** 2

    for i in range(1, len(xi)):
        if xi[i - 1] <= x <= xi[i]:
            j = i - 1
            return b[j] + 2 * c[j] * (x - xi[j]) + 3 * d[j] * (x -
xi[j]) ** 2

    if xi[-1] <= x:
        i = - 1
        return b[i] + 2 * c[i] * (x - xi[i]) + 3 * d[i] * (x - xi[i])
** 2
```

Вывод:

Значение кубического сплайна считается по формуле:

$$S_i(x) = a_i + b_i(x - x\_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \quad (1.5)$$

Значение производной считается по формуле:

$$S_i(x) = b_i + c_i(x - x_i) + d_i(x - x_i)^2 \quad (1.6)$$

Сами коэффициенты считаем по программе из предыдущего пункта.

Вывод

Код программы:

```
def graph_one():
    x = numpy.linspace(0, 1, 11)
    y = [3.37, 3.95, 3.73, 3.59, 3.15, 3.15, 3.05, 3.86, 3.60, 3.70,
3.02]
    a = qubic_spline_coef(x, y)

    x1 = numpy.linspace(0, 1, 1000)
    y1 = numpy.array([qubic_spline(i, a) for i in x1])
    plt.figure(figsize=(8, 5))
    plt.title('Кубический сплайн, проходящий через узлы из таблицы 1',
fontsize=17)

    plt.plot(x1, y1)
```

```
    plt.plot(x, y, 'ro')

    plt.show()


def graph_two():
    x = numpy.linspace(0, 1, 11)
    y = [3.37, 3.95, 3.73, 3.59, 3.15, 3.15, 3.05, 3.86, 3.60, 3.70,
3.02]
    a = qubic_spline_coef(x, y)

    x1 = numpy.linspace(0, 1, 1000)
    y1 = numpy.array([d_qubic_spline(i, a) for i in x1])

    plt.plot(x1, y1)
    plt.plot(x, y, 'ro')

    plt.show()
```

Интерполяция кубическим сплайдом даёт функцию:

Кубический сплайн, проходящий через узлы из таблицы 1

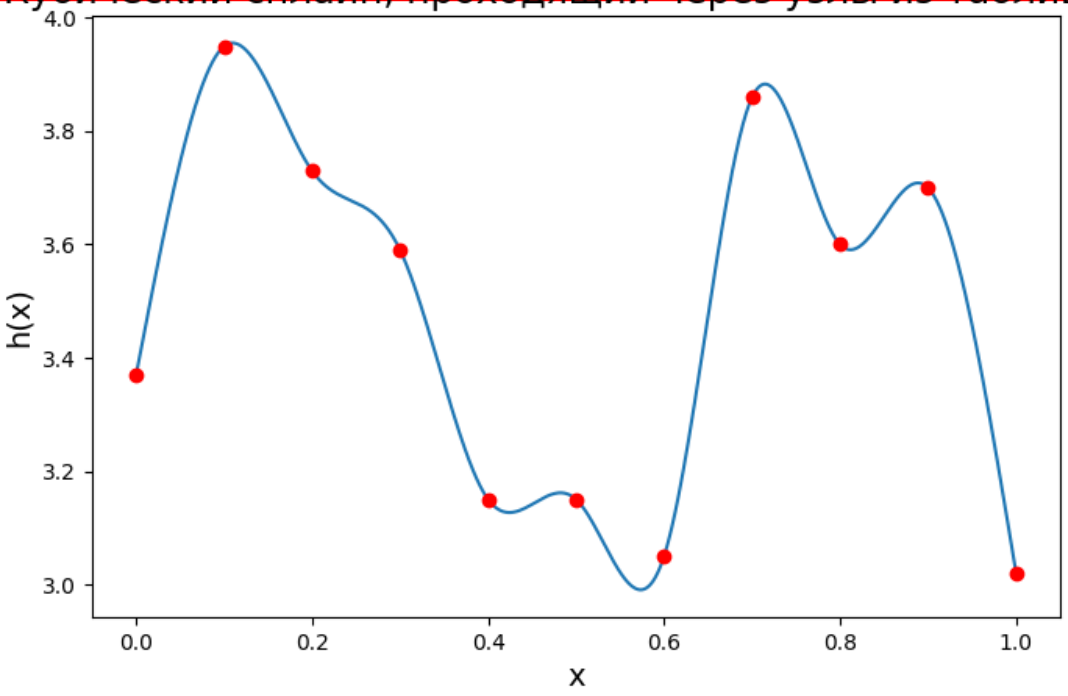

Рисунок 2

График производной:

Производная кубического сплайн, проходящий через узлы из таблицы 1

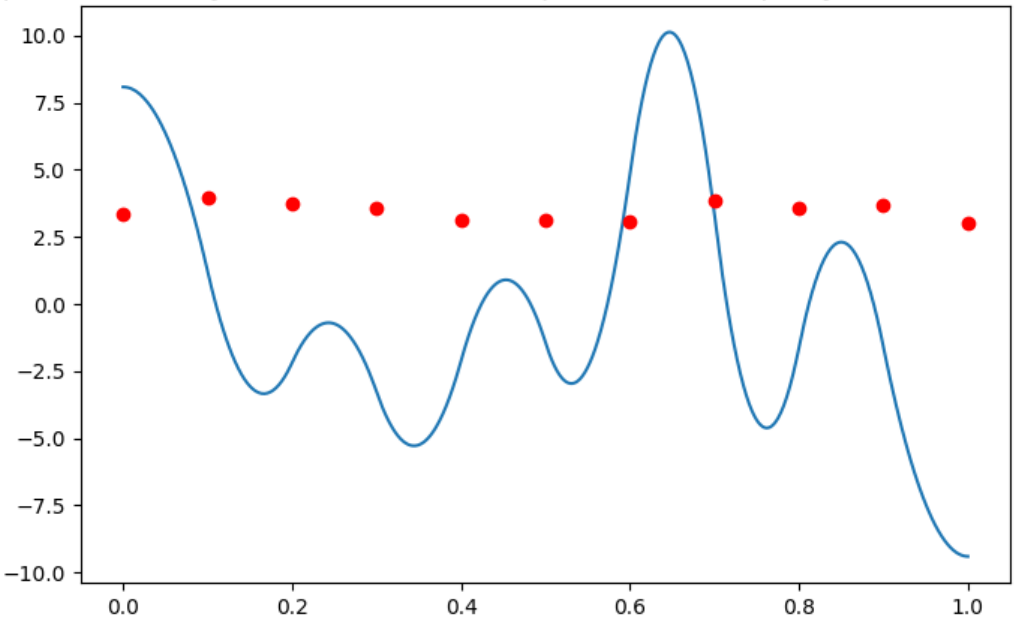

Рисунок 3

Вычисление значение i-го базисного полинома Лагранжа

```
def l_i(i, x, x_nodes):
    li = 1
    for k in range(len(x_nodes)):
        if k != i - 1:
            li = li * ((x - x_nodes[k]) / (x_nodes[i - 1] -
x_nodes[k]))
    return li
```

Вычисление происходит по формуле:

$$l_i(x) = \prod_{j=0, i \neq j}^{n} \frac{x - x_j}{x_i - x_j} \quad (2.1)$$

Вычисление интерполяционного полинома Лагранжа

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

    for i in range(n):
        result += y_nodes[i] * l_i(i + 1, x, x_nodes)

    return result
```

Вычисление интерполяционного полинома Лагранжа происходит по формуле:

$$L(x) = \sum_{i=0}^{n} y_i l_i(x), \quad (2.2)$$

Интерполянты Лагранжа

Погрешность имеют координаты X

Построение интерполянт Лагранжа

Сгенерирует 1000 векторов значений $[\tilde{x}_1, \ldots, \tilde{x}_{11}]^T$, предполагая, что $\tilde{x_i} = x_i + Z$, где x_i является случайной величиной с нормальным распределением с нулевым математическим ожиданием и стандартным отклонением 0.01.

Для каждого из полученных векторов построим интерполянт Лагранжа, предполагая, что в качестве абсцисс узлов используются значения $\tilde{x}_1$, а ординат – h_i из таблицы 1.

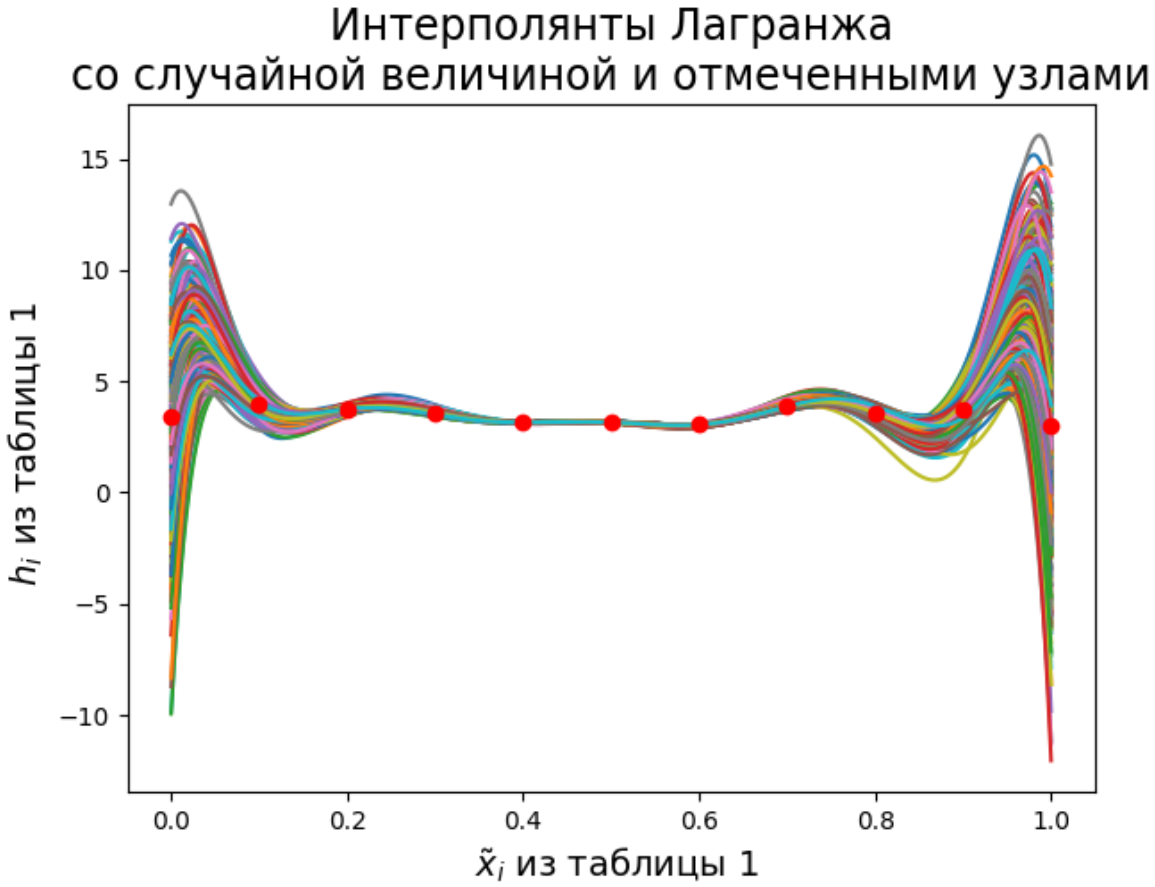

Рисунок 4

Усреднённый интерполянт

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

$$CI = \bar{x} \pm z\frac{s}{\sqrt{n}} \quad (2.3)$$

где CL — доверительный интервал $\bar{x}$ — выборочное среднее, z – значение доверительного уровня, s - среднеквадратичное отклонение, n – размер выборки.

Из формулы (2.3) получим верхнюю и нижнюю границу доверительной полосы. Найдём среднюю и затем выведем график.

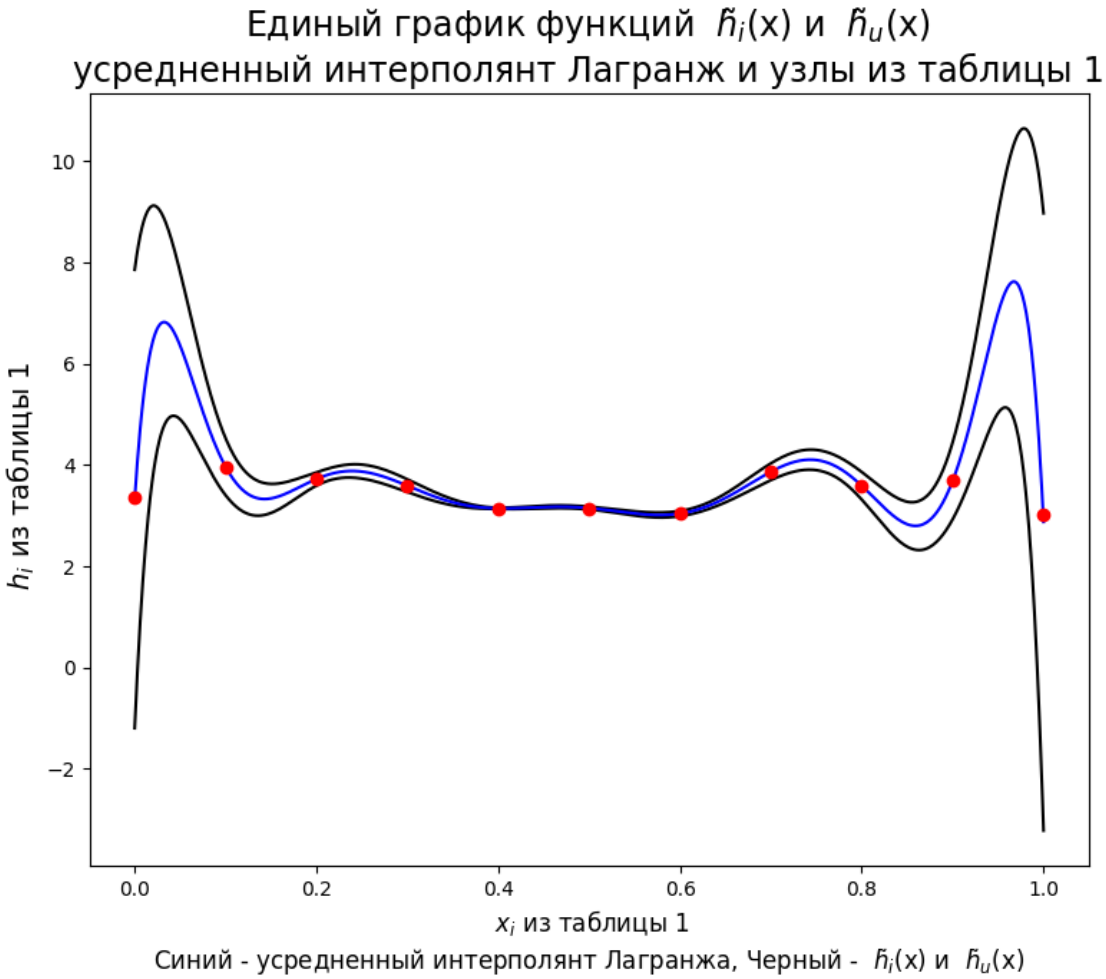

Рисунок 5

Участки интерполянты чувствительных к погрешности

Наиболее чувствительными участками интерполянта к погрешностям являться граничные отрезки, т.к. в начале и в конце на графиках

нитерполянтов Лагранжа накапливаются паразитные осцилляции. Происходит это из-за того, что такие узлы не являются оптимальными для интерполянта Лагранжа.

Погрешность имеют уровень поверхности h

Сгенерирует 1000 векторов значений $[\tilde{h}_1, \ldots, \tilde{h}_{11}]^T$, предполагая, что $\widetilde{h_i} = h_i + Z$, где x_i является случайной величиной с нормальным распределением с нулевым математическим ожиданием и стандартным отклонением 0.01.

Для каждого из полученных векторов построим интерполянт Лагранжа, предполагая, что в качестве абсцисс узлов используются значения $\tilde{x}_1$, а ординат – h_i из таблицы 1.

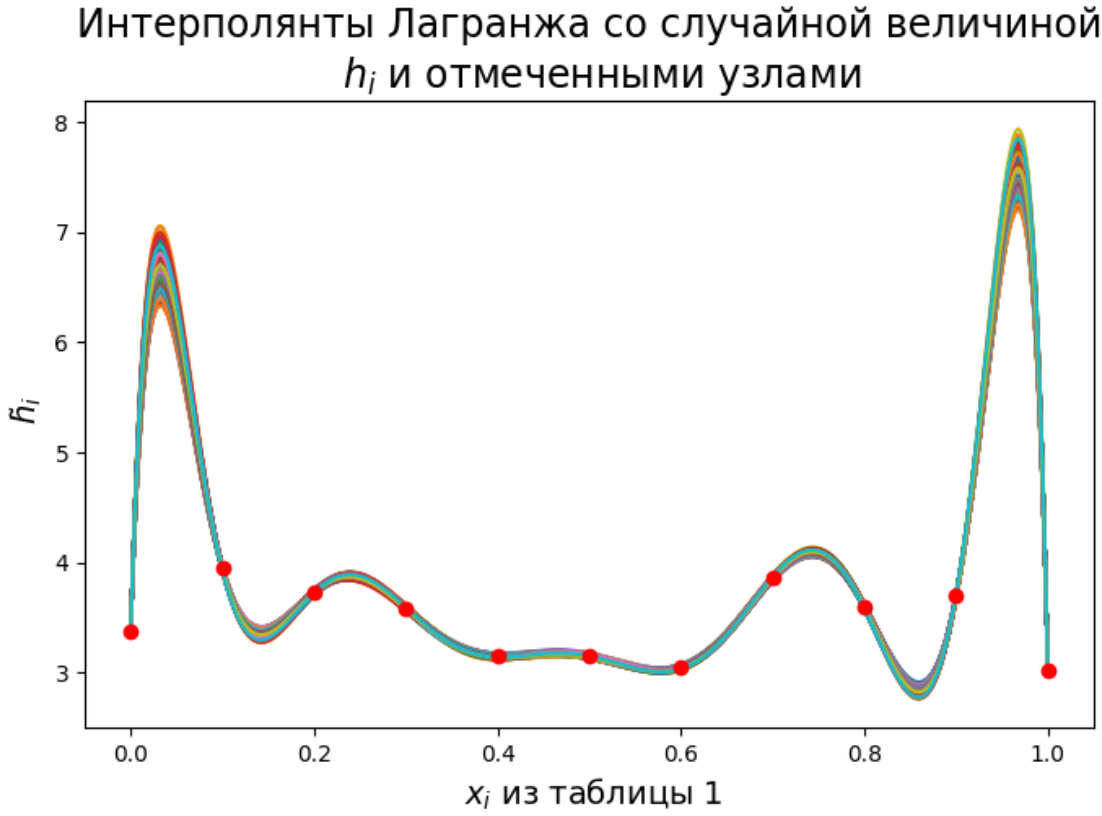

Рисунок 6

Усреднённый интерполянт

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

Найдём среднюю и затем выведем график.

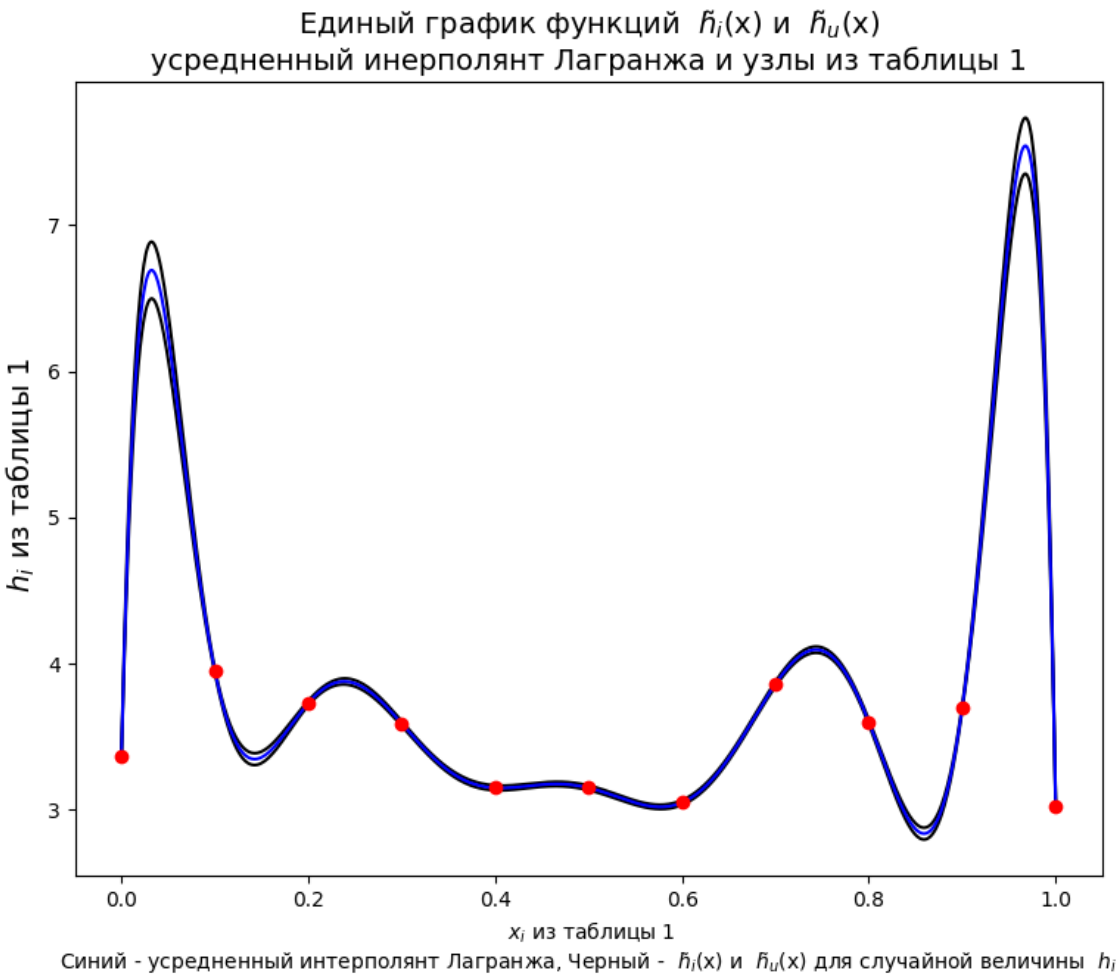

Рисунок 7

Участки интерполянты чувствительных к погрешности

Наиболее чувствительными участками интерполянта к погрешностям являться граничные отрезки, т.к. в начале и в конце на графиках нитерполянтов Лагранжа накапливаются паразитные осцилляции. Происходит это из-за того, что такие узлы не являются оптимальными для интерполянта Лагранжа.

Интерполяция кубическим сплайном

Погрешность имеют координаты X

Построение интерполянт Лагранжа

Сгенерирует 1000 векторов значений $[\tilde{x}_1, \dots, \tilde{x}_{11}]^T$, предполагая, что $\tilde{x_i} = x_i + Z$, где x_i является случайной величиной с нормальным распределением с нулевым математическим ожиданием и стандартным отклонением 0.01.

Для каждого из полученных векторов построим интерполянт, предполагая, что в качестве абсцисс узлов используются значения $\tilde{x}_1$, а ординат – h_i из таблицы 1.

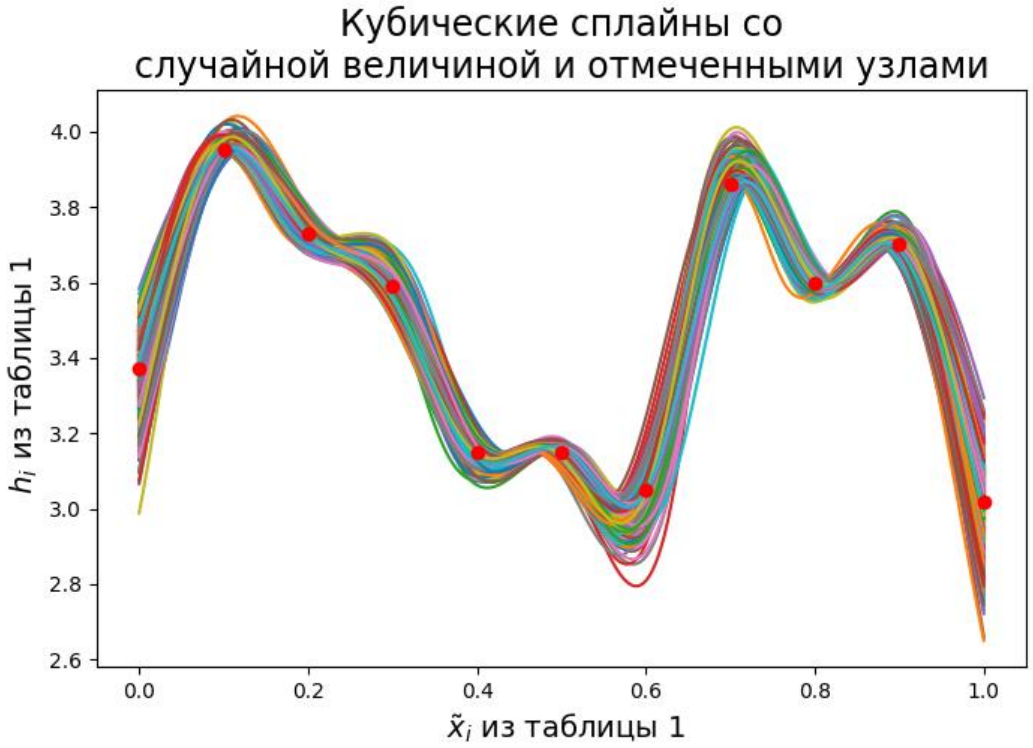

Рисунок 8

Усреднённый интерполянт

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

$$CI = \bar{x} \pm z\frac{s}{\sqrt{n}} \quad (2.3)$$

где CL — доверительный интервал $\bar{x}$ — выборочное среднее, z – значение доверительного уровня, s - среднеквадратичное отклонение, n – размер выборки.

Из формулы (2.3) получим верхнюю и нижнюю границу доверительной полосы. Найдём среднюю и затем выведем график.

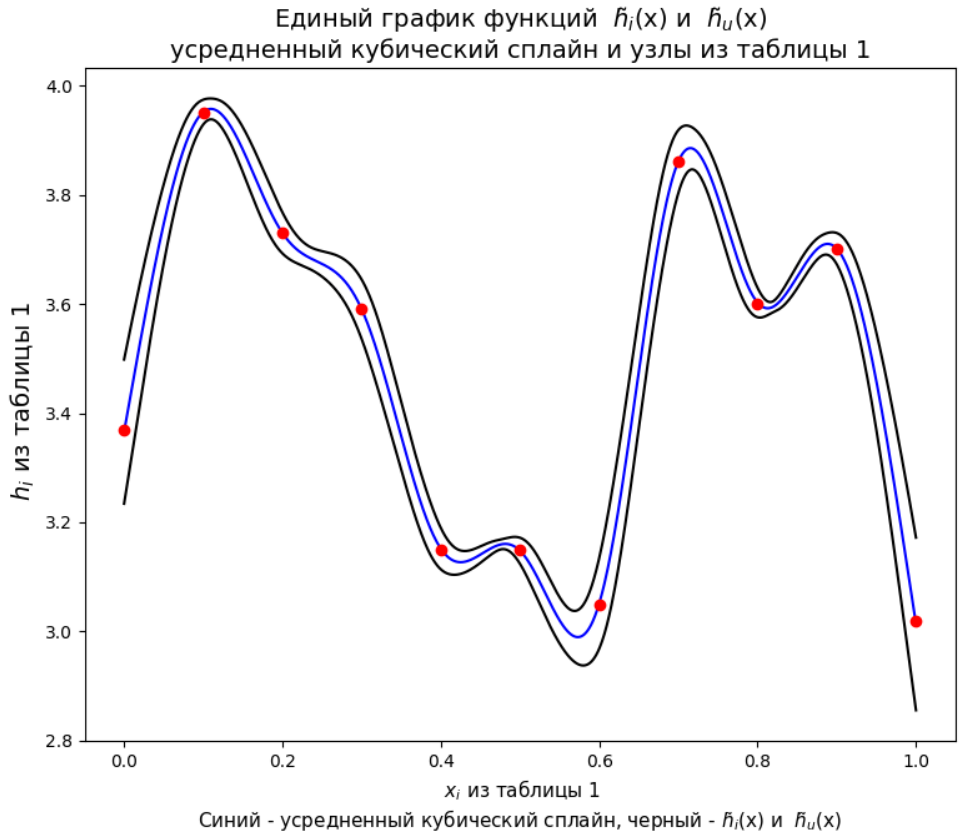

Рисунок 9

Участки интерполянты чувствительных к погрешности

Сплайн паразитных осцилляции кубический сплайн не накапливает в данном случае. Однако они появиться при использовании больших размерностей.

Погрешность имеют уровень поверхности h

Сгенерирует 1000 векторов значений $\left[\tilde{h}_1, \ldots, \tilde{h}_{11}\right]^T$, предполагая, что $\widetilde{h_i} = h_i + Z$, где x_i является случайной величиной с нормальным распределением с нулевым математическим ожиданием и стандартным отклонением 0.01.

Для каждого из полученных векторов построим интерполянт Лагранжа, предполагая, что в качестве абсцисс узлов используются значения $\tilde{x}_1$, а ординат – h_i из таблицы 1.

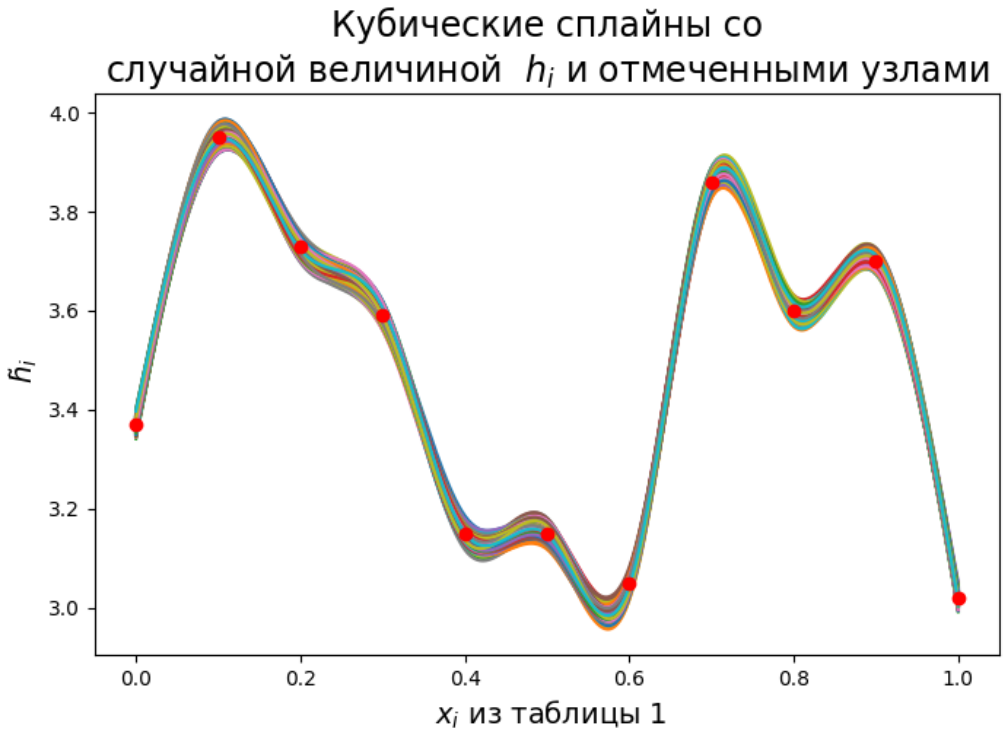

Рисунок 10

Усреднённый интерполянт

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

Найдём среднюю и затем выведем график.

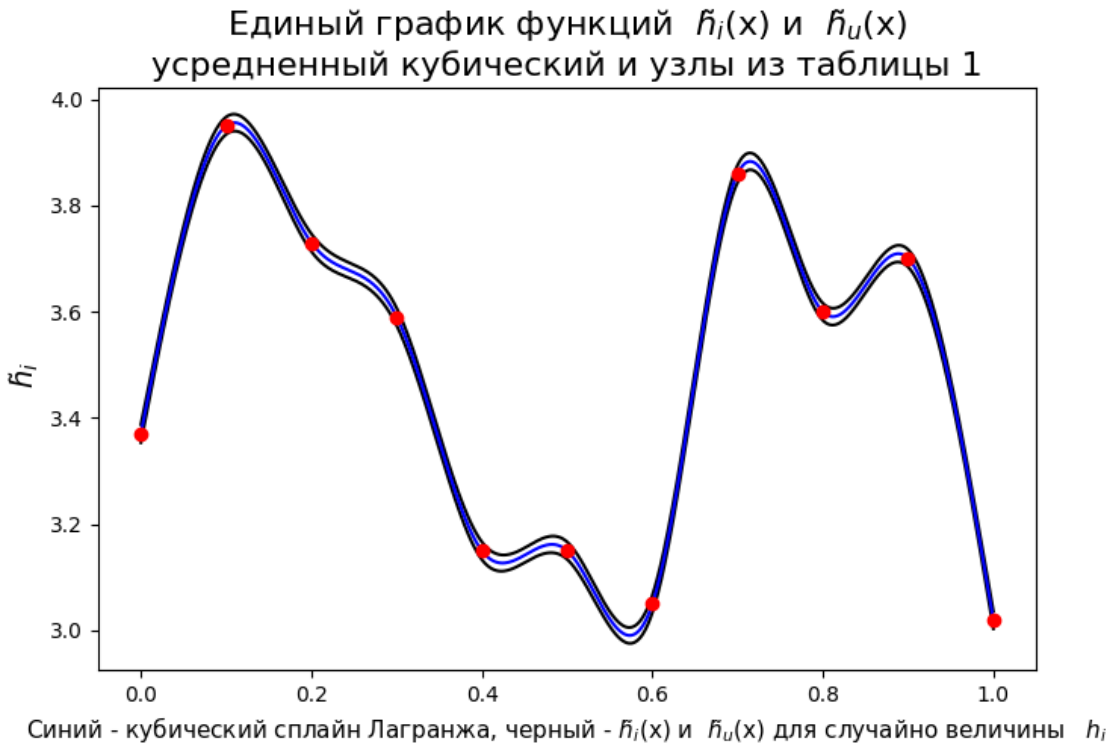

Рисунок 11

Участки интерполянты чувствительных к погрешности

Сплайн паразитных осцилляции кубический сплайн не накапливает в данном случае. Однако они появиться при использовании больших размерностей.

Заключение

При выполнении лабораторной работы были реализованы функции интерполяции многочленом Лагранжа и кубическим сплайном, реализованы доверительные и усредненные интервалы для интерполяции, а также было проверено влияние погрешности на интерполяцию.

Список литературы

1) Першин А.Ю., Соколов А.П. Вычислительная математика. Лабораторные работы / Учебная литература, г. Москва, 2018. — 11 с.
2) Першин А.Ю. Лекции по вычислительной математике (черновик) / Учебная литература. Кафедра РКб (Системы автоматизированного проектирования)