



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

Морозов Александр Юрьевич

Группа

РК6-54Б

Тип задания

лабораторная работа № 1

Тема лабораторной работы

Интерполяция в условиях измерений с
неопределенностью

Студент

подпись, дата

Морозов А.Ю.

фамилия, и.о.

Преподаватель

подпись, дата

Соколов А. П.

фамилия, и.о.

Москва, 2021 г.

Оглавление

Задание на лабораторную работу	3
Цель выполнения лабораторной работы	5
Выполненные задачи.....	5
1. Базовая часть	7
2. Продвинутая часть	8
Заключение	15
Список использованных источников.....	15

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).

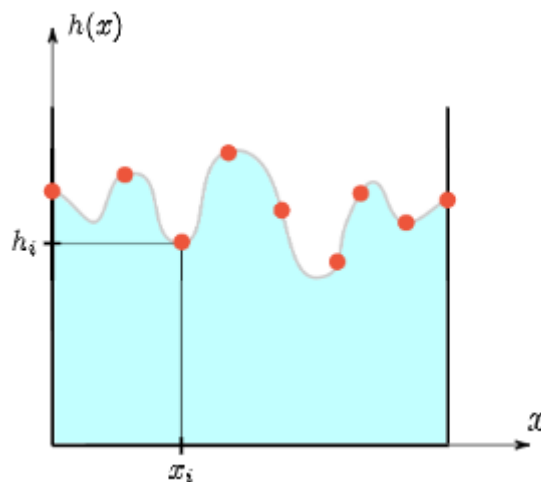


Рис. 1: Поверхность вязкой жидкости (серая кривая), движущейся сквозь некоторую среду (например, пористую). Её значения известны только в нескольких точках (красные узлы).

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 , 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_i 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.
 - с. 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?
6. Опциональное задание. Изложенный выше анализ позволяет строить доверительные интервалы исключительно для интерполянтов, не оценивая доверительные интервалы с точки зрения предсказаний значений между узлами. Интересным методом интерполяции, позволяющим получить именно такие вероятностные оценки, является регрессия на основе гауссовских процессов, известная также как кригинг. В этом опциональном задании предлагается провести интерполяцию по данным из таблицы 1, используя кригинг.

Цель выполнения лабораторной работы

Цель выполнения лабораторной работы – на практике испытать метод интерполирования кубическими сплайнами и метод интерполирования с использованием полиномов Лагранжа. Внести погрешность в узлы x_i и y_i , проанализировать результаты, полученные в каждом из методов интерполяции. Написать программу для проведения этого исследования и проинтерпретировать полученные в результате работы программы графики функций.

Выполненные задачи

Базовая часть

1. Разработка функции, вычисляющей коэффициенты естественного кубического сплайна
2. Разработка функции, вычисляющей значение кубического сплайна

3. Построение аппроксимации зависимости уровня поверхности жидкости $h(x)$ от координаты x при помощи кубического сплайна

Продвинутая часть

1. Разработка функции, возвращающей значение i – го базисного полинома Лагранжа
2. Разработка функции, возвращающей значение интерполяционного полинома Лагранжа
3. Анализ влияния погрешности в x_i на интерполяцию
 - а) Генерация 1000 векторов значений $[\tilde{x}_1, \dots, \tilde{x}_{11}]^T$
 - б) Построение интерполантов Лагранжа для каждого сгенерированного вектора
 - в) Расчёт доверительного интервала
 - г) Вывод графика с доверительным интервалом и усредненным интерполантом
 - е) Анализ: какие участки интерполанта и почему являются наиболее чувствительными к погрешностям
4. Анализ влияния погрешности в h_i на интерполяцию
5. Анализ как в п. 3 и 4 с использованием кубического сплайна
6. Опционально. Не выполнял

1. Базовая часть

1.1.

Уравнение кубического сплайна имеет вид:

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

Коэффициенты a_i можно определить сразу:

$$a_i = f(x_i)$$

Чтобы определить коэффициенты c_i необходимо решить данное матричное уравнение (используются граничные условия $c_1 = c_n = 0$):

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

Зная коэффициенты c_i и $h_i = x_{i+1} - x_i$, можно легко найти b_i и d_i :

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

$$d_i = \frac{c_{i+1} - c_i}{3h_i}$$

Для удобства решения матричного уравнения была использована функция `numpy.linalg.solve()`. Полученная функция возвращает $(N - 1) \times 5$ – матрицу, содержащую значения найденных коэффициентов и x_i (сделано для удобства). N – количество узлов интерполяции.

1.2.

В соответствии с формулой кубического сплайна

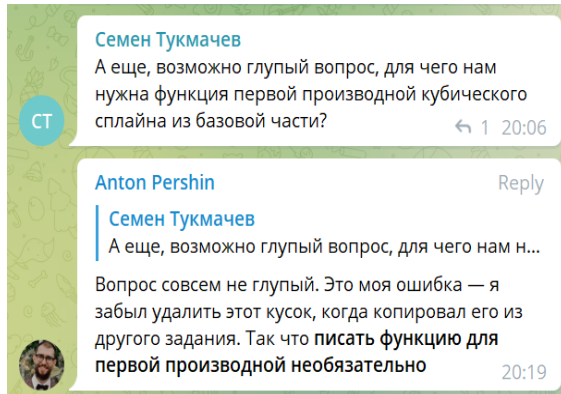
$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$$

Была написана функция, которая возвращает его значения.

Для вычисления значения кубического сплайна в точке x необходимо определить к какому из отрезков $[x_i; x_{i+1}]$ она принадлежит, а также знать матрицу коэффициентов сплайна. В таком случае значение сплайна $S(x)$ будет равно значению кубического

многочлена $Si(x)$.

Что касается функции для первой производной сплайна, выяснилось, что она не требуется:



1.3.

Составив матрицу коэффициентов при помощи функций из п. 1 и сформировав тысячу точек при помощи функции `numpy.linspace()`, высчитываем значения кубического сплайна в каждой из этих точек, используя функцию из п.2. Получаем график функции кубического сплайна при помощи библиотеки `matplotlib`:

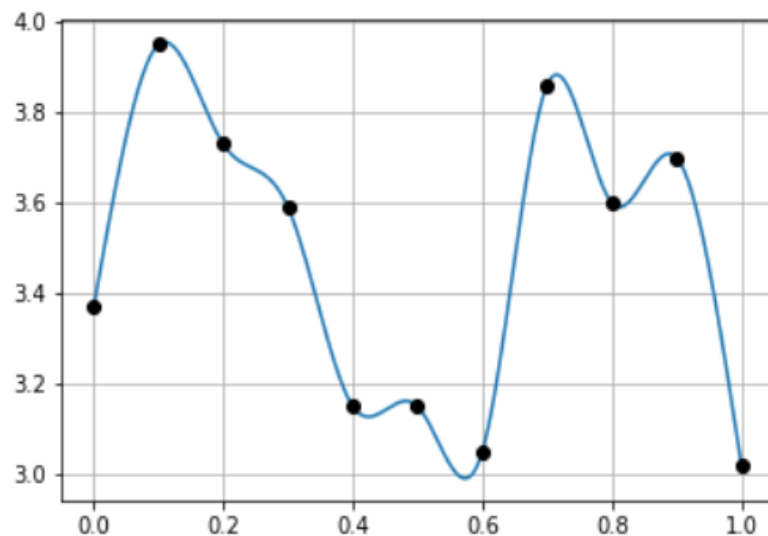


Рис 2. Кубический сплайн, проходящий через узлы, представленные в таблице 1.

2. Продвинутая часть

2.1.

Формула базисного полинома Лагранжа $n-1$ степени имеет вид:

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

На основе этой формулы была написана функция, возвращающая значение i -го базисного полинома Лагранжа, заданного на узлах с абсциссами `x_nodes`, в точке `x`:

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

2.2.

Функция, написанная в этом пункте, возвращает значение интерполяционного полинома Лагранжа, заданного на узлах с абсциссами `x_nodes` и ординатами `y_nodes`, в точке `x` в соответствии с данной формулой:

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

```
def L(x, x_nodes, y_nodes):
    L = 0
    for i in range(len(x_nodes)):
        L = L + y_nodes[i] * l_i(i + 1, x, x_nodes)
    return L
```

2.3.

а.

Формула для нормального распределения:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

По условию задания знаем, что математическое ожидание $\mu = 0$ и стандартное отклонение $\sigma = 10^{-2}$. Для генерации 1000 векторов $[\tilde{x}_1, \dots, \tilde{x}_{11}]^T$, где \tilde{x}_i – значение узла x_i с внесённой погрешностью, воспользуемся функцией `numpy.random.normal` (первый аргумент – матожидание, второй аргумент – стандартное отклонение), которая как раз реализует представленную выше формулу:

```
x = np.linspace(0,1,11)
X = np.array([[i + np.random.normal(0,0.01) for i in x] for j in range(1000)])
```

b.

Построим график интерполянтов Лагранжа, используя в качестве абсцисс значения \tilde{x}_i , полученные в предыдущем пункте, а в качестве ординат значения h_i из таблицы 1.

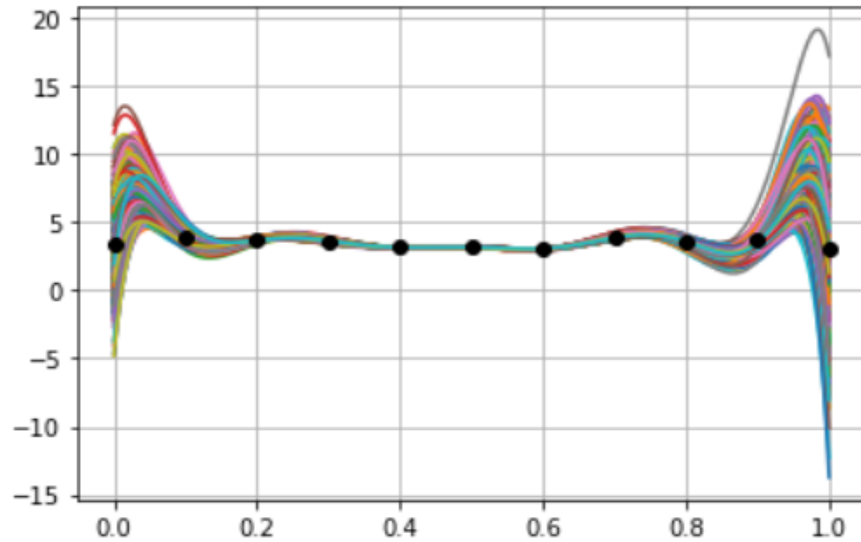


Рис 3. 1000 полиномов Лагранжа, построенные на узлах со внесённой погрешностью \tilde{x}_i

с. и d.

Построим такие функции $\tilde{h}_l(x), \tilde{h}_u(x)$, где $\tilde{h}_l(x) < \tilde{h}_u(x) \forall x \in [0; 1]$, что вероятность того, что значение интерполанта в точке x будет лежать в интервале $[\tilde{h}_l(x); \tilde{h}_u(x)]$ равна 0.9.

Формула для нахождения доверительного интервала в нашем случае будет иметь вид:

$$\left(\bar{x} - z \frac{\sigma}{\sqrt{n}}; \bar{x} + z \frac{\sigma}{\sqrt{n}} \right)$$

Где \bar{x} – выборочное среднее значение, z – критическое значение стандартного нормального распределения, σ – стандартное отклонение, \sqrt{n} – корень из величины выборки.

Величину z можно узнать из таблицы:

	Уровень значимости								
α	0.400	0.200	0.100	0.050	0.020	0.010	0.005	0.002	0.001
z	0.842	1.282	1.645	1.960	2.326	2.576	2.807	3.090	3.291

Таблица 2: Критические значения стандартного нормального распределения

В нашем случае $\alpha = 1 - P = 1 - 0.9 = 0.1$, следовательно, $z = 1.645$.

Для нахождения доверительного интервала в языке python существует функция `scipy.stats.norm.interval()` (первый аргумент – заданная надёжность, второй аргумент – среднее значение выборки, третий аргумент – стандартное отклонение).

```
for i in range(1000):
    slc = [Y[j][i] for j in range(N)]
    c_i = st.norm.interval(0.9, np.mean(slc), np.std(slc))
    h_l[i] = c_i[0]
    h_u[i] = c_i[1]
    aver_inter[i] = np.mean(slc)
```

Для определения среднего значения выборки воспользуемся функцией `numpy.mean()`, а для определения стандартного отклонения функцией `numpy.std()`.

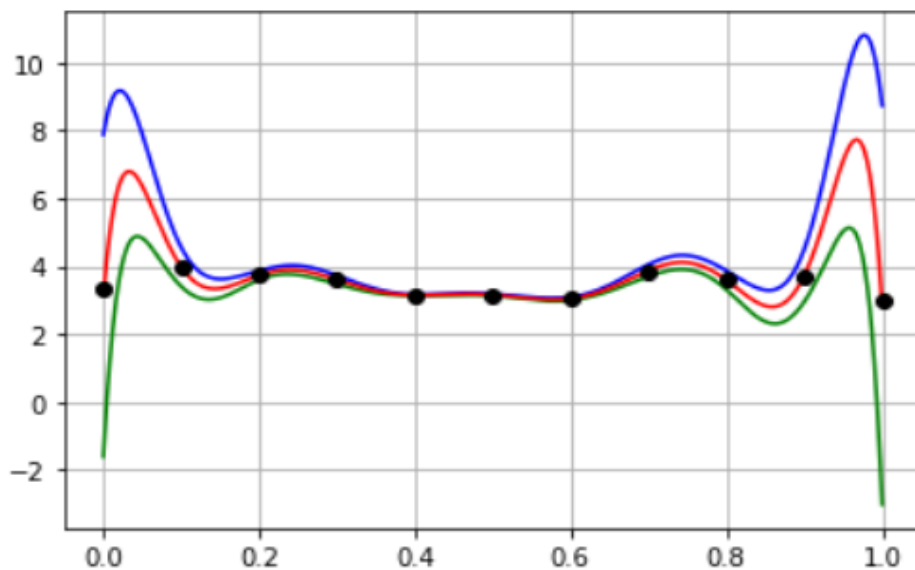


Рис 4. График доверительного интервала (Погрешность внесена в x_i). $\tilde{h}_u(x)$ – синяя кривая. $\tilde{h}_l(x)$ – зеленая кривая. Усредненный интерполянт – красная кривая. Узлы из таблицы 1 – чётные точки.

е.

Как можно заметить на рис. 3 и рис. 4 наиболее чувствительными к погрешностям являются граничные участки кривой по оси абсцисс. Это связано с тем, что на концах отрезка накапливаются паразитные осцилляции, возникающие из-за располагающегося близко к краям отрезка большого числа экстремумов, которые также имеют большую по модулю амплитуду, чем центральные. Это явления называют «Феномен Рунге». Решить эту проблему можно применяя узлы Чебышева или используя другой метод интерполяции.

2.4.

Повторим опыт из предыдущего пункта, только на этот внесем погрешность в ординаты узлов из таблицы 1. Получаем графики:

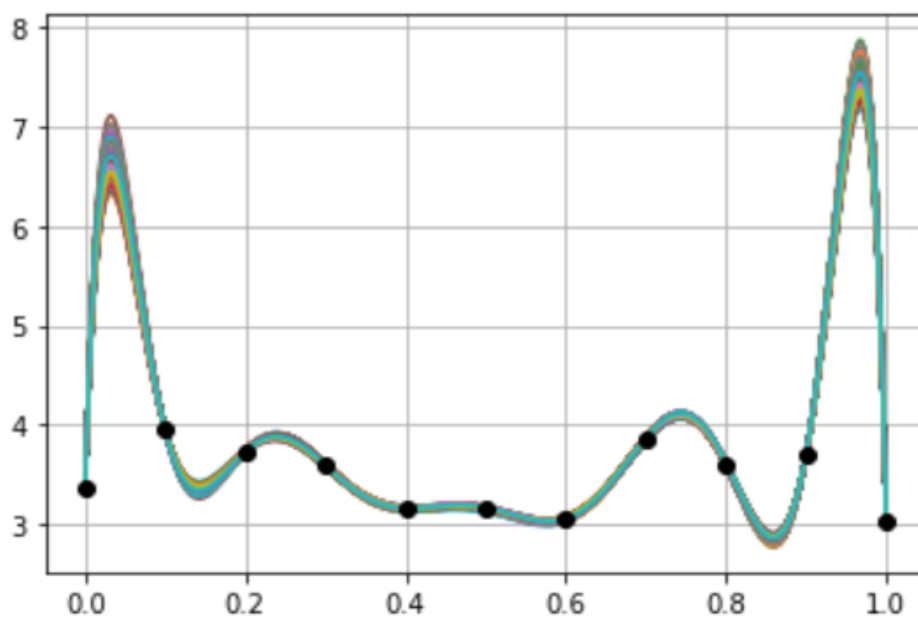


Рис 5. 1000 полиномов Лагранжа, построенные на узлах со внесённой погрешностью \tilde{y}_i

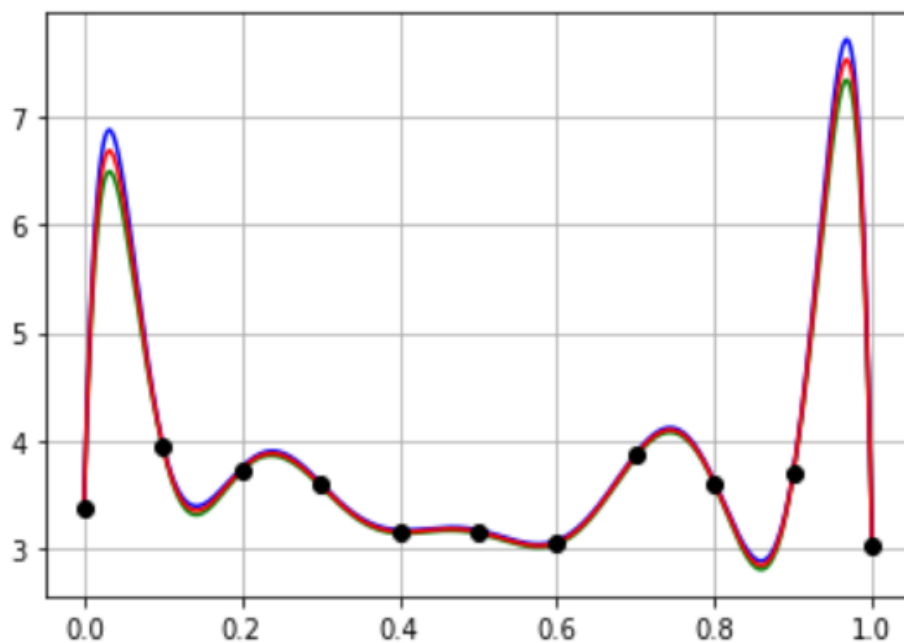


Рис 6. График доверительного интервала (Погрешность внесена в y_i). $\tilde{h}_u(x)$ – синяя кривая. $\tilde{h}_l(x)$ – зеленая кривая. Усредненный интерполянт – красная кривая. Узлы из таблицы 1 – чётные точки.

Как можно заметить из рис. 5 и 6 разброс значения значительно уменьшился на всём отрезке (особенно это заметно на границах). Из этого можно сделать вывод, что погрешности в ординатах узлов не так сильно влияют на результат интерполяции, как погрешности в абсциссах узлов.

2.5.

Повторим опыт из предыдущего пункта, но вместо интерполяции полиномами Лагранжа будем использовать интерполяцию кубическими сплайнами. Получаем графики:

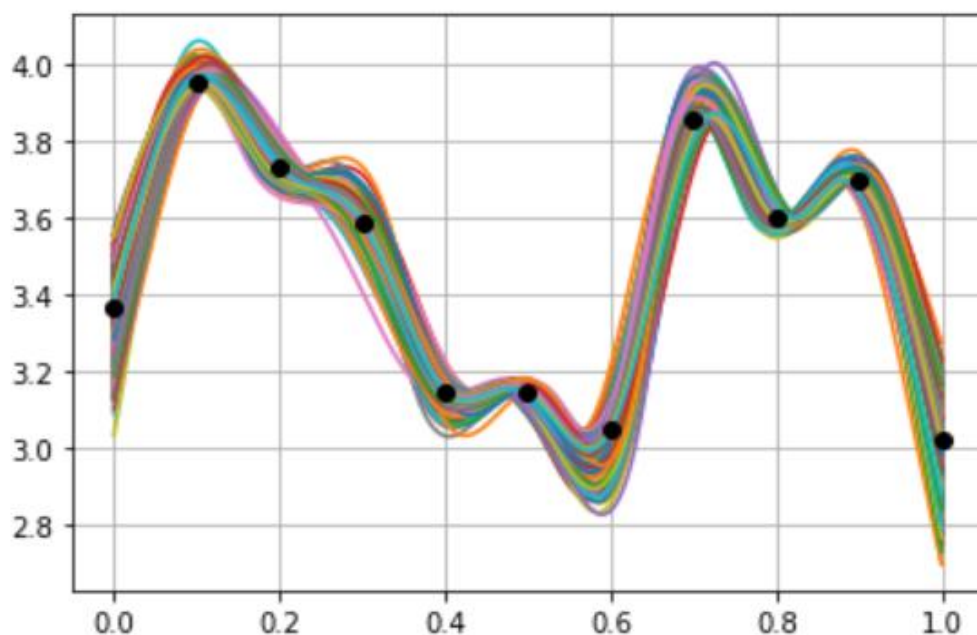


Рис 7. 1000 кубических сплайнов, построенные на узлах со внесённой погрешностью \tilde{x}_i

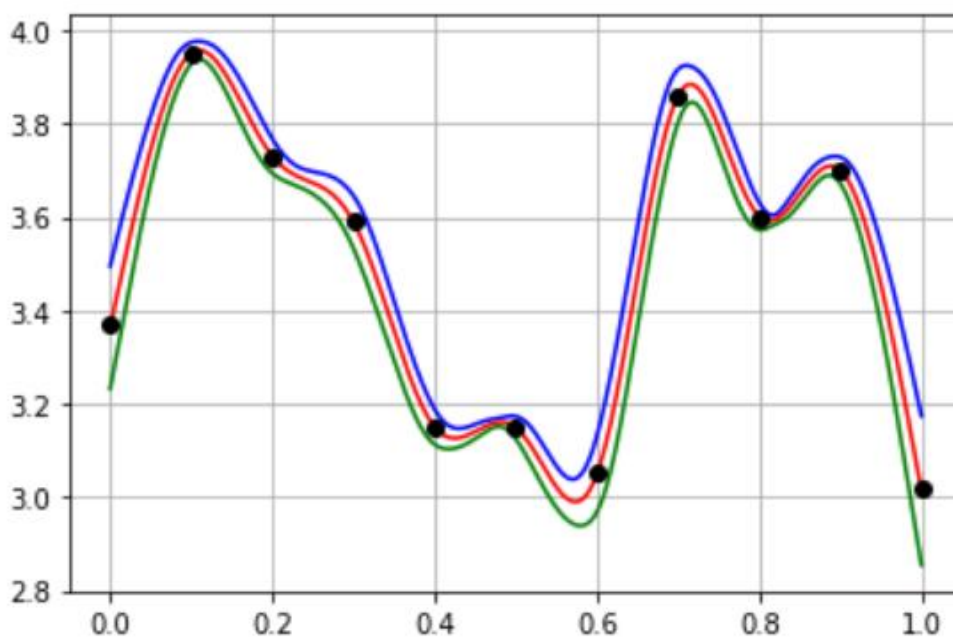


Рис 8. График доверительного интервала (Погрешность внесена в x_i). $\tilde{h}_u(x)$ – синяя кривая. $\tilde{h}_l(x)$ – зеленая кривая. Усредненный интерполянт – красная кривая. Узлы из таблицы 1 – чётные точки.

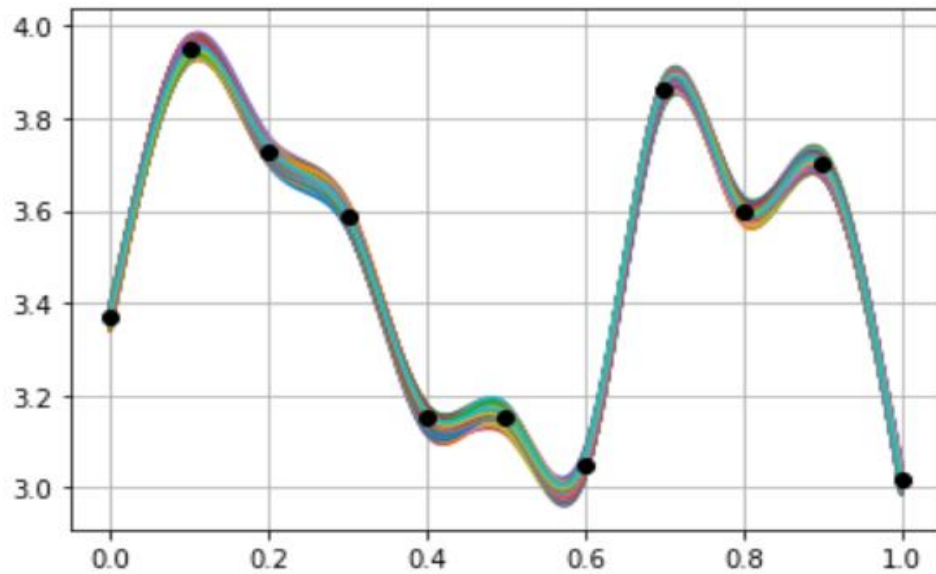


Рис 9. 1000 кубических сплайнов, построенные на узлах со внесённой погрешностью \tilde{y}_i

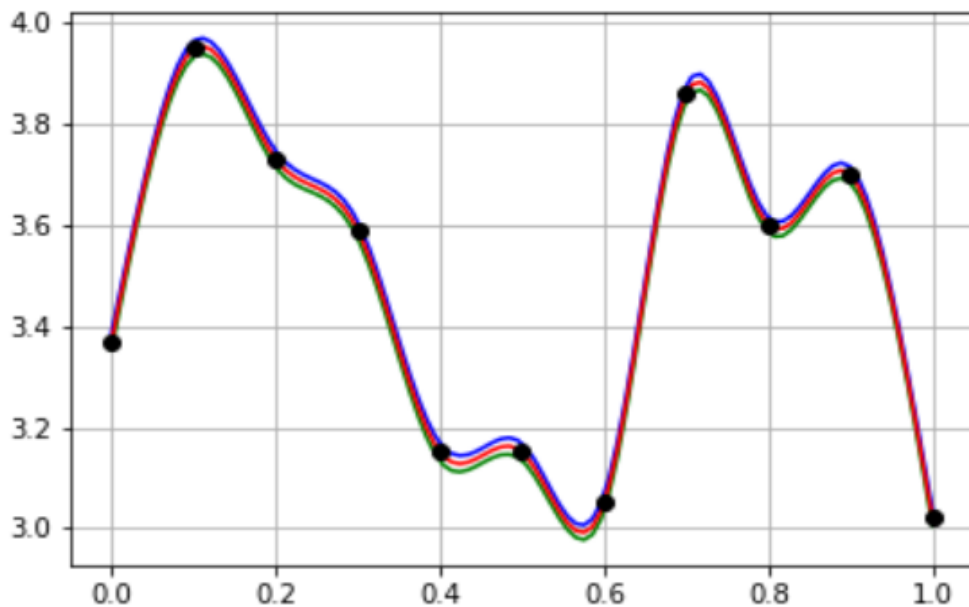


Рис 10. График доверительного интервала (Погрешность внесена в y_i). $\tilde{h}_u(x)$ – синяя кривая. $\tilde{h}_l(x)$ – зеленая кривая. Усредненный интерполянт – красная кривая. Узлы из таблицы 1 – чётные точки.

В отличие от интерполяции полиномами Лагранжа, при интерполяции кубическими сплайнами паразитные осцилляции на границах отрезка отсутствуют. Также можно заметить, что, как и при интерполяции многочленами Лагранжа, внесение погрешности в абсциссы узлов сильнее влияет на разброс значений функции, чем когда погрешность вносится в ординаты.

Заключение

В ходе выполнения лабораторной работы были рассмотрены и программно реализованы методы интерполяции полиномами Лагранжа и кубическими сплайнами. Были сделаны выводы, о том, что интерполяция кубическими сплайнами точнее интерполяции Лагранжа, это можно увидеть на графиках, где вносилась погрешность в исходные узлы из таблицы 1. Ведь при использовании многочленов Лагранжа возникают сильные паразитные осцилляции на границах рассматриваемого отрезка (Проявление феномена Рунге), однако стоит заметить, что в районе центра отрезка интерполяция полиномами Лагранжа оказывается несколько точнее, это видно на графиках с доверительными интервалами. Также оказалось, что погрешность в абсциссах узлов оказывает большее влияние на точность интерполяции, чем погрешность в ординатах.

Список использованных источников

1. Першин А.Ю. Лекции по курсу «Вычислительная математика». Москва, 2018-2021.
2. Першин А.Ю., Соколов А.П. Вычислительная математика, лабораторные работы (учебное пособие), МГТУ им. Баумана, Москва, 2018-2021.
3. Конспект лекций по теории вероятности МГТУ им. Н.Э. Баумана [Электронный ресурс] /. — Электрон. текстовые дан. Режим доступа: <http://fn.bmstu.ru/educational-work-fs-12/70-lections/244-ter-ver>
4. Артамонов Н.В. Таблицы критических значений. 2020