



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	Тетерин Никита Евгеньевич
Группа	РК6-546
Тип задания	лабораторная работа
Тема лабораторной работы	Интерполяция в условиях с измерений с неопределенностью (вариант 5)

Студент	_____	<u><b>Тетерин Н.Е.</b></u>
	<i>подпись, дата</i>	<i>фамилия, и.о.</i>
Преподаватель	_____	<u><b>Соколов А.П.</b></u>
	<i>подпись, дата</i>	<i>фамилия, и.о.</i>

Оценка \_\_\_\_\_

*Москва, 2021 г.*

## Оглавление

Задание на лабораторную работу .....	3
Цель выполнения лабораторной работы .....	3
Выполненные задачи.....	3
1. Расчёт коэффициентов естественного сплайна для заданного набора узлов .....	4
2. Вычисление значения естественного кубического сплайна в точке .....	5
3. Аппроксимация интерполяционным полиномом Лагранжа .....	6
4. Анализ влияния погрешностей входных данных на результат интерполяции .....	7
5. Влияние погрешностей в случае естественного кубического сплайна .....	10
Заключение.....	10
Список использованных источников .....	11

## **Задание на лабораторную работу**

Познакомиться на простейшем примере с интерполяцией, а также проанализировать, как неопределенность входных данных может влиять на её предсказания.

### **Цель выполнения лабораторной работы**

Цель выполнения лабораторной работы – имплементировать алгоритмы интерполяции полиномом Лагранжа и кубическими сплайнами, а также оценить устойчивость интерполяции при присутствии погрешности в исходных данных.

### **Выполненные задачи**

1. Разработать алгоритм расчёта коэффициентов естественного кубического сплайна по заданному набору узлов
2. Разработать метод для вычисления значения и величины первой производной естественного кубического сплайна в точке, построить аппроксимацию на основе полученных методов для данного набора узлов
3. Разработать методы для вычисления значения базисного полинома Лагранжа и интерполяционного полинома Лагранжа в точке для данного набора узлов
4. Произвести анализ влияния погрешностей измерений координаты  $x_i$  и величины  $h_i(x)$  на результат интерполирования для полинома Лагранжа
5. Повторить предыдущий эксперимент для интерполяции естественным кубическим сплайном, сравнить результаты

# 1. Расчёт коэффициентов естественного сплайна для заданного набора узлов

Если функция  $f(x)$  задана в  $n$  интерполяционных узлах на интервале  $[a; b]$ , известно, что для фиксированного набора узлов можно построить единственный естественный кубический сплайн, то есть кусочно-заданную функцию, обладающую следующими свойствами:

$$S_i(x_i) = f(x_i), S_i(x_{i+1}) = f(x_{i+1}), \forall i \in [1; n-2] \quad (1.1.1)$$

$$\forall i \in [1; n-2], n = 1, 2, S_i^n(x_{i+1}) = S_{i+1}^n(x_{i+1}) \quad (1.1.2)$$

$$S''(x_1) = S''(x_n) = 0, S'(x_i) = f'(x_i) \quad (1.1.3)$$

На каждом участке сплайн описывается 4 неизвестными параметрами, которые необходимо определить последовательно. Из курса лекций известен способ сформировать матричное уравнение относительно вектора параметров  $c$ :

$$\begin{bmatrix} 1 & 0 & \dots & \dots & \dots & 0 \\ h_1 & 2(h_2 + h_1) & h_2 & 0 & \dots & 0 \\ \vdots & h_2 & 2(h_3 + h_2) & h_3 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & h_{n-2} & 2(h_{n-2} + h_{n-1}) & h_{n-1} \\ 0 & \dots & \dots & \dots & 0 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} 0 \\ (3/h_2)(a_3 - a_2) - (3/h_1)(a_2 - a_1) \\ (3/h_3)(a_4 - a_3) - (3/h_2)(a_3 - a_2) \\ \vdots \\ (3/h_{n-1})(a_n - a_{n-1}) - (3/h_{n-2})(a_n - a_{n-1}) \\ 0 \end{bmatrix} \quad (1.2)$$

При этом  $h_i = x_{i+1} - x_i$  и  $a_i = f(x_i)$ . Из (1.2) можно определить вектор параметров  $c$ , а далее по итеративным формулам (1.3) и (1.4) определить вектора  $b$  и  $d$ .

$$b_i = \frac{1}{h_i}(a_{i+1} - a_i) - \frac{h_i}{3}(c_{i+1} + 2c_i) \quad (1.3)$$

$$d_i = \frac{c_{i+1} - c_i}{3h_i} \quad (1.4)$$

При этом вектор параметров  $a$  по понятным соображениям считается известным. Полученные таким образом параметры однозначно описывают естественный кубический сплайн, построенный на выбранных узлах. В программной реализации получение коэффициентов осуществляется функцией `cubic_spline_coeff(x_nodes, y_nodes)`, первым аргументом которой является коллекция абсцисс узлов интерполяции, а вторым коллекция ординат узлов интерполяции (в нашей задаче это координаты точек измерений и высота жидкости в этих точках соответственно). Эту функцию можно найти в модуле `utils.methods.cubspline`. Методика довольно прямолинейна: сформировать

матрицу и известный вектор в уравнении (1.2), получить обратную матрицу и, умножив на известный вектор, получить вектор параметров  $c$ . Далее необходимо итеративно вычислить векторы оставшихся параметров согласно формулам (1.3–4) и объединить их в матрицу. Последний элемент  $c_n$  при этом отбрасывается, поскольку размерность результата  $(n - 1) \times 3$  (последний элемент был *фиктивным*, и его значение выбиралось из естественных граничных условий, *никакого реального смысла он не несет*).

## 2. Вычисление значения естественного кубического сплайна в точке

Имея матрицу параметров, полученную в ходе выполнения предыдущей задачи, не составляет труда получить как само значение сплайна, так и его производную в некоторой точке (сразу стоит отметить, что корректным будет вычислять значение только на диапазоне  $x \in [x_1; x_n]$ ): для этого сперва определим узлы, промежутку между которыми принадлежит данная точка, далее подставим её значение в формулу (2.1) с соответствующим индексом.

$$S(x) = S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3, x \in [x_i; x_{i+1}] \quad (2.1)$$

$$S'(x) = S'_i(x) = b_i + 2c_i(x - x_i) + 3d_i(x - x_i)^2, x \in [x_i; x_{i+1}] \quad (2.2)$$

Аналогично, для производной сплайна в точке можно использовать выражение (2.2), полученное путем дифференцирования (2.1). Методы, задействующие эти выражения, реализованы в рамках класса *CubicSplineMethod*, находящийся в модуле *utils.interpolate*. Процесс интерполяции и построения графика, содержащего естественный сплайн, объединен в метод *interpolate()*. На рис. 2.1 изображен результат работы метода (серым цветом окрашена «жидкость», уровень которой описывает полученная кривая). Работа с пакетом *matplotlib* вынесена для удобства в класс *PlotArtist (utils.plotting)*, который создает фигуру, добавляет в неё данные графиков и точки, обозначающие исходный набор узлов, а также сохраняет результат в формате *svg* (в целях сохранения качества). Данный класс будет использоваться для создания остальных графиков данной ЛР.

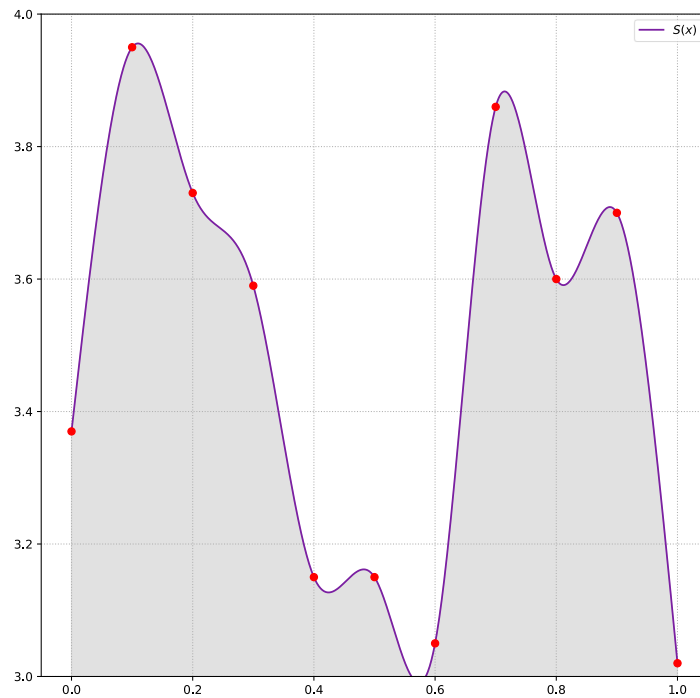


Рис. 2.1 Естественный кубический сплайн, построенный на 11 узлах

### 3. Аппроксимация интерполяционным полиномом Лагранжа

Значение  $i$ -го базисного полинома Лагранжа в точке  $x$  есть произведение вида (3.1)

$$l_i(x) = \prod_{j \neq i} \frac{(x - x_j)}{(x_i - x_j)} \quad (3.1)$$

Функция  $L\_i\_at\_x(i, x, x\_nodes)$ , возвращающая значение выражения (3.1), находится в модуле `utils.methods.lagrange`. В ней результат вычисляется итеративно (полагаю, в таком случае величина ошибок округления ниже). Значение самого интерполяционного полинома в точке можно найти как линейную комбинацию значений базисных полиномов с коэффициентами, равными значениям ординат соответствующих узлов:

$$L_{n-1}(x) = \sum_{i=0}^n f(x_i) l_i(x) \quad (3.2)$$

На рис. 3.1 приведен график аппроксимации полиномом Лагранжа 10 степени, узлы интерполяции обозначены красным. Можно наблюдать, что хоть (ожидаемо) полином проходит через все узлы, большая старшая степень дает о себе знать и полученная кривая имеет ярко выраженные осцилляции при приближении к границам отрезка

интерполяции. Как мы пронаблюдаем далее, именно в этих местах полученная функция наиболее уязвима для неточностей входных данных.

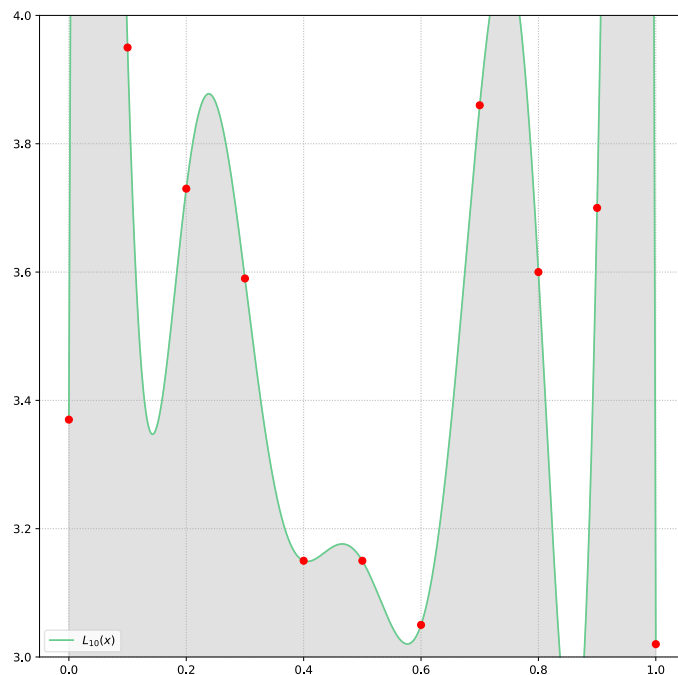


Рис. 3.1 Результат аппроксимации полиномом Лагранжа для исходных 11 узлов

#### 4. Анализ влияния погрешностей входных данных на результат интерполяции

Согласно условию задания была сформирована выборка из 1000 векторов абсцисс исходных узлов вида  $[\tilde{x}_1, \dots, \tilde{x}_{11}]^T$ , где  $\tilde{x}_i = x_i \pm Z$ , а  $Z$  является нормально распределенной случайной величиной с характеристиками (4.1).

$$\sigma_Z = 10^{-2}, \mu_Z = 0 \quad (4.1)$$

Далее была произведена интерполяция для каждого вектора смещенных абсцисс и вектора точных значений ординат  $h(x)$  и рассчитан доверительный интервал с уровнем надежности 0.99. Здесь необходимо сделать отступление, поскольку, строго говоря, нельзя говорить о расчёте во всех точках интервала  $[0; 1]$ , ведь после добавления случайной величины  $Z$   $\tilde{x}_1$  и  $\tilde{x}_{11}$  не обязательно захватывают исходный интервал полностью. Но в рамках данной задачи было сделано допущение, согласно которому каждый интерполянт был определен на всём исходном интервале (что в некотором смысле соответствует реалиям моделируемой задачи). Также было принято решение использовать уровень надежности 0.99 против предложенного 0.9, поскольку при

использовании последнего границы интервала практически не отклонялись от среднего значения на всём диапазоне расчёта. На рис. 4. (слева) приведен полученный график. Шаг по  $x$  составлял  $10^{-3}$ . Такой выбор был компромиссом между временем выполнения вычислений и внешним видом полученного графика. Для расчета доверительной полосы на каждом «срезе» из тысячи значений приближений для каждого  $x$  с выбранным шагом на промежутке от 0 до 1 определялись среднее значение и среднеквадратичное отклонение, на их основе с помощью, например, функции Лапласа можно определять границы необходимого диапазона. В программной реализации для этой цели я воспользовался функцией *scipy.stats.norm.interval*, которая рассчитывает границы диапазона, в котором с вероятностью 0.99 находится значение приближения в данном  $x$ . То, что распределение значений интерполянта можно считать нормальным, связано с числом элементов в генеральной совокупности (1000) и центральной предельной теоремой.

Следующим шагом стало повторение эксперимента уже при погрешности для  $h(x)$ . Процесс аналогичен произведенному выше, результат приведен на рис. 4.1 (справа). Обнаруживается, что доверительный интервал заметно сузился на всём промежутке интерполирования (контраст особенно ярко проявляется на границах промежутка), то есть результат приближения не так сильно чувствителен к выбору абсцисс (это оказывается довольно последовательным, поскольку в формуле (3.2) малая ошибка в абсциссе узла вносит вклад многократно при вычислении значения базисного полинома, в то время как ошибка ординаты узла будет всегда в первой степени умножаться на значение базисного полинома, в результате в узлах интерполяции имеем практически полное совпадение значений всех случайных многочленов, погрешность там сопоставима со вкладом случайной величины в значение ординаты). Однако можно наблюдать более широкие участки доверительной полосы в нулях 1 и 2 производных (экстремумах и точках перегиба функции соответственно), эти характерные точки у полученных интерполянтов, вообще говоря, не совпадают.



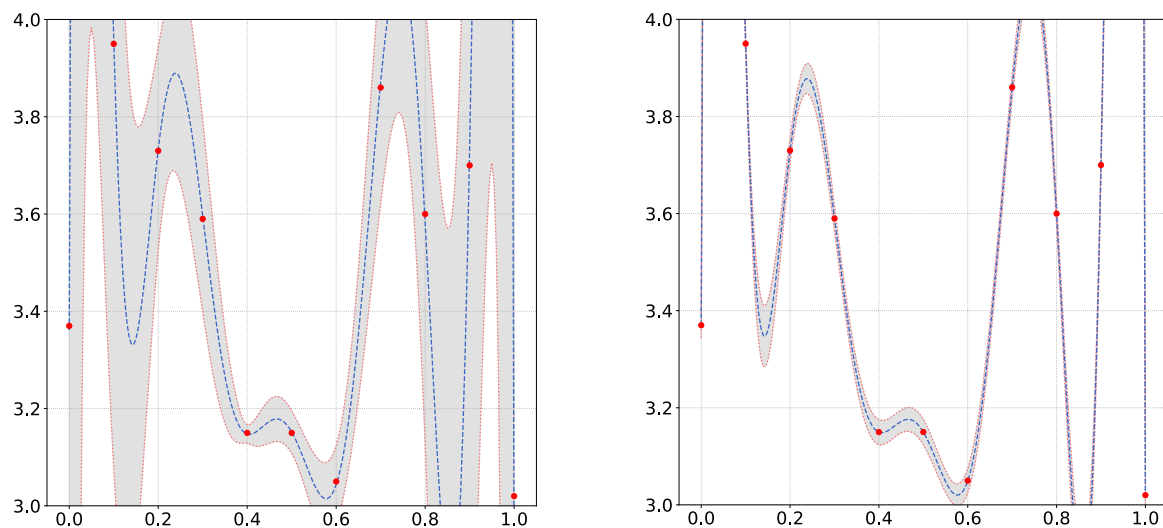


Рис. 4.1 Усредненный интерполянт (обозначен синим пунктиром) и доверительный интервал для уровня надежности 0.99 (зона окрашена серым), слева для случая погрешности в  $x$ , справа для случая погрешности в  $h(x)$ , для интерполяционного полинома Лагранжа

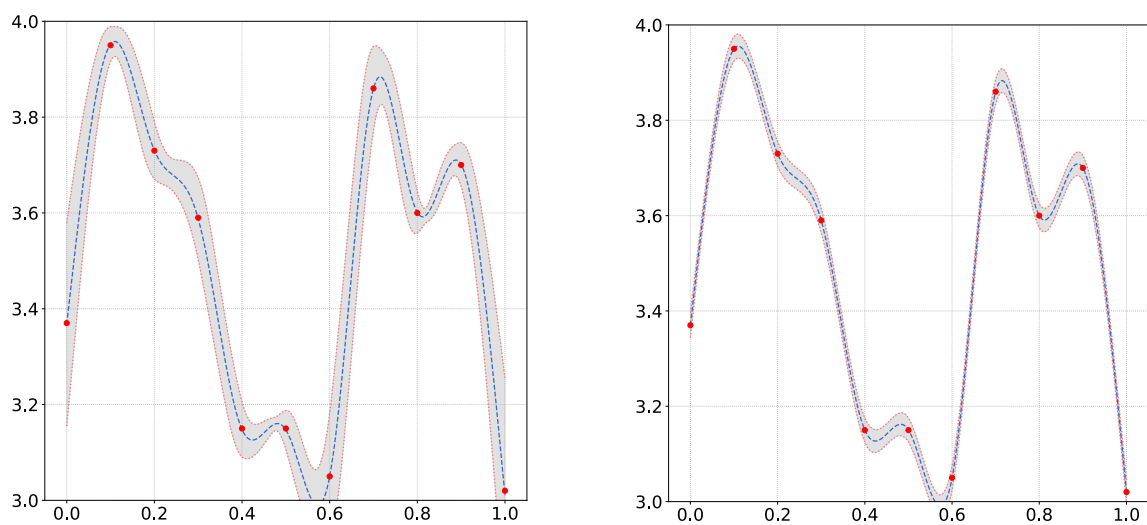


Рис. 4.2 Усредненный интерполянт (обозначен синим пунктиром) и доверительный интервал для уровня надежности 0.99 (зона окрашена серым), слева для случая погрешности в  $x$ , справа для случая погрешности в  $h(x)$ , для естественного кубического сплайна

## 5. Влияние погрешностей в случае естественного кубического сплайна

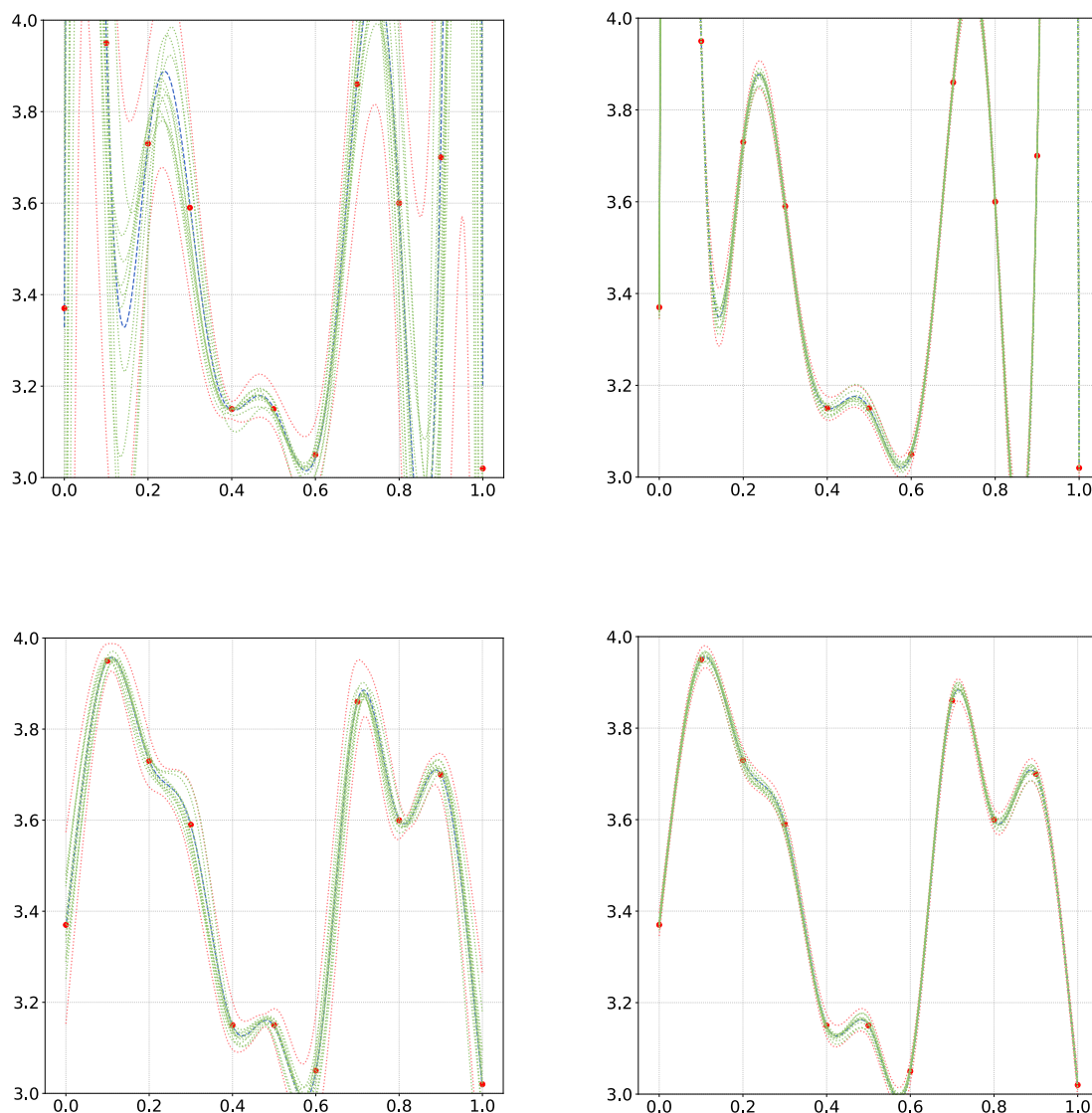
Далее я повторил вычисления уже для случая естественного кубического сплайна, и результат ожидаемо оказался более точным: на каждом интервале кусочно-заданный сплайн суть есть полином степени не выше 3, поэтому паразитных осцилляций на краях интервала, возникавших в случае интерполяции полиномом Лагранжа в следствие феномена Рунге, нет. На рис. 4.2 приведены графики, аналогичные таковым на рис. 4.1, но уже для естественного кубического сплайна. На рис. 5.1 я также поместил несколько полиномов из числа той 1000 (а именно 10).

Пару слов о программной реализации: я писал код на *python* и старался по возможности обобщить алгоритм расчётов, параметры для скриптов хранятся в папке *utils/res/cfg* в формате *json*, папка *utils/res/plots* является местом сохранения графиков. Используется векторный формат *svg* для сохранения качества при увеличении (он потрясюще выглядит на *highdpi* экранах). В реализации статистического анализа использовался пакет *scipy.stats*. У базового класса *InterpolationMethod* есть 2 наследника: *CubSplineMethod* и *LagrangeMethod*, имплементирующие соответствующие методы. Параметром в базовой части является только набор узлов, для продвинутой части необходимо также задать такие величины, как  $\sigma$ ,  $\mu$  для случайной величины  $Z$ , уровень надежности и границы интервала, шаг вычисления и число случайных векторов.

### Заключение

В ходе анализа влияния погрешностей входных данных оказалось, что случайные смещения в абсциссах узлов имеют много большее влияние на полученное приближение функции, чем случайные смещения в ординатах тех же узлов. Самыми уязвимыми к ошибкам по абсциссам участками для интерполяционного полинома Лагранжа являются края интервала применения метода (феномен Рунге). При возмущениях в ординатах самые широкие зоны доверительного интервала появляются в локальных экстремумах, притом полученные полиномы с хорошей точностью совпадают во всех узлах. В случае кубического сплайна доверительный интервал не имеет подобной дифференциации по ширине и остается сравнительно постоянным, более широкие интервалы можно наблюдать в местах изменения 1 или 2 производных (полагаю, это логично, при появлении погрешностей каждого вида смещаются нули обеих производных и, как следствие, места экстремумов и точки смены выпуклости). Я не стал включать сюда подробное описание реализованных методов, они находятся в

соответствующем каталоге и содержат (на мой взгляд) исчерпывающее количество комментариев, облегчающих понимание происходящего.



*Рис. 5.1 (верхний ряд) результаты интерполяции с рис. 4.1, но дополнительно приведены 10 из 1000 случайных интерполянтов, (нижний ряд) аналогично для рис.4.2*

## Список использованных источников

1. **Соколов А. П., Першин А. Ю.** Инструкция по выполнению лабораторных работ (общая). // кафедра «Системы автоматизированного проектирования» МГТУ им. Н. Э. Баумана, Москва, 2021.
2. **Першин А. Ю.** Лекции по вычислительной математике. // Кафедра РК6 (Системы автоматизированного проектирования) МГТУ им. Н. Э. Баумана, 2020.
3. **Higham, Nicholas J.** Accuracy, and stability of numerical algorithms // University of Manchester, Manchester, England, 2002.