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: | Пролыгина Алина Максимовна |
| Группа: | РК6-51Б |
| Тип задания: | лабораторная работа |
| Тема: | Интерполяция кубическими сплайнами |

Студент ______ подпись, дата — Пролыгина А.М. (Фамилия, И.О.)

Преподаватель ______ подпись, дата — Соколов А.П. (Фамилия, И.О.)

Москва, 2021

# Содержание

# Интерполяция кубическими сплайнами

## 1 Задание

Требуется (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)

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $x_i$ | 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 | 3.15 | 3.05 | 3.86 | 3.60 | 3.70 | 3.02 |

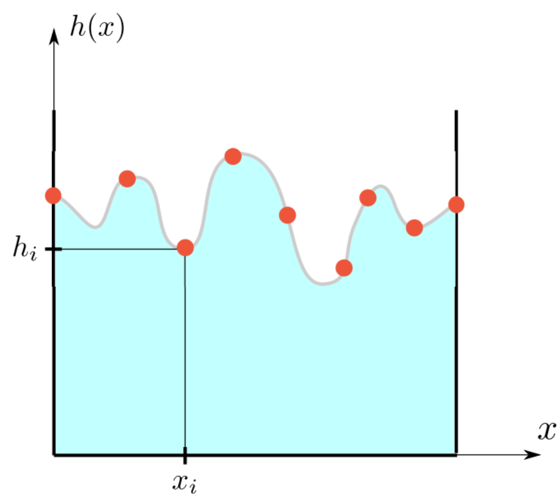


Рис. 1. Поверхность вязкой жидкости (серая кривая), движущейся сквозь некоторую среду (например, пористую). Её значения известны только в нескольких точках (красные узлы).

Требуется (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:

   (a) Сгенерировать 1000 векторов значений $[\tilde{x}_0, ..., \tilde{x}_{10}]^T$, 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 $\tilde{h}_l(x)$ 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 $x$ 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?

## 2 Цель выполнения лабораторной работы

Цель выполнения лабораторной работы – познакомиться с интерполяцией в целом (базовая часть) и проанализировать, как неопределенности влияют на ее предсказания (продвинутая часть).

## 3 Выполненные задачи

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

2. Написаны 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, построена аппроксимация зависимости уровня поверхности жидкости $h(x)$ от координаты $x$ (см. рисунок 1) с помощью кубического сплайна и продемонстрирована на графике вместе с исходными узлами.

4. Разработана функция *l_i(i, x, x_nodes)*, которая возвращает значение $i$-го базисного полинома Лагранжа, заданного на узлах с абсциссами *x_nodes*, в точке $x$.

5. Написана функция *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 $x$.

6. Проведен анализ, позволяющий выявить влияние погрешности, которая возникает при измерении координаты $x_i$, на интерполяцию:

   (a) Сгенерировано 1000 векторов значений $[\tilde{x}_0, ..., \tilde{x}_{10}]^T$, 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 построен интерполянт Лагранжа, предполагая, что в качестве абсцисс узлов используются значения $\tilde{x}_i$, а ординат – $h_i$ из таблицы 1.

   (c) Предполагая, что все интерполянты представляют собой равновероятные события, построены 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 $x$ будет лежать в интервале $[\tilde{h}_l(x); \tilde{h}_u(x)]$ равна 0.9.

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

7. Проведен 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.

8. Проведен анализ, описанный в двух предыдущих пунктах, для случая интерполяции кубическим сплайном.

# 4 Базовая часть

## 4.1 Вычисление коэффициентов естественного кубического сплайна

Пусть функция $f(x)$ задана в $n$ интерполяционных узлах $a = x_1, x_2, ..., x_n = b$ на отрезке $[a; b]$. Тогда кубическим сплайном для функции $f(x)$ называется функция $S(x)$, для которой верно:

1. $S(x)$ кусочно задана кубическими многочленами $S_i(x)$ на каждом отрезке $[x_i; x_{i+1}]$, $i = 1, ..., n-1$;

2. $S_i(x_i) = f(x_i)$ и $S_i(x_{i+1}) = f(x_{i+1})$, $i = 1, ..., n-1$;

3. значения смежных многочленов совпадают в общих узлах: $S_i(x_{i+1}) = S_{i+1}(x_{i+1})$, $i = 1, ..., n-2$;

4. значения первых производных смежных многочленов совпадают в общих узлах: $S'_i(x_{i+1}) = S'_{i+1}(x_{i+1})$, $i = 1, ..., n-2$;

5. значения вторых производных смежных многочленов совпадают в общих узлах: $S''_i(x_{i+1}) = S''_{i+1}(x_{i+1}$, $i = 1, ..., n-2$;

6. заданы граничные условия (в данной задаче используются естественные граничные условия): $S''(x_1) = S''(x_n) = 0$

Форма кубического многочлена $S_i(x)$ на отрезке $[x_i, x_{i+1}]$:

$$S_i(x) = a_i + b_i \cdot (x - x_i) + c_i \cdot (x - x_i)^2 + d_i \cdot (x - x_i)^3, \tag{1}$$

где $a_i, b_i, c_i, d_i$ называются коэффициентами кубического сплайна. Коэффициенты вычисляются с помощью решения СЛАУ $A \cdot c = B$:

$$\begin{pmatrix} 1 & 0 & \dots & \dots & \dots & 0 \\ q_0 & 2 \cdot (q_1 + q_0) & q_1 & 0 & \dots & 0 \\ 0 & q_1 & 2 \cdot (q_2 + q_1) & q_2 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & q_{n-3} & 2 \cdot (q_{n-2} + q_{n-3}) & q_{n-2} \\ 0 & \dots & \dots & \dots & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{n-2} \\ c_{n-1} \end{pmatrix} =$$

$$= \begin{pmatrix} 0 \\ \frac{3}{q_1} \cdot (a_2 - a_1) - \frac{3}{q_0} \cdot (a_1 - a_0) \\ \frac{3}{q_2} \cdot (a_3 - a_2) - \frac{3}{q_1} \cdot (a_2 - a_1) \\ \vdots \\ \frac{3}{q_{n-2}} \cdot (a_{n-1} - a_{n-2}) - \frac{3}{q_{n-3}} \cdot (a_{n-2} - a_{n-3}) \\ 0 \end{pmatrix} \tag{2}$$

где $q_i = x_{i+1} - x_i$, при $i = 0, ..., n - 2$.

Из уравнения (2) находится вектор коэффициентов $c$, остальные коэффициенты находятся по формулам (3), (4) и (5).

$$a_i = f(x_i), \tag{3}$$

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

$$d_i = \frac{c_{i+1} - c_i}{3 \cdot q_i}; \tag{5}$$

Алгоритм функции *qubic_spline_coeff(x_nodes, y_nodes)*:

1. Создаются три списка с главной и смежной с ними диагоналями. Из них формируется итоговая матрица $A$ (Листинг 1).

2. В цикле заполняется матрица $B$.

3. С помощью функции np.linalg.inv() матрица  транспонируется для возможности решения СЛАУ (2) с последующим нахождением вектора $c$.

4. По формулам (4) и (5) находятся остальные коэффицинты и объединяются в искомую матрицу *qs_coeff* (Листинг 2).

Функция *qubic_spline_coeff(x_nodes, y_nodes)*, разработанная для нахождения коэффициентов кубического сплайна, представлена фрагментами в листингах 1 и 2 на языке Python.

Листинг 1. Заполнение матрицы $A$ по формуле (2)

```
for i in range(N-2):
    K = 2*(x_nodes[i+2] - x_nodes[i])
    K0.append(K)
K0.append(1)

for i in range(N-2):
    K = x_nodes[i+1] - x_nodes[i]
    K_1.append(K)
K_1.append(0)

for i in range(N-2):
    K = x_nodes[i+2] - x_nodes[i+1]
    K1.append(K)

A = np.diag(K0, 0)
A1 = np.diag(K1, 1)
A_1 = np.diag(K_1, -1)
A = A + A1 + A_1
```

Листинг 2. Создание матрицы коэффициентов *qs_coeff*

```
res_c = np.dot(A_T, B)
res_b = np.zeros((N))
res_d = np.zeros((N))

for i in range(N-1):
    res_b[i] = 1/(x_nodes[i+1]-x_nodes[i])*(y_nodes[i+1]-y_nodes[i]) -
        (x_nodes[i+1]-x_nodes[i])/3*(res_c[i+1]+2*res_c[i])
    res_d[i] = (res_c[i+1]-res_c[i])/(3*(x_nodes[i+1]-x_nodes[i]))

qs_coeff = [res_b, res_c, res_d]

return qs_coeff
```

## 4.2 Построение сплайна и его производной

В пункте 4.1 были определены коэффициенты естественного кубического сплайна. Для вычисления значения сплайна $S_i(x)$ в точке $x$ необходимо подставить значения коэффициентов $b_i$, $c_i$, $d_i$ в выражение (1). Для построения достаточно плавного графика взяты 100 точек, которые приближаются к исходным узлам *x_ nodes* с помощью функции *approx(x, x_ nodes)*, представленной в листинге 3. Функция *qubic_ spline(x, qs_ coeff, x_ nodes, y_ nodes)*, разработанная для вычисления значения сплайна $S_i(x)$ в точке $x$, отображена в листинге 4 на языке Python.

Листинг 3. Реализация функции *approx(x, x_ nodes)*

```
def approx(x, x_nodes):
i = 0
N = len(x_nodes)

while((x > x_nodes[i]) and i < 10):

    i = i + 1

if(x <= x_nodes[0]):
    i = 1

if(x >= x_nodes[N-1]):
    i = N-1
return i - 1
```

Листинг 4. Реализация функции *qubic_ spline(x, qs_ coeff, x_ nodes, y_ nodes)*

```
def qubic_spline(x, qs_coeff, x_nodes, y_nodes): # функциясплайна

i = approx(x, x_nodes)
S = y_nodes[i] + qs_coeff[0][i]*(x-x_nodes[i]) + qs_coeff[1][i]*(x-x_nodes[i])**2 +
    qs_coeff[2][i]*(x - x_nodes[i])**3

return S
```

Аналогично, для вычисления значения производной сплайна $S_i^{'}(x)$ в точке $x$ коэффициенты $b_i$, $c_i$, $d_i$ подставляются в выражение (6):

$$S_i^{'}(x) = b_i + 2c_i \cdot (x - x_i) + 3d_i \cdot (x - x_i)^2. \tag{6}$$

Функция вычисления производной сплайна *d_ qubic_ spline(x, qs_ coeff, x_ nodes)* представлена в листинге 5.

Листинг 5. Реализация функции *d_qubic_spline(x, qs_coeff, x_nodes)*

```
def d_qubic_spline(x, qs_coeff, x_nodes): # функциявычисленияпроизводнойсплайна

i = approx(x, x_nodes)
S_ = qs_coeff[0][i] + 2*qs_coeff[1][i]*(x−x_nodes[i]) + 3*qs_coeff[2][i]*(x−x_nodes[i])**2

return S_

return S
```

### 4.3 Построение графика кубического сплайна и его дифференциала

Для построения приближения зависимости уровня поверхности жидкости $h(x)$ от координаты $x$ по точкам из таблицы 1 использован метод интерполяции кубическими сплайнами. С использованием бибилиотеки matplotlib был построен график составного кубического сплайна $S(x)$ и его производной $S'(x)$. При построении использовался шаг $h$=0.01.(Рис. 2 и 3)

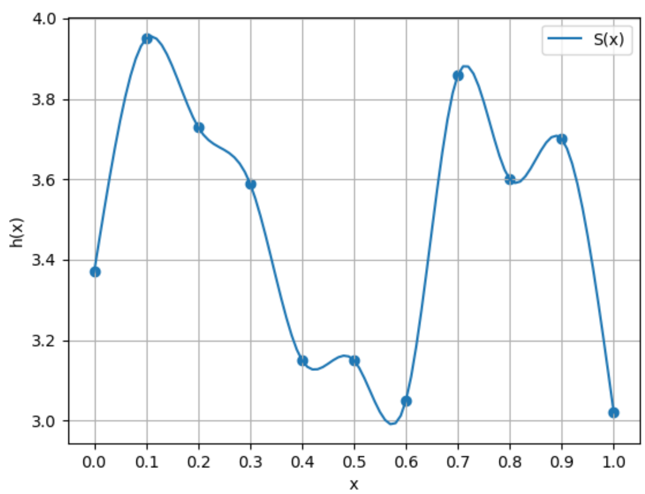


Рис. 2. Аппроксимация зависимости уровня жидкости $h(x)$ от координаты $x$ с помощью естественного кубического сплайна вместе с исходными узлами.

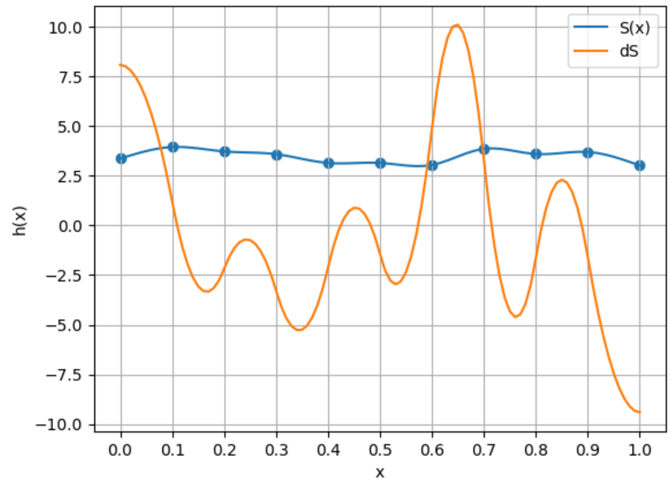


Рис. 3. График кубического сплайна и его производной

# 5 Продвинутая часть

## 5.1 Вычисление базисного полинома Лагранжа

Функция *l_i(i, x, x_nodes)*, программная реализация которой представлена в листинге 6, вычисляет значение $i$-го базисного полинома Лагранжа по формуле (7):

$$l_i(x) = \prod_{i \neq j} \frac{x - x_j}{x_i - x_j}; \tag{7}$$

где $j = 0, ..., n - 1$.

Листинг 6. Реализация функции *l_i(i, x, x_nodes)*

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

## 5.2 Вычисление интерполяционного полинома Лагранжа

Функция *L(x, x_nodes, y_nodes)*, представленная в листинге 7, находит значение интерполяционного полинома Лагранжа по формуле (8):

$$L_{n-1}(x) = \sum_{i=0}^{n-1} f(x_i) \cdot l_i(x); \tag{8}$$

где $l_i(x)$ - $i$-тый базисный полином Лагранжа, который вычисляется по формуле (7).

Листинг 7. Реализация функции *L(x, x_nodes, y_nodes)*

```
def L(x, x_nodes, y_nodes): # вычислениеинтерполяционногополиномаЛагранжа
Lx = 0
for i in range(N):
    Lx = Lx + y_nodes[i]*(l_i(i, x, x_nodes))
return Lx
```

### 5.3 Влияние погрешности величины $\mathrm{x_i}$ на интерполяцию Лагранжа

Для генерации 1000 векторов $\tilde{x}_i$ необходимо нормально распределить с нулевым математическим ожиданием и стандартным отклонением $10^{-2}$ величину $Z$, т.к. $\tilde{x}_i = x_i + Z$. Задание величины $Z$ с последующим определением $\tilde{x}_i$ представлено в листинге 8.

Листинг 8. Вычисление $\tilde{x}$

```
for i in range(1000): # заполнениемассивапогрешностей
ZZ = []
for j in range(N):
    ZZ.append(np.random.normal(0, 0.01))
Z.append(ZZ)

for i in range(1000): # заполнениемассиваиксовспогрешностью
Xtt = []
for j in range(N):
    Xt_ = x_nodes[j] + Z[i][j]
    Xtt.append(Xt_)
Xt.append(Xtt)
```

Передавая $\tilde{x}_i$ вместо *x_nodes* в функцию *L(x, x_nodes, y_nodes)*, строим график 1000 интерполянтов Лагранжа (Рис. 4).

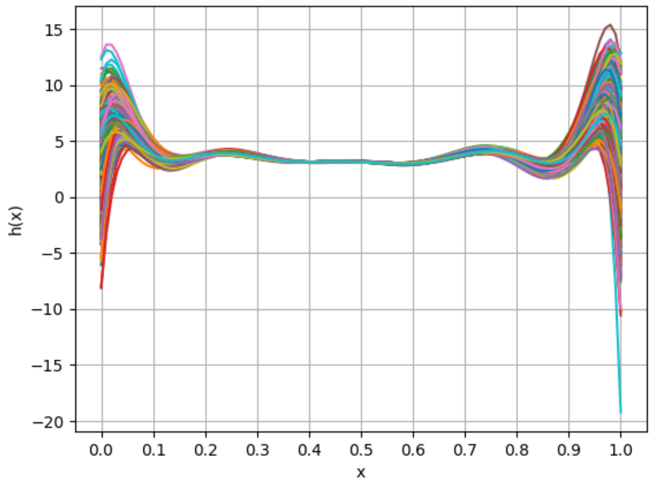


Рис. 4. 1000 различных интерполянтов Лагранжа на основе значений $\tilde{x_i}$.

Так как всего 1000 интерполянтов Лагранжа (номера элементов массива интерполянтов от 0 до 999), то, чтобы вероятность того, что значение интерполянта в точке $x$ принадлежало интервалу $[\tilde{h}_l(x);\ \tilde{h}_u(x)]$, равнялась 0.9, $\tilde{h}_l(x)$ должен быть выше, чем 5% всех сплайнов, а $\tilde{h}_u(x)$, наоборот, ниже чем 5% всех сплайнов. Следовательно, $\tilde{h}_l(x)$ – это 49-й элемент отсортированного массива интерполянтов Лагранжа, а $\tilde{h}_u(x)$ – 949-й элемент этого массива ($\tilde{h}_l(x)$ - верхняя граница доверительной полосы, а $\tilde{h}_u(x)$- нижняя граница доверительной полосы). Средний элемент соответствует 499-тому элементу. Сортировка осуществляется с помощью функции np.sort().(Рис. 5)

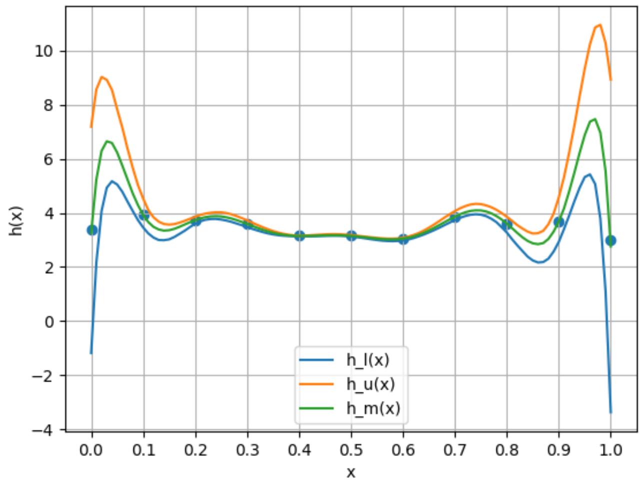


Рис. 5. График функций $\tilde{h}_l(x)$, $\tilde{h}_u(x)$ и усредненного интерполянта с выводом узлов на основе значений $\tilde{x}_i$.

На графике (Рис. 5) видно, что погрешность интерполяции с ростом $j$ до определенного момента убывает (примерно до $j = 50$), затем начинает возрастать. Это связано с ошибками округления, которые при большом числе равномерно распределенных узлов оказывают все большее негативное влияние на результат. Значит, наиболее чувствительными к погрешностям участками интерполянта являются граничные участки (начальные и конечные), т. к. на них накапливаются паразитные осцилляции. При большом числе узлов следует использовать другие наборы узлов интерполяции, являющиеся нулями ортогональных полиномов, например, полиномов Чебышева.

### 5.4 Влияние погрешности величины $h_i$ на интерполяцию Лагранжа

Повторен анализ из предыдущего пункта, предполагая, что $\tilde{h}_i = h_i + Z$, где $h_i$ соответствует значению в таблице 1, а $Z$ является случайной величиной с нормальным распределением с нулевым математическим ожиданием и стандартным отклонением $10^{-2}$. (Рис. 6 и 7)

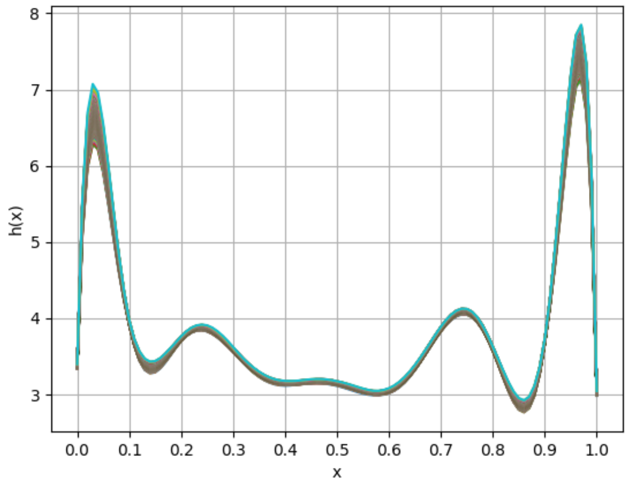


Рис. 6. 1000 различных интерполянтов Лагранжа на основе значений $\tilde{h_i}$.

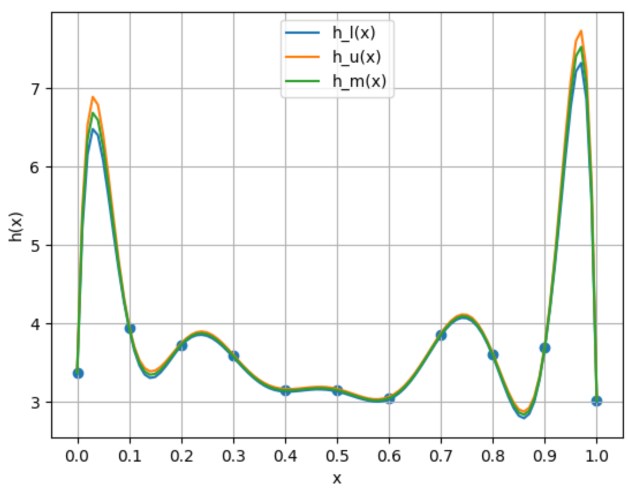


Рис. 7. График функций $\tilde{h_l}(x)$, $\tilde{h_u}(x)$ и усредненного интерполянта с выводом узлов на основе значений $\tilde{h_i}$.

На графике (Рис. 7) также присутствуют паразитные осцилляции на граничных участках.

### 5.5 Влияние погрешности величин $x_i$ и $h_i$ на интерполяцию кубическим сплайном

Повторен анализ из двух предыдущих пунктов для интерполяции кубическим сплайном, используя функции из базовой части задачи.

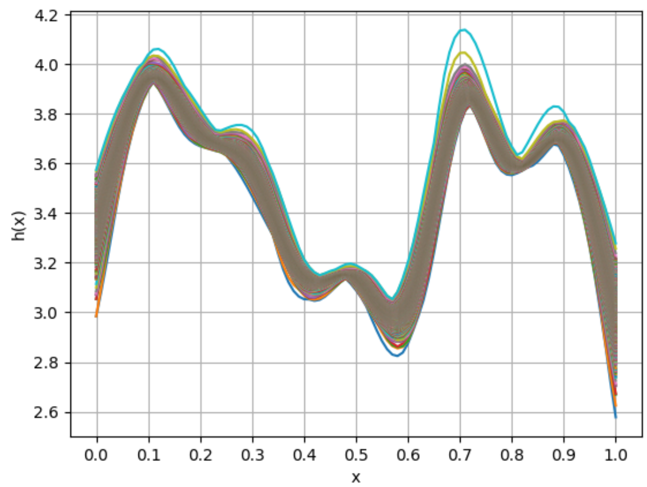


Рис. 8. 1000 различных кубических сплайнов на основе значений $\tilde{x}_i$.

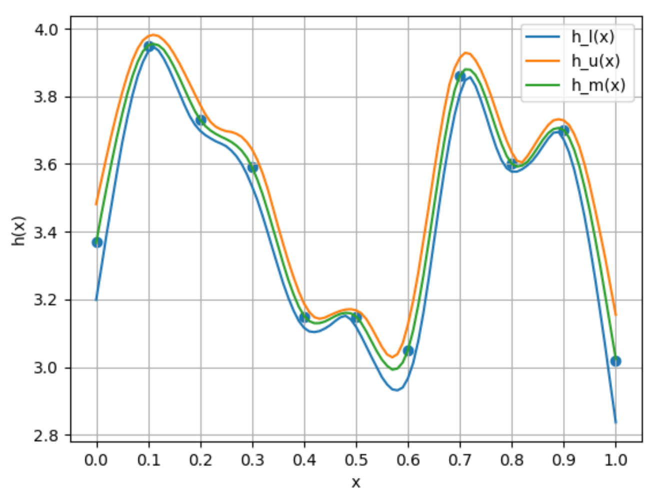


Рис. 9. График функций $\tilde{h_l}(x)$, $\tilde{h_u}(x)$ и усредненного интерполянта с выводом узлов на основе значений $\tilde{x}_i$.

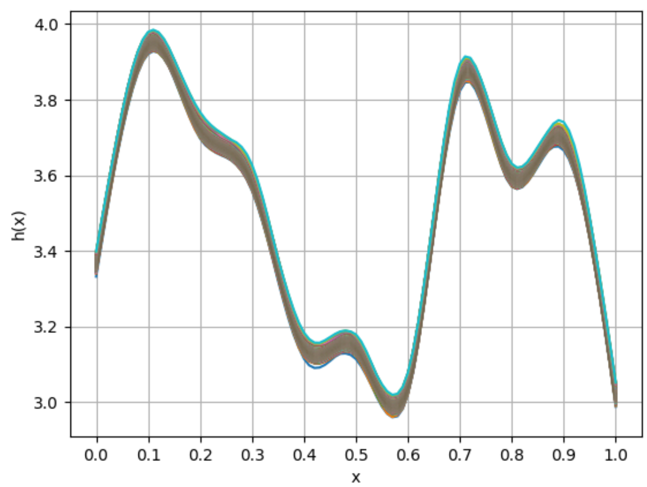


Рис. 10. 1000 различных кубических сплайнов на основе значений $\tilde{h_i}$.

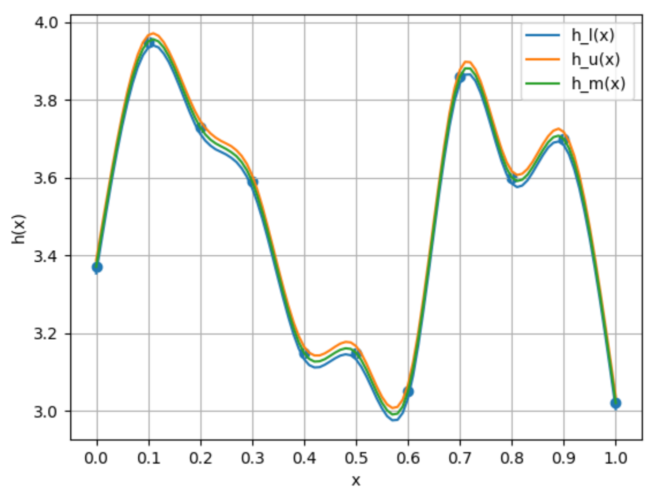


Рис. 11. График функций $\tilde{h_l}(x)$, $\tilde{h_u}(x)$ и усредненного интерполянта с выводом узлов на основе значений $\tilde{h_i}$.

На графиках (Рис. 9 и Рис. 11) видно, что сплайны не накапливают паразитные осцилляции, так как используются сплайны небольших порядков.

## 6 Заключение

В результате выполнения лабораторной работы были созданы функции для интерполяции кубическим сплайном и интерполяции Лагранжа. С помощью реа-лизации доверительных и усредненных интервалов стало возможным сравнение влияния погрешностей на интерполяцию этими методами. В итоге, выяснено, что паразитные осцилляции возникают при интерполяции Лагранжа, метод кубического сплайна позволяет избежать этого, т. к. в этой задаче вычисляются сплайны достаточно небольших порядков.

### Список использованных источников


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

2. Тимербаев М.Р. Приближение функций. Численное интегрирование. Казанский федеральный университет, Казань, 2015, С. 92.


### Выходные данные


*Пролыгина А.М.. Отчет о выполнении лабораторной работы по дисциплине «Вычислительная математика». [Электронный ресурс] — Москва: 2021. — 18 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 РК6-51Б, Пролыгина А.М.*

*2021, осенний семестр*