Министерство образования Республики Беларусь
Белорусский государственный университет
Факультет прикладной математики и информатики

Лабораторная работа 1

Приближение функции

Гончаренко Андрей Дмитриевич
Студент 3 курса, 13 группа
Преподаватель:
Будник Анатолий Михайлович

Минск, 2022

Постановка задачи. Часть 1

Рассмотрим на отрезке $[a, b]$ набор различных точек

$$x_0, \ldots, x_n : x_0 < x_1 < \ldots < x_{n-1} < x_n.$$

В этих точках заданы значения $f(x) : f_i = f(x_i), i = \overline{0, n}$. Требуется восстановить значение функции $f(x)$ в трех точках: x^*, x^{**}, x^{***}, точки принадлежат отрезку $[a, b]$.

Для построения таблицы $(x_i; f(x_i))$ дано:

1. $[a; b] = [\alpha_j; 1 + \alpha_j], \alpha_j = 0.1 + 0.05j, j = 6;$
2. $x_i = \alpha_j + ih, i = \overline{0, n}, n = 10, h = \frac{1}{n};$
3. $f(x) = \alpha_j e^x + (1 - \alpha_j) sin(x).$

Точки восстановления:

$$x^* = x_0 + \frac{2}{3}h,\ x^{**} = x_{\frac{n}{2}} + \frac{1}{2}h,\ x^{***} = x_n - \frac{1}{3}h$$

Построить интерполяционный многочлен Лагранжа. Вычислить приближенное значение в x^*, x^{**}, x^{***} Оценить погрешность $R_n(x)$ по формуле Ньютона.

Алгоритм решения задачи

1. Составить таблицу $(x_i, f(x_i))$ по заданным формулам.
2. Вычислить значения x^*, x^{**}, x^{***} и значения функции в этих точках.
3. Сделать оценку точности, для этого найти погрешность по форму Ньютона, так же найти истинную погрешность.
   - Для оценки погрешности по формуле требуется построить базовую таблицу разделенных разностей, а на ее основе построить таблицы с учетом x^*, x^{**}, x^{***}. Для нахождения погрешности нам достаточно последнего значения элемента первой строки. Далее домножаем на $\omega(x) \rightarrow$ получаем погрешность.
   - Для оценки истинной погрешности нам нужно найти разность между результатом подстановки x^*, x^{**}, x^{***} в исходную формулу и в многочлен Лагранжа.
4. Сделать вывод.

Таблица 1: $(x_i, f(x_i))$:

x_i	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3	1.4
$f(x_i)$	0.830	0.947	1.068	1.192	1.321	1.454	1.592	1.736	1.887	2.046	2.213

Таблица 2: Исходная таблица разделенных разностей (значения округлены):

0.83	1.16	0.18	0.03	0.05	0.01	5.86e-04	8.16e-05	3.27e-05	3.67e-06	1.43e-07
0.94	1.21	0.19	0.04	0.05	0.01	6.43e-04	1.07e-04	3.61e-05	3.81e-06	0
1.07	1.24	0.21	0.06	0.06	0.01	7.18e-04	1.36e-04	3.95e-05	0	0
1.19	1.29	0.23	0.09	0.06	0.01	8.14e-04	1.68e-04	0	0	0
1.32	1.33	0.26	0.11	0.06	0.01	9.32e-04	0	0	0	0
1.45	1.38	0.29	0.14	0.07	0.01	0	0	0	0	0
1.59	1.44	0.33	0.17	0.08	0	0	0	0	0	0
1.73	1.51	0.39	0.20	0	0	0	0	0	0	0
1.88	1.59	0.45	0	0	0	0	0	0	0	0
2.05	1.67	0	0	0	0	0	0	0	0	0
2.21	0	0	0	0	0	0	0	0	0	0

Точки восстановления и значения функции в этих точках:

- $x^* = 0.4666666666666667, f(x^*) = 0.9078150316251946$;
- $x^{**} = 0.9500000000000001, f(x^{**}) = 1.5223331665999482$;
- $x^{***} = 1.3666666666666665, f(x^{***}) = 2.156444551203661$.

Оценка точности:

1. $R_n(x)$ будем оценивать по формуле Ньютона. Для этого воспользуемся следующей формулой:

   $$R_n(x) = \omega(x) \cdot f(x, x_0, \ldots, x_n), \omega_{n+1}(x) = (x - x_0) \cdot \ldots \cdot (x - x_n).$$

   Для нахождения $f(x, x_0, \ldots, x_n)$ была построена новая таблица разделенных разностей, на основе исходной. В новой таблице мы получили новую диагональ для $x = \{x^*, x^{**}, x^{***}\}$ и нам нужно взять последнее значение первой строки, это и будет значение $f(x, x_0, \ldots, x_n)$. Подставляем значения x^*, x^{**}, x^{***} в $\omega(x)$ и получаем следующие результаты:

   - $x^* = 0.4666666666666667, R_n(x^*) = 2.694453941530423 \cdot 10^{-14}$;

- $x^{**} = 0.9500000000000001, R_n(x^{**}) = 7.71925111597307 \cdot 10^{-16}$;
- $x^{***} = 1.3666666666666665, R_n(x^{***}) = 6.966752827239019 \cdot 10^{-14}$.

2. Оценим истинную погрешность по формуле:

$$|r| = |f(x) - L(x)|.$$

Подставим точки x^*, x^{**}, x^{***} в исходную функцию и отнимем теоретический результат интерполирования, получим следующие результаты:

- $x^* = 0.4666666666666667, |r| = 2.6756374893466273 \cdot 10^{-14}$;
- $x^{**} = 0.9500000000000001, |r| = 8.881784197001252 \cdot 10^{-16}$;
- $x^{***} = 1.3666666666666665, |r| = 7.105427357601002 \cdot 10^{-14}$.

Вывод.

1. Почему в разных точках разная погрешность? Погрершность в точке, в первую очередь зависит от удаления этой точик от узла, в которм проводится интерполирование. В самой точке погрешность равна 0, чем дальше мы удаляемся от точки, тем больше погрешность.

2. Почему теоретическая оценка и истинная разные? Они разные, потому что теоретическая оценка - оценка сверху, т.е. мы береме максимальные значения, следовательно, истинная погрешность меньше.

Постановка задачи. Часть 2

Необходимо построить интерполяционный многочлен Ньютона на равномерной сетке узлов, для вычислений будем брать $k = 3$, т.е. будем брать 4 точки x_0, x_1, x_2, x_3 (для начала таблицы). Требуется найти приближение значения функции в точке x^*, оценить погрешность. Для построение таблицы:

1. $f(x_i) = f_i$;
2. $[a, b] = [\alpha_j, 1 + \alpha_j], \alpha_j = 0.1 + 0.05j, j = 6$;
3. $x_i = \alpha_j + ih, i = \overline{0, n}, n = 10, h = \frac{1}{n}$;
4. $f(x) = \alpha_j e^x + (1 - \alpha_j) sin(x)$.

Точка для исследования: $x^* = x_0 + \frac{2}{3}h$

Алгоритм решения задачи

Построим таблицу конечных разностей

f_0	Δf_0	$\Delta^2 f_0$	$\dots$	$\Delta^n f_0$
f_1	Δf_1	$\Delta^2 f_1$	$\dots$	-
f_2	Δf_2	$\Delta^2 f_2$	$\dots$	-
$\cdot$	$\cdot$	$\cdot$	$\dots$	$\cdot$
$\cdot$	$\cdot$	$\cdot$	$\dots$	$\cdot$
$\cdot$	$\cdot$	$\Delta^2 f_{n-2}$	$\dots$	-
$\cdot$	Δf_{n-1}	-	$\dots$	-
f_n	-	-	$\dots$	-

где $\Delta^k f_i = \Delta^{k-1}(f_{i+1} - f_i)$.

После программируем формулу Ньютона для начала таблицы

$$P_n(x) = f_0 + \frac{t}{1!}\Delta f_0 + \frac{t(t-1)}{2!}\Delta^2 f_0 + \ldots + \frac{t(t-1)\ldots(t-n+1)}{n!}\Delta^n f_0,$$

где $t = \frac{x-x_0}{h}$.

Для нахождения погрешности воспользуемся формулой

$$R_n(x) = h^{n+1}\frac{t(t-1)\ldots(t-n)}{(n+1)!}f^{(n+1)}(\xi), \xi \in [x_0, x_0 + kh],$$

где значение $f^{(n+1)}(\xi)$ будем оценивать аналитически.

Таблица 3: Таблица конечных разностей:

8.30380884e-01	1.16762947e-01	3.72622575e-03	1.83292637e-04
9.47143831e-01	1.20489173e-01	3.90951839e-03	-
1.06763300	1.24398691e-01	-	-
1.19203170	-	-	-

Из получившейся таблицы нам нужна первая строка. Подставляем значения из таблицы в формулу Ньютона, получаем 0.9078178755138939.

Для оценки погрешности сначала нужно аналитически оценить $f^{(n+1)}(\xi)$. После построения графика (Приложение 2, Рис. 7), на нашем отрезке можно увидеть, что функция возрастающая, следовательно, масимум функции достигается в максимальном $x = 0.7$. Получаем, что значение функции в этой точке равно 1.1920316953308. Подставляем получившееся значение в формулу. Получаем $R_n(x) = 3.4338361593891376 \cdot 10^{-6}$.

Вывод. Для теоертической оценки погрешности была взята оценка сверху, путем взятия максимального значения $f^{(n+1)}(\xi)$.

Постановка задачи. Часть 3

Построить интерполяционный многочлен Лагранжа для $n = 10$, выбрав в качестве узлов корни многочлена Чебышева. Найти приближеные значения функции в точках x^*, x^{**}, x^{***}. Оценить погрешность.

1. $f(x_i) = f_i$;
2. $[a, b] = [\alpha_j, 1 + \alpha_j], \alpha_j = 0.1 + 0.05j, j = 6$;
3. $t_k \in [-1; 1], k = \overline{0, n}, n = 10$;
4. $f(x) = \alpha_j e^x + (1 - \alpha_j) sin(x)$.

Точки для исследования:

$$x^* = x_0 + \frac{2}{3}h,\ x^{**} = x_{\frac{n}{2}} + \frac{1}{2}h,\ x^{***} = x_n - \frac{1}{3}h.$$

Алгоритм решения задачи

1. Находим корни многочлена Чебышева:

$$t_k = cos\left(\frac{2k+1}{2n+2}\pi\right), n = 10, k = \overline{0, n}, t_k \in [-1; 1]$$

2. Отображаем полученные значения на $[a; b]$:

$$x_k = \frac{a+b}{2} + \frac{b-a}{2}t_k, n = 10, k = \overline{0, n}, x_k \in [a; b]$$

3. Найти погрешность по формуле Ньютона и истинную погрешность, используя формулы из первой части. Найти погрешность по формуле:

$$|R_n(x)| \leq \frac{M}{(n+1)!} \cdot \frac{(b-a)^{n+1}}{2^{2n+1}}$$

Таблица 4: $(t_k, x_k, f(x_k))$:

t_k	0.988	0.901	0.733	0.5	0.223	-0.075	-0.365	-0.623	-0.826	-0.956	-1.0
x_k	1.394	1.351	1.267	1.15	1.011	0.863	0.717	0.588	0.487	0.422	0.399
$f(x_k)$	2.204	2.129	1.992	1.811	1.608	1.403	1.214	1.053	0.932	0.856	0.830

После того, как мы получили значения $t_k, x_k, f(x_k)$, воспользуемся значениями x^*, x^{**}, x^{***} из части 1, подставим полученные данные в метод для подсчета интерполяционного многочлена Лагранжа и получим следующие результаты:

- $x^* = 0.4666666666666667, f(x^*) = 0.907815031624929$;
- $x^{**} = 0.9500000000000001, f(x^{**}) = 1.5223331665999726$;
- $x^{***} = 1.3666666666666665, f(x^{***}) = 2.1564445512212136$.

Оценка точности:

1. $R_n(x)$ будем оценивать по формуле Ньютона. Для этого воспользуемся следующей формулой:

   $$R_n(x) = \omega(x) \cdot f(x, x_0, \ldots, x_n), \omega_{n+1}(x) = (x - x_0) \cdot \ldots \cdot (x - x_n).$$

   Более подробно шаги были расписаны в части 1. Получили следующие результаты:

   - $x^* = 0.4666666666666667, R_n(x^*) = 2.490086139770498 \cdot 10^{-15}$;
   - $x^{**} = 0.9500000000000001, R_n(x^{**}) = 9.948068350873343 \cdot 10^{-15}$;
   - $x^{***} = 1.3666666666666665, R_n(x^{***}) = 1.1566508608581059 \cdot 10^{-14}$.

2. Оценим истинную погрешность по формуле:

   $$|r| = |f(x) - L(x)|.$$

   Более подробно шаги были расписаны в части 1. Получили следующие результаты:

   - $x^* = 0.4666666666666667, |r| = 2.9976021664879227 \cdot 10^{-15}$;

- $x^{**} = 0.9500000000000001, |r| = 9.769962616701378 \cdot 10^{-15}$;
- $x^{***} = 1.3666666666666665, |r| = 1.2434497875801753 \cdot 10^{-14}$.

3. Оценим формульную погрешность. Значение M оценим аналитически с построением графика. Из Приложения 3 Рис. 9 видно, что график возрастающий, следовательно максимальное значение производной будет при максимальном $x = 0.7$, значение функции в этой точке равно 0.3465957706175. Подставим значения в формулу и получим следующее, что $|R_n(x)| \leq 4.140355497095727 \cdot 10^{-15}$.

Вывод. Используя корни Чебушева в качестве узлов интерполирования, мы умменьшили погрешность относительно первой части.

Постановка задачи. Часть 4

На заданной таблице, необходимо построить полином наилучшего квадратичного приближения 5 степени. Найти приближенное значение в точках x^*, x^{**}, x^{***}. Оценить погрешность.

1. $f(x_i) = f_i$;
2. $[a, b] = [\alpha_j, 1 + \alpha_j], \alpha_j = 0.1 + 0.05j, j = 6$;
3. $x_i = \alpha_j + ih, i = \overline{0, m}, m = 10, h = \frac{1}{m}$;
4. $f(x) = \alpha_j e^x + (1 - \alpha_j) sin(x)$.

Точки для исследования:

$$x^* = x_0 + \frac{2}{3}h,\ x^{**} = x_{\frac{m}{2}} + \frac{1}{2}h,\ x^{***} = x_m - \frac{1}{3}h.$$

Алгоритм решения задачи

1. Построим наш базис $\phi = (1, x, x^2, x^3, x^4, x^5)$.
2. Наилучшее среднеквадратичное приближение будет находить по формуле:
$$\phi(x) = \sum_{i=0}^{n} \alpha_i \phi_i(x).$$
3. Для нахождения α_i воспользуемся системой:
$$\prod_{jk} = F_j,\ \sum_{k=0}^{n} \left(\sum_{i=0}^{m} \phi_j(x_i)\phi_k(x_i) \right) \alpha_k = \sum_{i=0}^{m} \phi_j(x_i) f_i,\ j = \overline{0, n}.$$
4. Найти погрешность по формуле
$$R(x) = \left(\sum_{i=0}^{m} (\phi(x_i) - f(x_i))^2 \right)^{\frac{1}{2}}.$$
5. Найти истинную погрешность.

Таблица 5: Получившаяся матрица.

11.0	9.9	10.01	10.989	12.7589	15.39549
9.9	10.01	10.989	12.7589	15.39549	19.091501
10.01	10.989	12.7589	15.39549	19.091501	24.1559109
10.989	12.7589	15.39549	19.091501	24.1559109	31.03584869
12.7589	15.39549	19.091501	24.1559109	31.03584869	40.35755899
15.39549	19.091501	24.1559109	31.03584869	40.35755899	52.98822175

Таблица 6: Получившийся вектоп F.

16.2867166	16.17093283	17.56745736	20.27972967	24.39367481	30.19752578

Решаем получившуюся систему и получаем следующий многочлен ϕ:

$\phi = 0.399797 + 1.001607x + 0.194940x^2 - 0.0252416x^3 + 0.009732x^4 + 0.011359x^5$.

Подставляем значения x^*, x^{**}, x^{***}:

- $x^* = 0.466666666666666, \phi(x^*) = 0.90781552$;
- $x^{**} = 0.9500000000000001, \phi(x^{**}) = 1.52233343$;
- $x^{***} = 1.3666666666666665, \phi(x^{***}) = 2.15644498$.

Воспользуемся формулой из алгоритма и найдем погрешность:

$$8.450813871638088 \cdot 10^{-7}$$

Теперь найдем истинную погрешность:

- $x^* = 0.466666666666666, r(x^*) = 4.85576857 \cdot 10^{-7}$;
- $x^{**} = 0.9500000000000001, r(x^{**}) = 2.6583772 \cdot 10^{-7}$;
- $x^{***} = 1.3666666666666665, r(x^{***}) = 4.27508236 \cdot 10^{-7}$.

Вывод. Т.к. мы использовали метод среднеквадратичного приближения с построение многочлена 5 степени, то получаем меньшую погрешность вычислений, относительно второй части, где мы использовали 3 степень.

Приложение 1

```
def get_x1(x_i_list, h):
    return x_i_list[0] + (2 / 3) * h


def get_x2(x_i_list, h, n):
    return x_i_list[int(n / 2)] + 0.5 * h


def get_x3(x_i_list, h, n):
    return x_i_list[n] - (1 / 3) * h


def interpolate_lagrange_polynomial(x, x_i_list, f_i_list, size):
    lagrange_pol = 0

    for i in range(0, size):
        basics_pol = 1
        for j in range(0, size):
            if j != i:
                basics_pol *= (x - x_i_list[j]) / (x_i_list[i] - x_i_list[j])
        lagrange_pol += basics_pol * f_i_list[i]

    return lagrange_pol
```

Рис. 1: Функция интерполяционного многочлена Лагранжа и функции возвращающие точки восстановления

```
def split_difference_table(x_i_list, f_i_list):
    n = len(f_i_list)
    cf = np.zeros([n, n])
    cf[:, 0] = f_i_list

    for j in range(1, n):
        for i in range(n - j):
            cf[i][j] = (cf[i + 1][j - 1] - cf[i][j - 1]) / (x_i_list[i + j] - x_i_list[i])

    return cf
```

Рис. 2: Функция для построения таблицы разделенных разностей

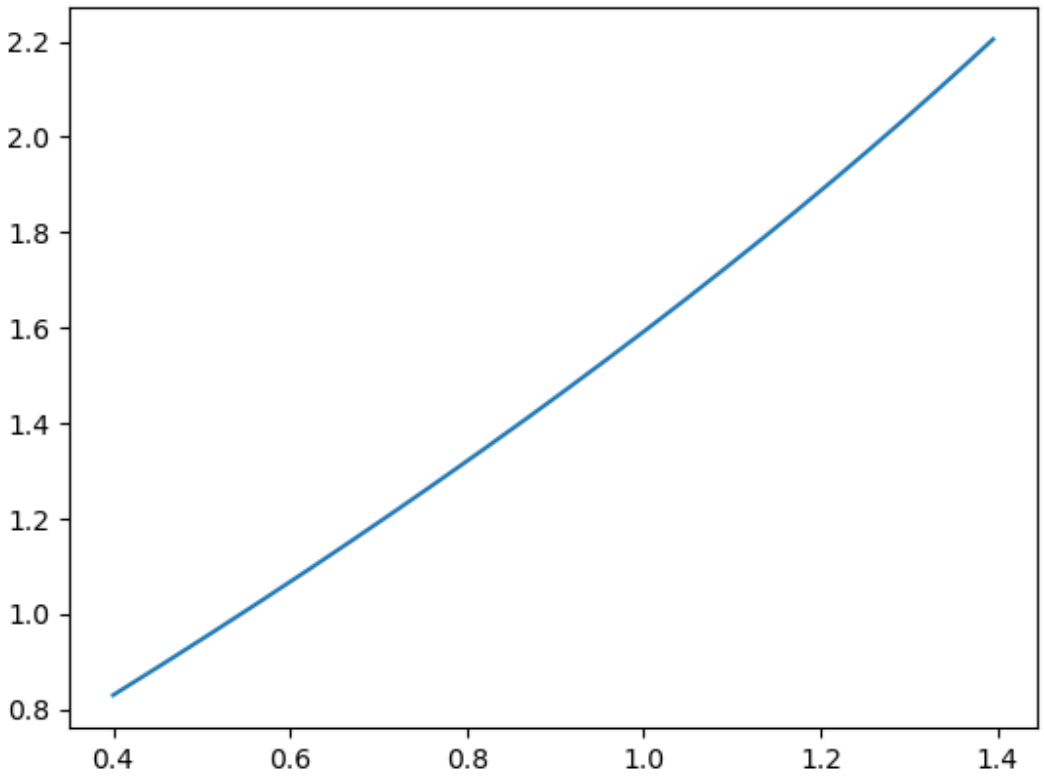

Рис. 3: График востановленной функции на отрезке $x \in [0.4, 1.4]$

Приложение 2

```
def create_finite_difference_table(n, f_i_list):
    cf = np.zeros([n, n])
    cf[:, 0] = f_i_list[:4]

    for i in range(1, n):
        for j in range(0, n - i):
            cf[j, i] = cf[j + 1, i - 1] - cf[j, i - 1]

    return cf
```

Рис. 4: Метод построения таблицы конечных разностей

```
def formula_newton(n, table, x, h, x0):
    result = 0
    t = (x - x0)/h
    for i in range(0, n):
        result += (table[0, i] / factorial(i)) * t_p(i, t)

    return result
```

Рис. 5: Метод Ньютона

```
def frac(t, n):
    result = 1
    for i in range(n+1):
        result *= t - i

    return result/factorial(n+1)


def r_n(h, n, x, x0):
    t = (x - x0) / h
    return math.pow(h, n+1) * frac(t, n) * 1.1920316953308
```

Рис. 6: Метод для нахождения погрешности

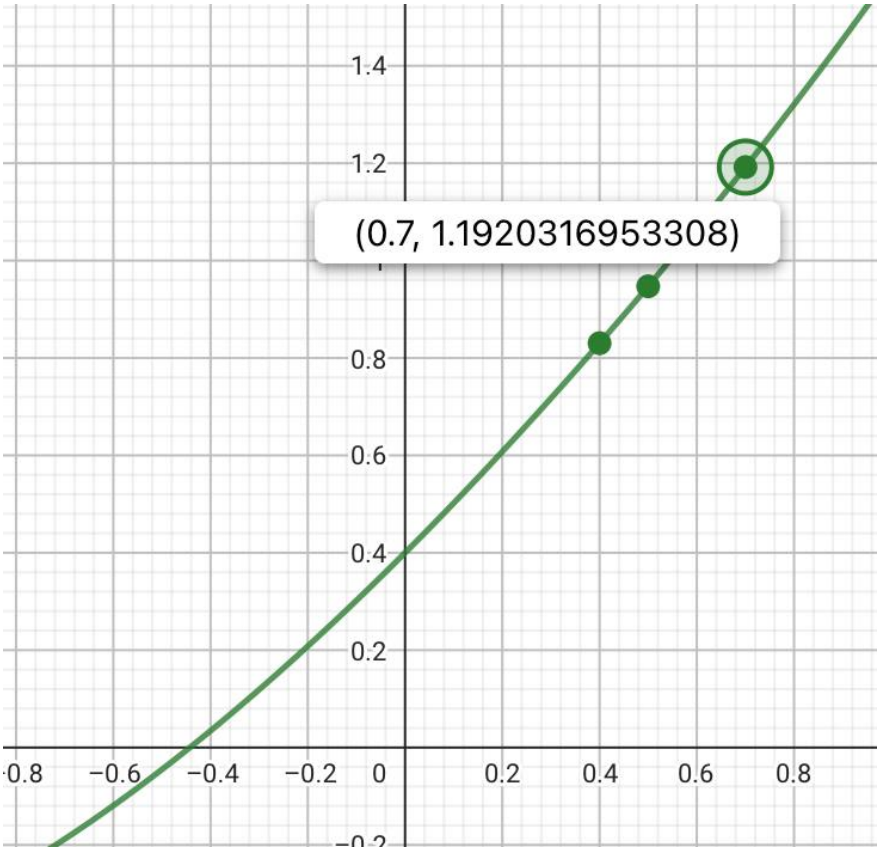

Рис. 7: График функции $f^{(4)}(\xi)$

Приложение 3

```
def tk(n):
    restk = []
    for i in range(0, n+1):
        a = ((2*i+1)/(2*n+1)) * math.pi
        restk.append(math.cos(a))

    return restk


def xk(a, b, tk, n):
    resxk = []
    for i in range(n+1):
        x = (a+b)/2 + ((b-a)/2)*tk[i]
        resxk.append(x)
    return resxk


def fault_m(n, a, b):
    m = 0.3465957706175
    x = m/(second_part.factorial(n+1))
    y = (math.pow(b-a, n+1))/(math.pow(2, 2*n+1))
    return x*y
```

Рис. 8: Методы для нахождения t_k, x_k и погрешности по формуле

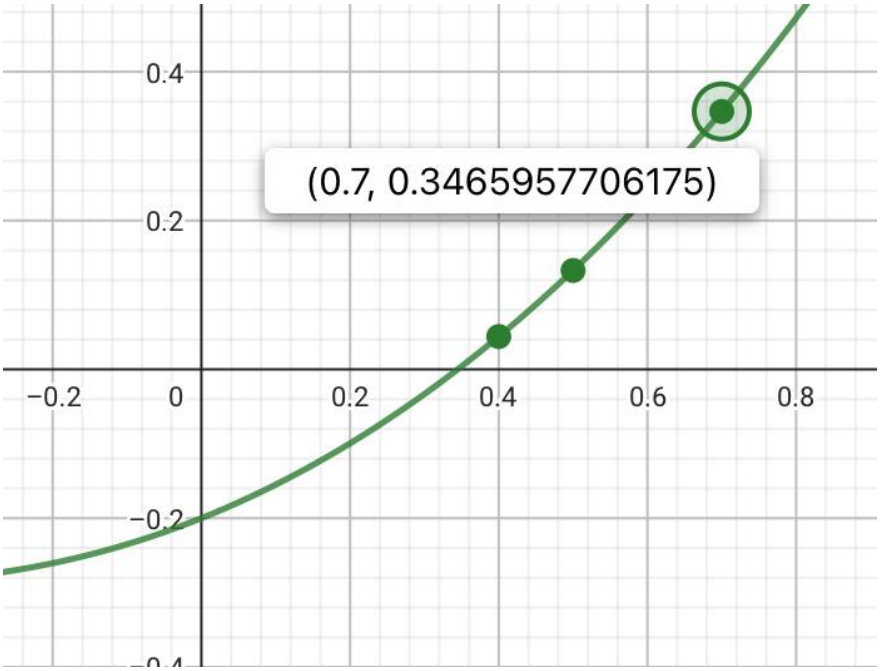

Рис. 9: График функции $f^{11}(x)$

Приложение 4

```
def sum_m(x_i_list, a, b, m):
    result = 0
    for i in range(m):
        result += math.pow(x_i_list[i], a) * math.pow(x_i_list[i], b)

    return result


def create_matrix(n, x_i_list, m):
    cf = np.zeros([n, n])
    for i in range(n):
        for j in range(n):
            elem = sum_m(x_i_list, i, j, m + 1)
            cf[i, j] = elem

    return cf
```

Рис. 10: Построение матрицы

```
def f_res(n, m, x_i_list, f_i_list):
    vec = []
    for i in range(n + 1):
        sum_f = 0
        for j in range(m + 1):
            sum_f += math.pow(x_i_list[j], i) * f_i_list[j]
        vec.append(sum_f)

    return vec
```

Рис. 11: Построение вектора F

```
def fault(m, x, n, x_i_list, f_i_list):
    sum_fault = 0
    for i in range(m+1):
        sum_f = phi(x, n, x_i_list[i]) - f_i_list[i]
        sum_fault += math.pow(sum_f, 2)

    return math.sqrt(sum_fault)
```

Рис. 12: Нахождение погрешности