МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»



Инженерная школа ядерных технологий

01.03.02 «Прикладная математика и информатика»

**Лабораторная работа №1**

по дисциплине:

**Численные методы**

Вариант 14

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Выполнил:** |  | | | | |
| Студенты группы | 0В01 |  | Белясов А.А. |  |  |
|  |  |  |  |  |
| **Проверил:** | Крицкий О.Л. | | | | |
| преподаватель |  |  |  |  |  |
|  |  |  |  |  |  |

Томск 2023

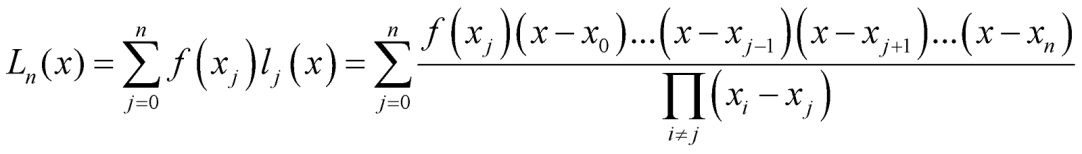
**Цель работы:**

При помощи двух различных численных методов интерполирования найти интерполяционные многочлены, построить графики по полученным результатам и сравнить результаты.

**Теория:**

*Интерполяционный многочлен Лагранжа*

Интерполяционный многочлен Лагранжа можно задать следующим образом:



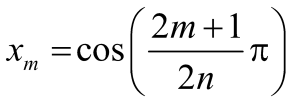
*Узлы Чебышева*

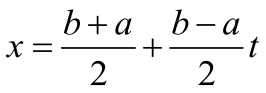
Под узлами Чебышева понимаются корни многочлена Чебышева первой степени. Они используются для улучшения результатов интерполяции.

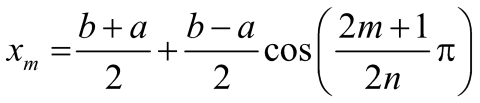
Корни многочлена легко находятся из

https://lh4.googleusercontent.com/PMlpOWCmwPWe_hEZr6XNxBD1yg2QtiNCmle9cwH0yYfAgFqWJRuKAfa2HDkNDb9UMqQVZ4u5jfM2tsA54oAwDucibjE3wCmatJoQApTx8loaf24xcYG7rSAtmuJLHCm5_1hzCI-te6WiJ-XH5nrVJhNgPqpN4UbiNDtaFuT9dwkclTfGv85DBpBOpEgrX_MhxOir3w,

тогда, выражая х, получим:

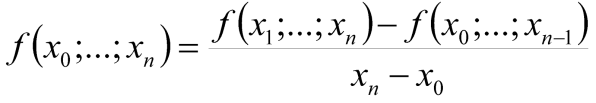
, m = 0,1,…, n-1, n>1

Если первоначальный интервал интерполирования не совпадает с [-1;1], то корни многочлена Чебышева нужно перенести в интервал [a;b] c помощью линейного преобразования вида , -1≤t≤1:



*Разделенные разности*

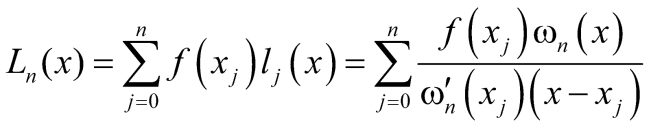
Пусть в известных узлах функции xi вычислены (заданы) значения функции f(xi), i = 0, 1, 2, …, n, тогда разделенной разностью n-го порядка назовем отношение вида



Получается рекурсивная функция, позволяющая вычислить разность n-го порядка по младшим порядкам.

*Формула Ньютона*

Для вывода интерполяционного многочлена Ньютона используем интерполяционный многочлен Лагранжа



Из чего можно получить формулу Ньютона для интерполяции:

https://lh3.googleusercontent.com/wndN5y_zb1jL98tiTCL8FT5bafDBMriOw6di78ZJdEnA5t_lr0ArZhBHFlhURXXDLgOHX0gLUWJsDhHTcMmEl5bXGSiSWu9179Y1_zpA0oMZ2E9tSLzAm70oN6YEhznw6xXpVSWtUSdQoiwsEYG6GRb3EPCH-MwlQI-2jUSqMjpbFJzrecHAKSSsJKykatLXE-385g

https://lh5.googleusercontent.com/KsYtYnZ0yx3BAbUctKkBg8_CPmpHt9ZfqESx4cUdD5wRY3M7TlDxdkx1N5RD48Q0a4NLEEzz7610JRWZjcxV3LH_uawyUDaf0W99fildfC2B8m4UeLmZyLU9SaVi5n9b25gWRgotVssyJMaGIsH2zateqhjdLeXuohI0h3YVlQK_X-7spp60-lQP9a_2c3_kuw3X8g,

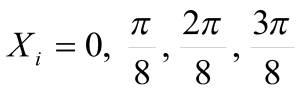
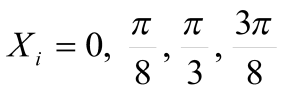
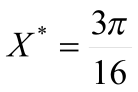
Где – разделенные разности.

*Погрешность*

Погрешность в точке будем считать как модуль разности точного значения в данной точке и интерполированного значения в этой же точке. Т.е.

**Ход работы:**

Используя аналитическое задание функции, вычислить таблицу ее значений в точках для каждого задания. Построить интерполяционные многочлены Лагранжа (задание a), Ньютона (задание b), проходящие через эти точки .Построить многочлен Лагранжа в точках, являющихся корнями многочленов Чебышева на том же интервале (задание с). Вычислить достигаемое практическое значение погрешности интерполяции в точке для всех использованных формул.

https://lh6.googleusercontent.com/GbR7PF_E5bUHPkVQwskec29u4QUJmtGEZsP4qK1X7dT9G6w7VzzQCSruDJuwyOJrfwOKohp_steJf6H0a8vOb37vNh7VAiZJfivnRLrdqwhLQR_0004B5eP-NPjQPVbKzSD393x2jvYA3Q_j4_r6y1AOJJwejxiw41X3MhTTY8zCAgl-RwujrBFrszxNvpb1YxaFFA, a) ; б) ; .

a) Построение интерполяционного многочлена Лагранжа.

Для этого создадим отдельную функцию, на вход которой будут поступать массивы известных аргументов и значений функций в данной точке, а так же массив точек, в которых необходимо узнать значения. Хотя изначальная функция нам и известна и есть возможность передать ее, чтобы внутри функции посчитать массив ординат, однако делать так не стоит, так как пострадает универсальность алгоритма. Поэтому сначала происходит расчет ординат, а уже после он передается в функцию.

function[result] = lagrange(x, y, z)

sum = 0;

for j=1:length(x)

p = 1;

for i=1:length(x)

if i~=j

p=p.\*(z-x(i))/(x(j)-x(i));

end

end

sum = sum+p\*y(j);

end

result=sum;

end

Собственно, результатом функция возвращает массив ординат для заданных точек.

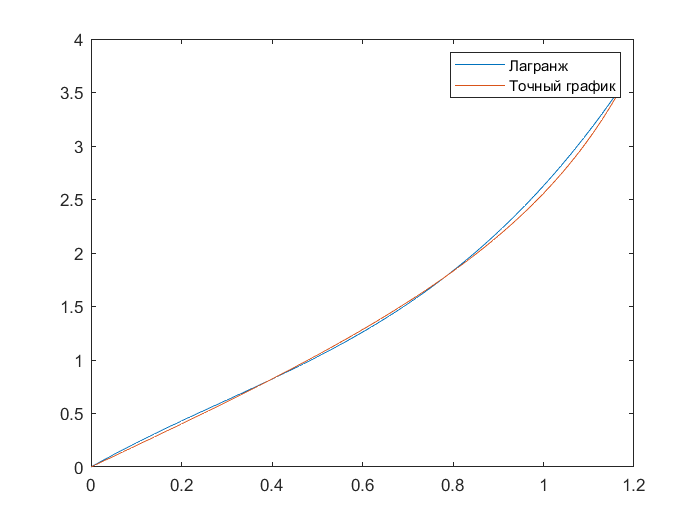


Рисунок 1 – График метода Лагранжа

Эти и все последующие графики построены и буду построены для 117 точек, а именно взят отрезок, указанный в задании [0, 3pi/8] с шагом 0.01.

b) Построим многочлен Лагранжа, узлами интерполяции которого являются корни многочлена Чебышева. Создадим новую функцию, на вход которой поступает исходный массив абсцисс. На выходе получается новый массив абсцисс, состоящий из корней многочлена Чебышева.

function [x] = chebyshev(x)

len = length(x);

a = x(1);

b = x(len);

for i = 1:len

x(i) = (a+b)/2 + (b-a)/2\*cos((2\*i-1)/2/len\*pi);

disp(x(i))

end

end

По сути, вместо передачи функции массива можно обойтись передачей трех переменных: левой границы интервала, правой границы и количества желаемых узлов интерполяции. Однако оставим этот пункт на дальнейшее усовершенствование алгоритма с возможной перегрузкой функции lagrange.

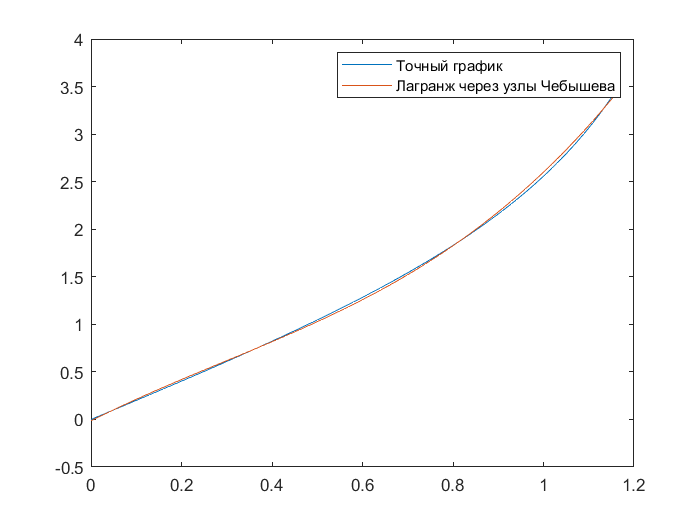


Рисунок 2 – График Лагранжа в узлах Чебышева

Как видно из графика, интерполяция в узлах Чебышева показывает более близкий к истине результат. Кроме того, данный способ нахождения узлов интерполяции позволяет избежать феномена Рунге.

с) Построим интерполяционный многочлен Ньютона. В этот раз для решения потребуется ввести две функции: одна будет считать разделенные разности, а вторая – полином Ньютона.

Рассмотрим функцию, считающую разности. На вход подаются два массива: абсцисс и ординат, которые заданы по условию. Так как этот алгоритм мало чем отличается от Лагранжа, то достаточно немного изменить одну строку и в результате уже будет разделенная разность, а не интерполяционный многочлен Лагранжа.

function [sum] = raznost(x, y)

sum = 0;

for i = 1:length(x)

p = 1;

for j = 1:length(x)

if (i~=j)

p = p\*(x(i)-x(j));

end

end

sum = sum + y(i)/p;

end

end

Аналогично, действуя согласно алгоритму нахождения многочлена Ньютона, реализовываем и этот алгоритм. На вход подаются все те же аргументы, что и в функцию Лагранжа: интерполяционная сетка и массив абсцисс. Данный алгоритм на языке Matlab примет следующий вид:

function sum = newton(x, y, z)

sum = 0;

for j = 1:length(x)

p = ones(size(z));

for i = 1:j-1

p = p .\* (z-x(i));

end

sum = sum + p .\* raznost(x(:, 1:j), y(:, 1:j));

end

end

Теперь построим график, используя те же самые 117 точек для честного сравнения результатов.

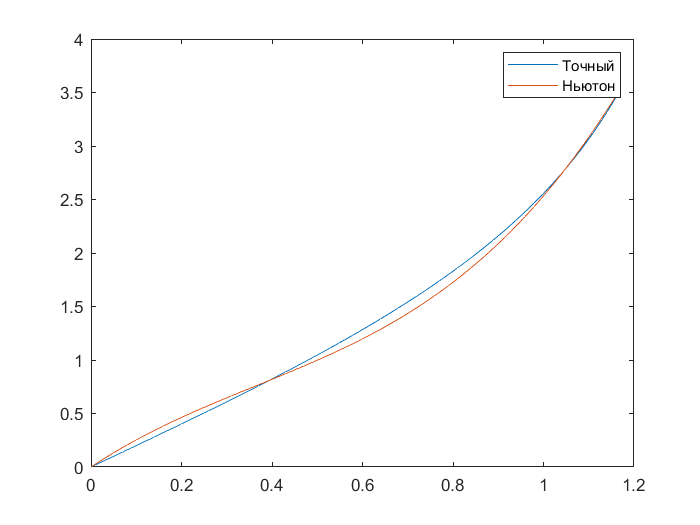


Рисунок 3 – График формулы Ньютона

Как видно из графика, результат немного отличается от предыдущих в худшую сторону.

И для чистоты эксперимента, сопоставим все полученные графики в одних осях координат.

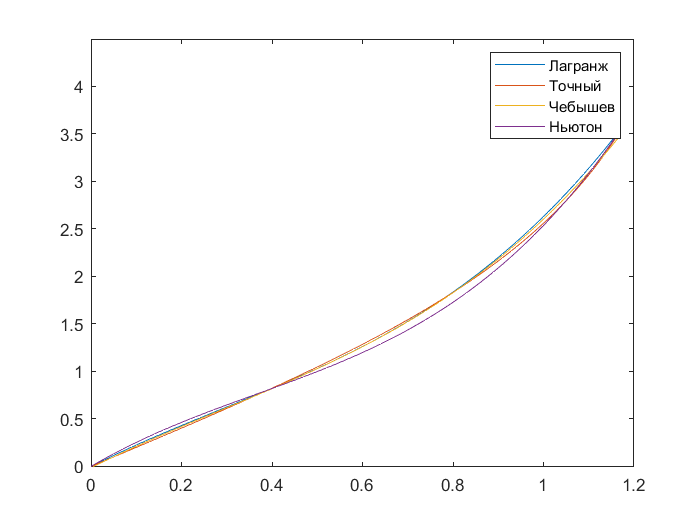


Рисунок 4 – Графики всех методов

c) Погрешность в заданной точке находится как модуль разности значения точной функции в данной точки и результата, полученного интерполяцией. Таким образом, получены следующие результаты:



Рисунок 5 – Погрешности методов

Результаты совпадают с полученными из графика, метод Ньютона получил наибольшую погрешность.

Основной текст программы имеет следующий вид:

clc; clear all;

x = [0,pi/8,2\*pi/8,3\*pi/8];

xx = 3\*pi/16;

f = @(y) tan(y)+y;

xi = 0:0.01:(3\*pi/8);

y = f(x);

figure

plot(xi, lagrange(x,y,xi))

errorLag = abs(f(xx)-lagrange(x,y,xx));

hold on

plot(xi, f(xi))

hold on

chebX = chebyshev(x);

plot(xi, lagrange(chebX, f(chebX), xi))

errorCheb = abs(f(xx)-lagrange(chebX, f(chebX), xx));

legend('Точный график','Лагранж через узлы Чебышева')

x = [0,pi/8,pi/3,3\*pi/8];

y = f(x);

plot(xi, newton(x,y,xi))

errorNewt = abs(f(xx)-newton(x,y, xx));

legend("Лагранж","Точный","Чебышев",'Ньютон')

hold off

disp('error Lagrange:')

disp(errorLag)

disp('error Chebyshev:')

disp(errorCheb)

disp('error NewTon:')

disp(errorNewt)

В зависимости от того, какие графики требовалось вывести, комментировались соответствующие ненужные куски кода. Построение графиков внутри функций интерполяций уменьшает универсальность кода, так как далеко не всегда требуется строить графики полученных результатов, поэтому было принято решение строим графики в основной программе.

**Вывод:**

В ходе выполнения лабораторной работы было реализовано 3 алгоритма численного интерполирования: метод Лагранжа, метод Лагранжа с узлами в корнях многочлена Чебышева, метод Ньютона. При сравнении результатов было выявлено, что алгоритмы Лагранжа имеют наибольшую точность.