Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Казанский (Приволжский) федеральный университет»

Институт вычислительной математики и информационных технологий Кафедра прикладной математики и искусственного интеллекта



ОТЧЕТ

по дисциплине «Численные методы»

Интерполирование трансцендентных функций алгебраическими многочленами Вариант 8

Выполнил: студент гр. 09-221

Саитов М. А. Проверил: ассистент Глазырина

О. В.

КАЗАНЬ 2022

СОДЕРЖАНИЕ

[ПОСТАНОВКА ЗАДАЧИ 3](#_bookmark0)

[ХОД РАБОТЫ 5](#_bookmark1)

[ВЫВОД 12](#_bookmark2)

[ЛИСТИНГ](#_bookmark3) [ПРОГРАММЫ 13](#_bookmark3)

ПОСТАНОВКА ЗАДАЧИ

1. Протабулировать erf(x) на отрезке [a;b] с шагом h точностью *ε*, основываясь на ряде Тейлора, предварительно вычислив его

2 ∞

*n x*2*n*+1

*erf* ( *x*)= √*~~π~~*∑ (− 1)

*n*= 0

*n*!( 2*n*+1)

(1)

где 𝑎 = 0, 𝑏 = 2, ℎ = 0,2, 𝜀 = 10−6 и получить, таким образом, таблицу:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ◻0 | ◻1 | ◻2 | *…* | ◻n |
| f0 | f1 | f2 | *…* | fn |

f1 = erf(x), ◻i = ◻ + ◻ ∗ ℎ, ◻ = 0, … , ◻

1. По полученной таблице значений построить интерполяционный полином Лагранжа, приближающий erf(x)

*∞ n x*− *xj*

*Ln* ( *x*) = ∑ *f* (*xi* ) ∏ *x* − *x* ¿¿

*n*= 0

*j* = 0 *i j i* ≠ *j*

(2)

1. Вычислить погрешность интерполирования

*εn*= *m*⏟*ax ε* ( *x*) *, ε* ( *x*)=|*erf* ( *x*) − *Ln*( *x*)|

*xϵ* ( *a,b*)

(3)

1. Изменяя количество узлов интерполяции от 5 до 51 и вычисляя максимальную погрешность для каждого случая, построить таблицу изменения погрешности.
2. Выполнить пункты 1, 2, 3, 4 для узлов Чебышева

*x* = *b*+*a* + *b*− *a* cos 2*i*+1 *π; i* = 0*, n*

*i* 2 2

2*n*+2

(4)

ХОД РАБОТЫ

1. Для построения значений интерполирующей функции в каждой точке, выделяется общий член *an* из ряда Тейлора, и затем находится *an*+1 *.* После этого

*q* = *an*+1 *.*

*a q* ∗ *a .*

вычисляется *n*

*a*

Благодаря этому возможно вычислить любой

*n*

*i*= *i* − 1

*i* − 1

Далее

вычисляем значение функции в 6 узлах интерполяции:

|  |  |
| --- | --- |
| *xn* | f(*xn* ) |
| 0 | 0 |
| 0.4 | 0.428392 |
| 0.8 | 0.742101 |
| 1.2 | 0.910314 |
| 1.6 | 0.976348 |
| 2.0 | 0.995322 |

Таблица 1. Вычисление значений функции в 6 узлах интерполяции

1. Вычислим функцию для 11 узлов интерполяции и построим интерполяционный полином Лагранжа, приближающий erf(x) по формуле (1)

|  |  |
| --- | --- |
| xn | f(xn) |
| 0 | 0 |
| 0.2 | 0.222703 |
| 0.4 | 0.428392 |
| 0.6 | 0.603856 |
| 0.8 | 0.742101 |
| 1.0 | 0.842701 |
| 1.2 | 0.910314 |
| 1.4 | 0.952285 |
| 1.6 | 0.976348 |
| 1.8 | 0.989091 |
| 2.0 | 0.995322 |

Таблица 2. Вычисление значений функции для 11 узлов интерполяции

Построим интерполяционный полином Лагранжа на основе приведенной таблицы (Таблица 2)

|  |  |
| --- | --- |
| xn | Ln(xn) |
| 0 | 0 |
| 0.2 | 0.222302 |
| 0.4 | 0.428501 |
| 0.6 | 0.603918 |
| 0.8 | 0.742029 |
| 1.0 | 0.842701 |
| 1.2 | 0.910379 |
| 1.4 | 0.952234 |
| 1.6 | 0.976267 |
| 1.8 | 0.989366 |
| 2.0 | 0.995322 |

Таблица 3. Полином Лагранжа для 11 узлов

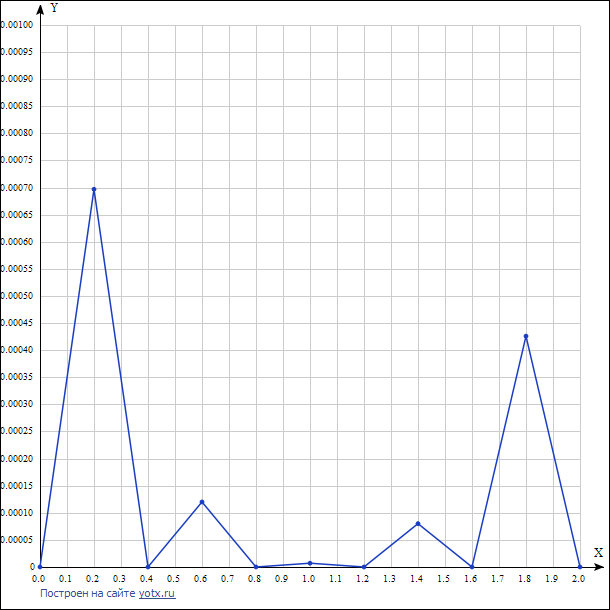
1. Вычисляем погрешность интерполирования по формуле (3) и получаем следующую таблицу значений:

Таблица 4. Погрешность интерполирования для 11 узлов Получили погрешности в 11 узлах интерполяции, построим график на основе

|  |  |
| --- | --- |
| *xn* | *ε* ( *x*) |
| 0 | 0 |
| 0.2 | 0.000697 |
| 0.4 | 0 |
| 0.6 | 0.000120 |
| 0.8 | 0 |
| 1.0 | 0.000007 |
| 1.2 | 0 |
| 1.4 | 0.000080 |
| 1.6 | 0 |
| 1.8 | 0.000426 |
| 2.0 | 0 |

полученных значений:

Ри су но к 1.



Гр аф ик по гр еш но ст и дл я 11

уз ло в ин те рп ол яц ии

Построим таблицу по полученным значениям:

|  |  |  |  |
| --- | --- | --- | --- |
| x | f(x) | L(x) | ε(x) |
| 0 | 0 | 0 | 0 |
| 0.2 | 0.222703 | 0.222302 | 0.000697 |
| 0.4 | 0.428392 | 0.428501 | 0 |
| 0.6 | 0.603856 | 0.603918 | 0.000120 |
| 0.8 | 0.742101 | 0.742029 | 0 |
| 1.0 | 0.842701 | 0.842701 | 0.000007 |
| 1.2 | 0.910314 | 0.910379 | 0 |
| 1.4 | 0.952285 | 0.952234 | 0.000080 |
| 1.6 | 0.976348 | 0.976267 | 0 |
| 1.8 | 0.989091 | 0.989366 | 0.000426 |
| 2.0 | 0.995322 | 0.995322 | 0 |

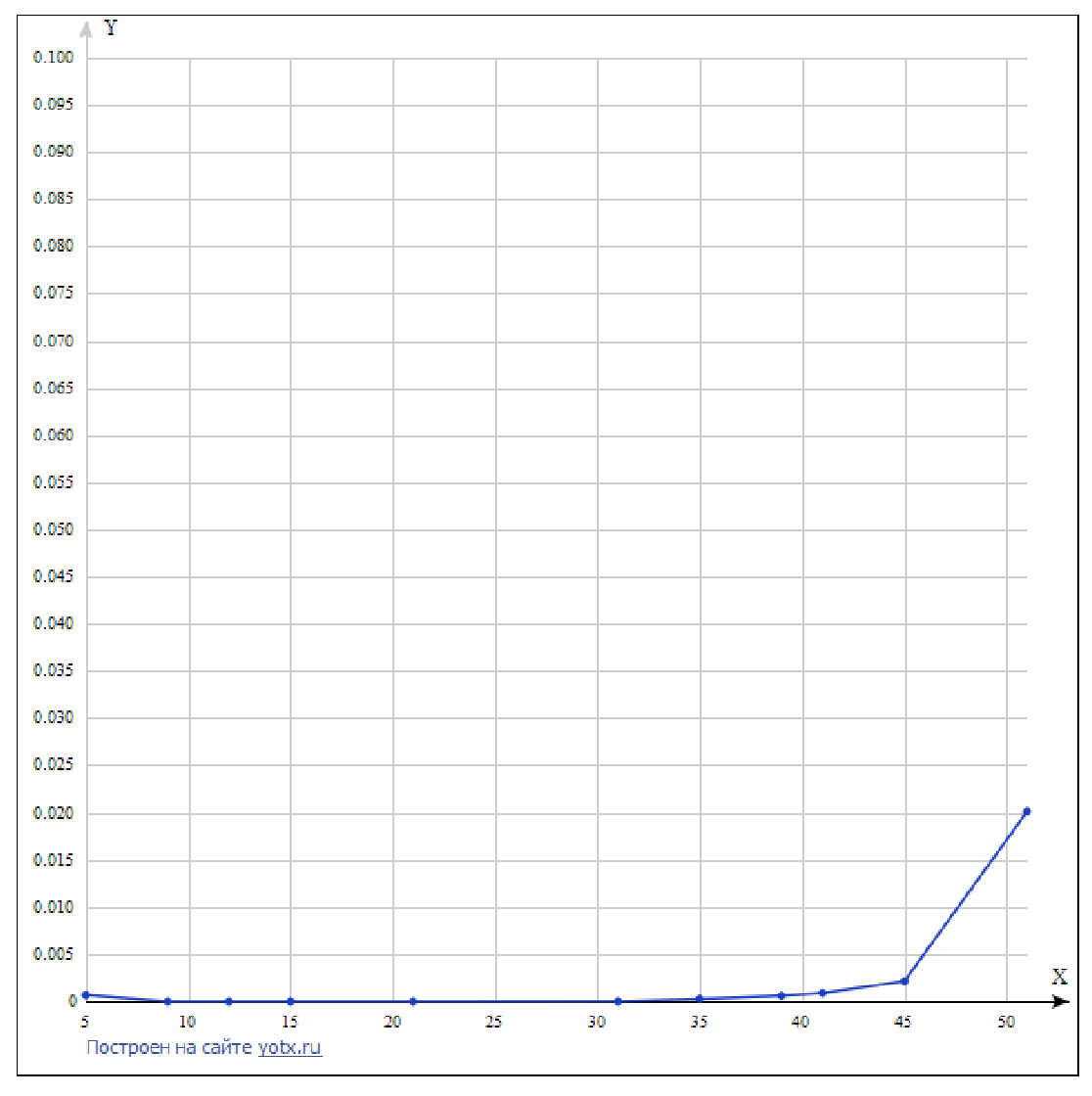
Таблица 5. Данные по 11 узлам интерполяции

1. Вычислим погрешности в различных узлах интерполяции (от 5 до 51) с помощью нахождения полинома Лагранжа и функции в них, выбрав максимальную погрешность, построим следующую таблицу (Таблица 6)

|  |  |
| --- | --- |
| n | εmax |
| 5 | 0.0006965558 |
| 9 | 0.0000012007 |
| 12 | 0.0000001829 |
| 15 | 0.0000019789 |
| 21 | 0.0000007804 |
| 31 | 0.0000091787 |
| 35 | 0.0002644998 |
| 39 | 0.0006179874 |
| 41 | 0.0009027823 |
| 45 | 0.0021416198 |
| 51 | 0.0201359372 |

Таблица 6. Максимальная погрешность интерполирования для различных узлов (от 5 до 51)

Построим график на основе полученных максимальных значений:

Рисуно к 2.

Макси мальна я погреш ность в различ ных узлах интерп оляции (от 5

до 51)

ожно замети ть, что при количе стве узлов n

М

= 51

происх

одит резкое возрастание максимальной погрешности из-за накопления вычислительной ошибки

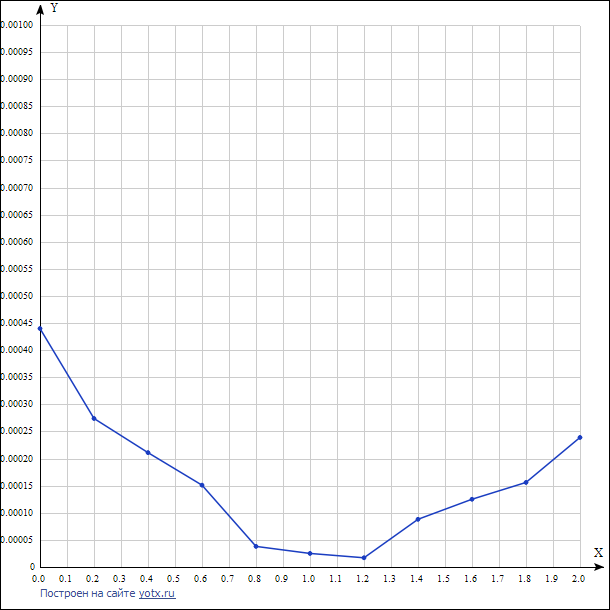
1. Проделаем пункты 1-4 для узлов Чебышева. Функция f(xn) не изменяется, а Ln(xn) при n = 5 изменилась:

|  |  |
| --- | --- |
| xn | Ln(xn) |
| 0 | 0.000440 |
| 0.2 | 0.222429 |
| 0.4 | 0.428603 |
| 0.6 | 0.604007 |
| 0.8 | 0.742063 |
| 1.0 | 0.842676 |
| 1.2 | 0.910331 |
| 1.4 | 0.952197 |
| 1.6 | 0.976224 |
| 1.8 | 0.989246 |
| 2.0 | 0.995083 |

Таблица 7.Полином Лагранжа для 11 узлов Чебышева Теперь составим таблицу для погрешности 11 узлов Чебышева

|  |  |
| --- | --- |
| xn | E(x) |
| 0 | 0.000440 |
| 0.2 | 0.000274 |
| 0.4 | 0.000211 |
| 0.6 | 0.000151 |
| 0.8 | 0.000038 |
| 1.0 | 0.000025 |
| 1.2 | 0.000017 |
| 1.4 | 0.000088 |
| 1.6 | 0.000125 |
| 1.8 | 0.000156 |
| 2.0 | 0.000239 |

Таблица 8. Погрешности для 11 узлов Чебышева Построим график:

исун ок 3. Граф ик погре шнос ти для 11

чебы шевс ких узлов

цу по известным значениям при узлах Чебышева:

П

остро им табли

|  |  |  |  |
| --- | --- | --- | --- |
| x | f(x) | L(x) | ε(x) |
| 0 | 0 | 0.000440 | 0.000440 |
| 0.2 | 0.222703 | 0.222429 | 0.000274 |
| 0.4 | 0.428392 | 0.428603 | 0.000211 |
| 0.6 | 0.603856 | 0.604007 | 0.000151 |
| 0.8 | 0.742101 | 0.742063 | 0.000038 |
| 1.0 | 0.842701 | 0.842676 | 0.000025 |
| 1.2 | 0.910314 | 0.910331 | 0.000017 |
| 1.4 | 0.952285 | 0.952197 | 0.000088 |
| 1.6 | 0.976348 | 0.976224 | 0.000125 |
| 1.8 | 0.989091 | 0.989246 | 0.000156 |

12

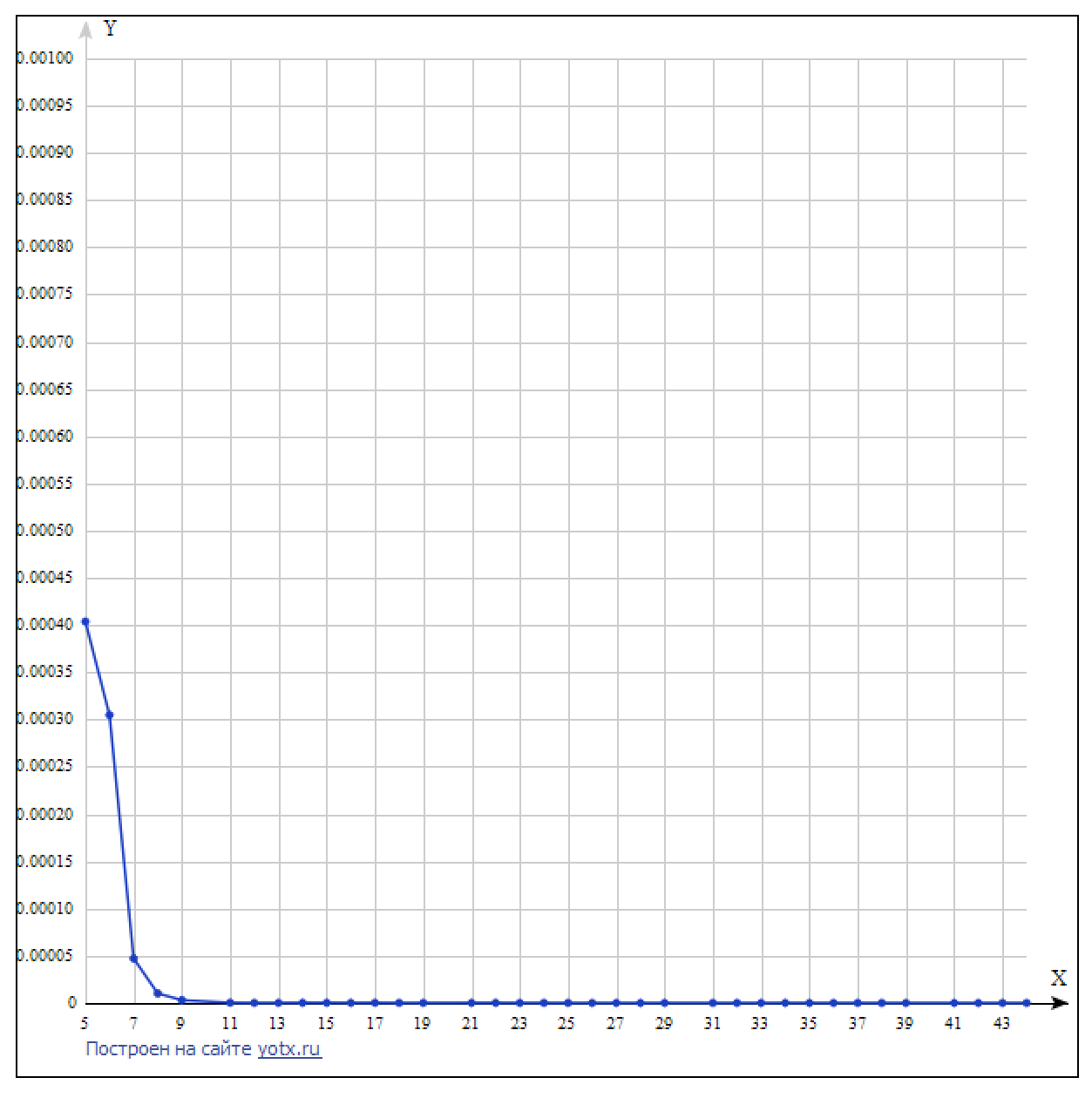
|  |  |  |  |
| --- | --- | --- | --- |
| 2.0 | 0.995322 | 0.995083 | 0.000239 |

Таблица 9. Данные по 6 узлам Чебышева

Меняя количество узлов Чебышева, построим таблицу с максимальными значениями погрешностей в различных узлах интерполяции:

|  |  |
| --- | --- |
| n | max |
| 5 | 0.0004038165 |
| 9 | 0.0000030230 |
| 11 | 0.0000002762 |
| 13 | 0.0000001375 |
| 35 | 0.0000000661 |
| 44 | 0.0000000445 |

Таблица 10. Погрешности в различных узлах Чебышева (от 5 до 44)

По ст ро им гр аф ик:

ВЫВОД

В ходе проделанной работы, был построен полином Лагранжа в различных узлах интерполяции и узлах Чебышева. Найдена погрешность при различных узлах. Подводя итог проделанной работы, можно сделать следующие выводы. При увеличении количества узлов интерполирования, равнораспределённые узлы сначала дают резкое уменьшение погрешности, но из-за накапливающейся ошибкив памяти компьютера при 𝑛 = 51 погрешность начинает возрастать. Обратный результат был получен при использовании узлов Чебышева, а именно, при увеличении количества узлов график погрешности стремительно убывал начиная с

𝑛 = 9 и не имел аномалий на большом количестве узлов. По итогу проделанных экспериментов, узлы Чебышева показали себя лучше при вычислении приближающей функции.

ЛИСТИНГ ПРОГРАММЫ

(Язык программирования C++)

#include <stdio.h> #include <math.h> #define Eps pow(10, -6)

double max(double\* function, size\_t size){ double max = -1000;

for(int i = 0; i < size; i++){ if (function[i] > max)

max = function[i];

}

return max;

}

double Qn(double n, double x){

return -(((x \* x) / (n + 1)) \* ((2 \* n + 1) / (2 \* n + 3)));

}

double countF(double x){ double eps = Eps;

int n = 1;

double prevA = x; double currentA = x; double result = x;

while (abs(prevA) >= eps){

currentA = Qn(n - 1, x) \* prevA; result += currentA;

prevA = currentA; n++;

}

result \*= 2 / sqrt(M\_PI); return result;

}

double\* F\_Array(){

double\* F = new double[11]; double eps = Eps;

double h = 0.2; int i = 0;

for(double x = 0; x <= 2; x += h, i++) F[i] = countF(x);

return F;

}

double countL(double x, int nodesCount){ double result = 0.0;

double h = 2/((double)nodesCount); for (int i = 0; i < nodesCount+1; i++){

double xi = i\*h;

double Fx = countF(xi);

for (int j = 0; j < nodesCount+1; j++){ if (i == j)

continue; double xj = j\*h;

Fx \*= (x - xj) / (xi - xj);

}

result += Fx;

}

return result;

}

double\* L\_Array(int nodesCount){ double\* L = new double[11]; double h = 0.2;

for(int i = 0; i < 11; i++){ double x = i\*h;

L[i] = countL(x, nodesCount+1);

}

return L;

}

double\* E\_Array(int nodesCount){ double\* E = new double[11]; double h = 0.2;

for(int i = 0; i < 11; i++){ double x = i\*h;

E[i] = abs(countF(x) - countL(x, nodesCount));

}

return E;

}

double chebNode(double i, int nodesCount){ double b = 2;

double a = 0;

double result = (b+a)/2 + ((b-a)/2)\*cos(((2\*i+1)/(2\*nodesCount+1))\*M\_PI); return result;

}

double countL\_Cheb(double x, int nodesCount){

double result = 0.0;

for (int i = 0; i < nodesCount+1; i++){ double xi = chebNode(i, nodesCount); double Fx = countF(xi);

for (int j = 0; j < nodesCount+1; j++){ if (i == j)

continue;

double xj = chebNode(j, nodesCount); Fx \*= (x - xj) / (xi - xj);

}

result += Fx;

}

return result;

}

double\* L\_ArrayCheb(int nodesCount){ double\* L = new double[11];

double x = 0.0;

for(int i = 0; i < 11; i++){

L[i] = countL\_Cheb(x, nodesCount); x+=0.2;

}

return L;

}

double\* E\_ArrayCheb(int nodesCount){ double\* E = new double[11];

double\* F = F\_Array();

double\* L = L\_ArrayCheb(nodesCount); for(int i = 0; i < 11; i++){

double F\_x = F[i]; double L\_x = L[i]; E[i] = abs(F\_x - L\_x);

}

delete[] F; delete[] L; return E;

}

void menu(bool& cheb){ int res;

printf("1 — Узлы Интерполяции\n2 — Узлы Чебышева\n"); scanf("%d", &res);

if (res == 1) cheb = true; else cheb = false;

}

void printResults(double\* F, double\* L, double\* E){ int spaces = 0;

double x = 0.0;

for(int i = 0; i < 11; i++){ spaces = abs((i/10) - 2);

printf("x%d = %.1f%\*s| ", i, x, spaces, "");

printf("F(x) = %.6f | L(x) = %.6f | E(x) = %.6f\n", F[i], L[i], E[i]); x+=0.2;

}

printf("Emax = %.6f\n", max(E, 11));

}

void printResults(int node, double\* E){

int spaces = (node/10 > 0)? 1: 2;

printf("nodes = %d%\*s| Emax = %.10f\n", node, spaces, "", max(E, 11));

}

int main(){

bool isCheb = false; menu(isCheb); if(!isCheb){

for(int nodesCount = 5; nodesCount < 52; nodesCount++){ if (nodesCount % 10 == 0)

continue;

printResults(nodesCount, E\_Array(nodesCount));

}

printf(" \n");

}

else{

for(int nodesCount = 5; nodesCount < 45; nodesCount++){ if (nodesCount % 10 == 0)

continue;

printResults(nodesCount, E\_ArrayCheb(nodesCount));

}

printf(" \n");

}

return 0;

}

Программа собирается через Makefile:

ifeq ($(wildcard ../build),)

$(shell mkdir ../build) endif

all: build compile

rebuild: clean build compile

build: countFunctions printResults

@g++ ../build/countFunctions ../build/printResults main.cpp -o ../build/main &&

clear

countFunctions:

@g++ -c countFunctions.cpp -o ../build/countFunctions

printResults:

@g++ -c printResults.cpp -o ../build/printResults

compile:

@../build/main

clean: