|  |
| --- |
| БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ |
| Факультет прикладной математики и информатики |
| Кафедра информационных систем управления |
| Лукьянович Александр Сергеевич  10 группа, 2 курс |
| Лабораторная работа |
|  |
|  |
| **Метод наименьших квадратов**    **Преподаватель**  *Полевиков Виктор Кузьмич* |
|  |
| Дата сдачи:  25.05.2017 |

Оглавление

[Постановка задачи 3](#_Toc483364576)

[Краткая теория 4](#_Toc483364577)

[1) n=1 5](#_Toc483364578)

[2) n=2 5](#_Toc483364579)

[3) n=3 6](#_Toc483364580)

[Листинг программы 7](#_Toc483364581)

[Результаты 8](#_Toc483364582)

# Постановка задачи

, рассматриваемая функция . Построим её табличное задание в этих узлах:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 0.2 | 0.4 | 0.6 | 0.8 | 1.0 |
|  | 0 | 0.239925 | 0.461484 | 0.353324 | 0.106633 | 0 |

Методом наименьших квадратов построить лучшее среднеквадратичное приближение, выбирая

Также вычислить абсолютные и относительные среднеквадратические погрешности для построенных полиномов.

# Краткая теория

Вес , потому формулы немного упрощаются.

Раз функции заданы таблично, то их скалярное произведение определим формулой

Тогда условие наилучшего среднеквадратичного приближения функции функцией принимает вид

У нас полиномиальная аппроксимация. В этом случаем необходимо решить систему

Поскольку степени на любом отрезке образуют чебышёвскую систему, то определитель Грама отличен от нуля, следовательно, задача имеет единственное решение.

Погрешности вычисляются по формулам:

Поскольку функция задана таблично, то норма её будет находиться как для вектора: корень из суммы квадратов значений в узлах.

Найдём требуемые полиномы:

## n=1

Эта функция наименьшая, когда

Система имеет вид

Определяем погрешности:

## n=2

## n=3

# Листинг программы

Программа выполнена на языке программирования C++, в среде Microsoft Visual Studio Community 2015.

#include "resources.h"

int main(int argc, char argv[]) {

setlocale(LC\_ALL, ".1251");

ofstream fout("output.txt", ios\_base::trunc);

if (!fout.good())

throw exception("");

fout << fixed << setprecision(6);

const int n = 6;

double f\_norm = 0.0;

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

fout << f(x(i)) << endl;

f\_norm += pow(f(x(i)), 2);

}

f\_norm = sqrt(f\_norm);

//дельта 1

double temp = 0.0;

for (int i = 0; i < n; i++)

temp += pow(0.088313 - f(x(i)), 2);

temp = sqrt(temp);

fout << "\n\nдельта = " << temp << "\nдругая дельта = " << temp / f\_norm \* 100 << endl;

//дельта 2

temp = 0.0;

for (int i = 0; i < n; i++)

temp += pow(-0.0725764286\*x(i) + 0.2298490476 - f(x(i)), 2);

temp = sqrt(temp);

fout << "\n\nдельта = " << temp << "\nдругая дельта = " << temp / f\_norm \* 100 << endl;

//дельта 3

temp = 0.0;

for (int i = 0; i < n; i++)

temp += pow(-1.6097265625\*x(i)\*x(i) + 1.5371501339\*x(i) + 0.0152188393 - f(x(i)), 2);

temp = sqrt(temp);

fout << "\n\nдельта = " << temp << "\nдругая дельта = " << temp / f\_norm \* 100 << endl;

fout.close();

return 0;

}

double f(double x) {

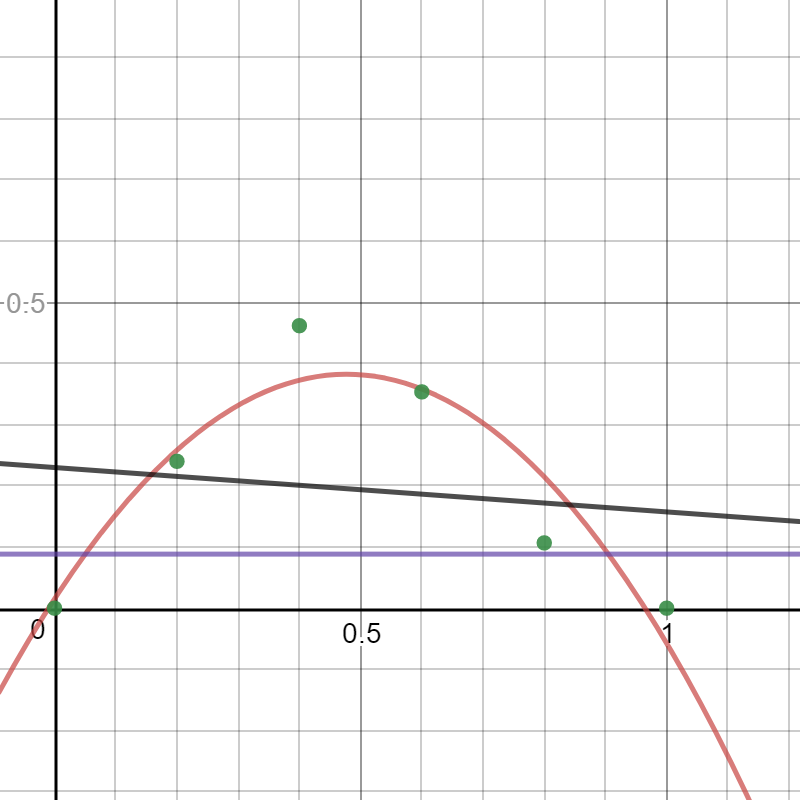
double temp = sin(PI\*x) / (x + 1);

return temp\*temp;

}

double x(int i) { return 0.2\*i; }

# Результаты

**