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

Выполнил: студент гр. 09-222 Фаррахова Л. Ф.

Проверил: ассистент Глазырина О. В.

Казань, 2023

СОДЕРЖАНИЕ

[ПОСТАНОВКА ЗАДАЧИ 3](#_Toc153290271)

[ХОД РАБОТЫ 4](#_Toc153290272)

[ВЫВОДЫ 12](#_Toc153290273)

[ЛИСТИНГ 13](#_Toc153290274)

# ПОСТАНОВКА ЗАДАЧИ

Необходимо изучить и сравнить различные способы приближённого вычисление функции.

Для этого необходимо:

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

где a = 0, b = 4, h = 0.4, ɛ = 10−6, и получить, таким образом, таблицу

|  |  |  |  |
| --- | --- | --- | --- |
| x0 | x1 | … | xn |
| f0 | f1 | … | fn |

𝑓𝑖 (𝑥) = 𝑆𝑖(𝑥), 𝑥𝑖 = 𝑎 + 𝑖 ∙ ℎ, 𝑖 = 0, … , 𝑛.

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

Затем вычислить погрешность интерполирования

* 1. Изменяя количество узлов интерполяции и вычисляя максимальную погрешность для каждого случая, построить таблицу изменения погрешности
  2. Повторить пункты 1-3 для узлов Чебышева, которые вычисляются по формуле

# ХОД РАБОТЫ

1) Вычислим значения функции в узлах интерполяции, основываясь на ряде Тейлора.

Чтобы избежать переполнения при вычислении факториалов в ряде Тейлора будем учитывать, что каждый последующий член ряда 𝑎𝑛+1 получается из предыдущего 𝑎𝑛 умножением на некоторую величину 𝑞𝑛, т.е. 𝑎n+1 = 𝑎𝑛𝑞𝑛.

Находим an =

Затем находим следующий член an+1 =

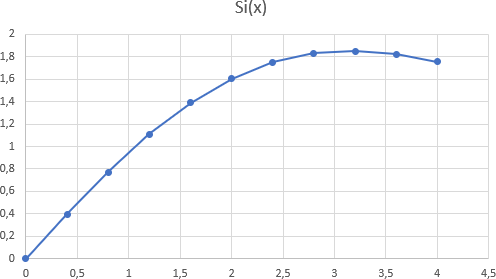
Тогда qn для данной функции: qn= =

Результаты табулирования функции на отрезке [0,4] с шагом 0,8 и точностью представлены в таблице 1.

|  |  |
| --- | --- |
| 𝑥 | 𝑆𝑖(𝑥) |
| 0 | 0 |
| 0,8 | 0,77209578 |
| 1,6 | 1,38918048 |
| 2,4 | 1,75248550 |
| 3,2 | 1,85140089 |
| 4 | 1,75820313 |

Таблица 1

На основе полученных вычислений постоим график (рисунок 1)

Рисунок 1. График полинома Лагранжа.

2) Вычислим интерполяционный полином Лагранжа, используя значения из таблицы 1 и формулу, приведенную в начале. Для определения погрешностей нам также понадобятся значения функции в 11 узлах интерполяции.

Результаты вычислений представлены в таблице 2.

|  |  |  |  |
| --- | --- | --- | --- |
| 𝑥 | 𝑆𝑖(𝑥) | 𝐿𝑛(𝑥) |  |
| 0 | 0 | 0 | 0 |
| 0,4 | 0,39646146 | 0,39566338 | 0,00079808 |
| 0,8 | 0,77209578 | 0,77209578 | 0 |
| 1,2 | 1,10804719 | 1,10831555 | 0,00026835 |
| 1,6 | 1,38918048 | 1,38918048 | 0 |
| 2 | 1,60541297 | 1,60522300 | 0,00018997 |
| 2,4 | 1,75248550 | 1,75248550 | 0 |
| 2,8 | 1,83209658 | 1,83235548 | 0,00025890 |
| 3,2 | 1,85140089 | 1,85140089 | 0 |
| 3,6 | 1,82194813 | 1,82120529 | 0,00074283 |
| 4 | 1,75820313 | 1,75820313 | 0 |

Таблица 2. Погрешность интерполирования для 11 узлов.

Можем заметить, что, если точка совпадает с узлом интерполяции, погрешность равна нулю, так как значение функции в этой точке совпадает со значением интерполяционного полинома Лагранжа.

На основе полученных вычислений постоим график погрешностей (рисунок 2).

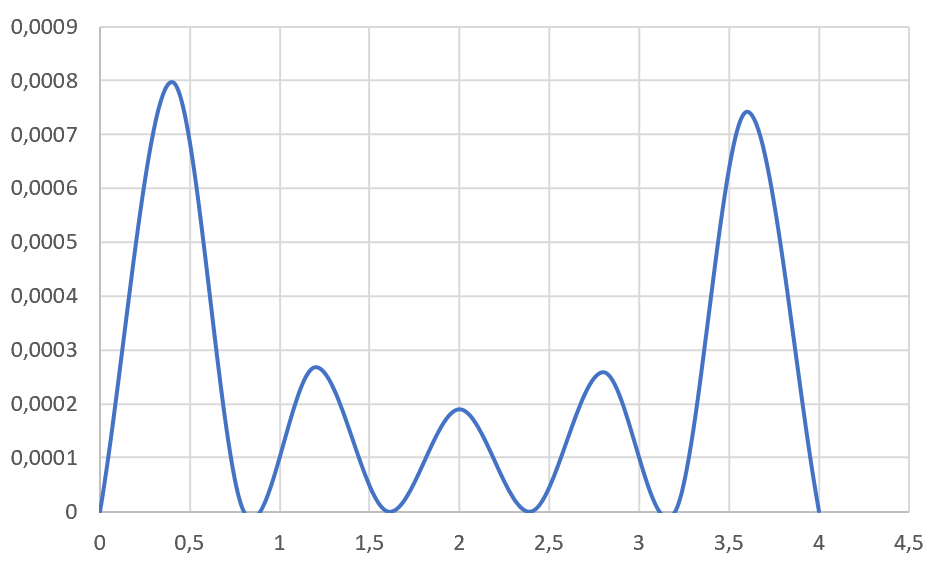


Рисунок 2. График погрешностей для 11 узлов интерполяции.

3) Для того чтобы исследовать зависимость погрешности нужно найти значения погрешностей при разном количестве узлов. Проведем эксперимент и выявим зависимость максимальной погрешности интерполирования от числа узлов интерполяции. Получим таблицу (таблица 3) и построим графики (рисунок 3, рисунок 4).

|  |  |  |  |
| --- | --- | --- | --- |
| 𝑛 | ε𝑛(𝑥) | 𝑛 | ε𝑛(𝑥) |
| 5 | 0,00079808 | 23 | 0,00003094 |
| 6 | 0,00004269 | 24 | 0,00005984 |
| 7 | 0,00001696 | 25 | 0,00011017 |
| 8 | 0,00000025 | 26 | 0,00009041 |
| 9 | 0,00000025 | 27 | 0,00115929 |
| 10 | 0,00000003 | 28 | 0,00067148 |
| 11 | 0,00000002 | 29 | 0,00407175 |
| 12 | 0,00000004 | 30 | 0,00346355 |
| 13 | 0,00000003 | 31 | 0,01396456 |
| 14 | 0,00000027 | 32 | 0,01501561 |
| 15 | 0,00000044 | 33 | 0,04655611 |
| 16 | 0,00000170 | 34 | 0,03495936 |
| 17 | 0,00000502 | 35 | 0,16693022 |
| 18 | 0,00000395 | 36 | 0,08486619 |
| 19 | 0,00001406 | 37 | 0,17307639 |
| 20 | 0,00001016 | 38 | 0,22084125 |
| 21 | 0,00001016 | 39 | 0,19307639 |
| 22 | 0,00003712 | 40 | 1,74220610 |

Таблица 3. Максимальные погрешности интерполирования.

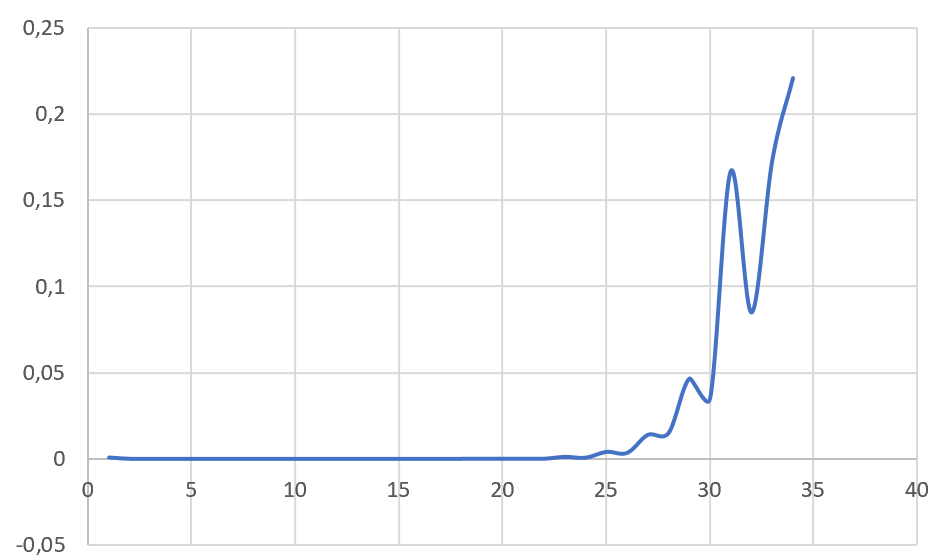


Рисунок 3. График погрешностей при количестве узлов = 35

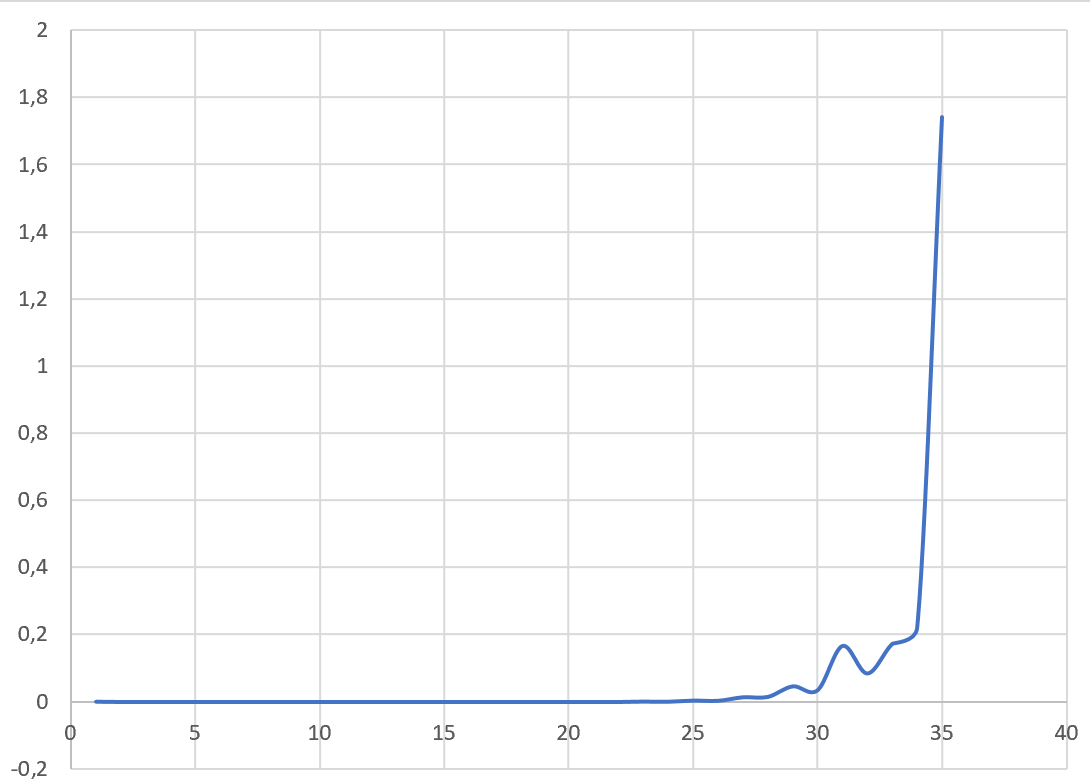


Рисунок 4. График погрешностей при количестве узлов = 40

Исследуя таблицу и график, можно заметить, что при количестве узлов n=40 наблюдается значительное усиление погрешности. А на количестве узлов от 9 до 16 погрешность, напротив, минимальна, поэтому такое начальное количество узлов является лучшим для интерполирования.

4) Повторим пункты 1-3 для узлов Чебышева.

Построим интерполяционный полином Лагранжа для 11 узлов Чебышева, подставим найденные значения в формулу и вычислим погрешности для данных узлов. Получим следующую таблицу значений (Таблица 5):

|  |  |  |  |
| --- | --- | --- | --- |
| 𝑥 | 𝑆𝑖(𝑥) | 𝐿𝑛(𝑥) |  |
| 0,1522 | 0.11372806 | 0,1135596 | 0.00016846 |
| 0,4132 | 0,38646146 | 0,38630287 | 0.00016159 |
| 0,7824 | 0,82649201 | 0,82634635 | 0.00014566 |
| 1,2346 | 1,11803915 | 1,11792063 | 0.00011852 |
| 1,7389 | 1,39912692 | 1,39905476 | 0.00007610 |
| 2,2610 | 1,55239228 | 1,55237505 | 0.00001723 |
| 2,7653 | 1,80359610 | 1,80354143 | 0.00005467 |
| 3,2175 | 1,81143041 | 1,81129938 | 0.00013103 |
| 3,5867 | 1,88194405 | 1,88173535 | 0.00020087 |
| 3,8477 | 0.22989878 | 0.99326 | 0.00025564 |

Таблица 5. Значения полинома Лагранжа и его погрешностей.

После выполненной работы для узлов Чебышева, можно пронаблюдать, что значения погрешностей имеют более плавный характер, без резких перепадов и сами значения погрешностей меньше, чем при равнораспределённых узлах (рисунок 6).

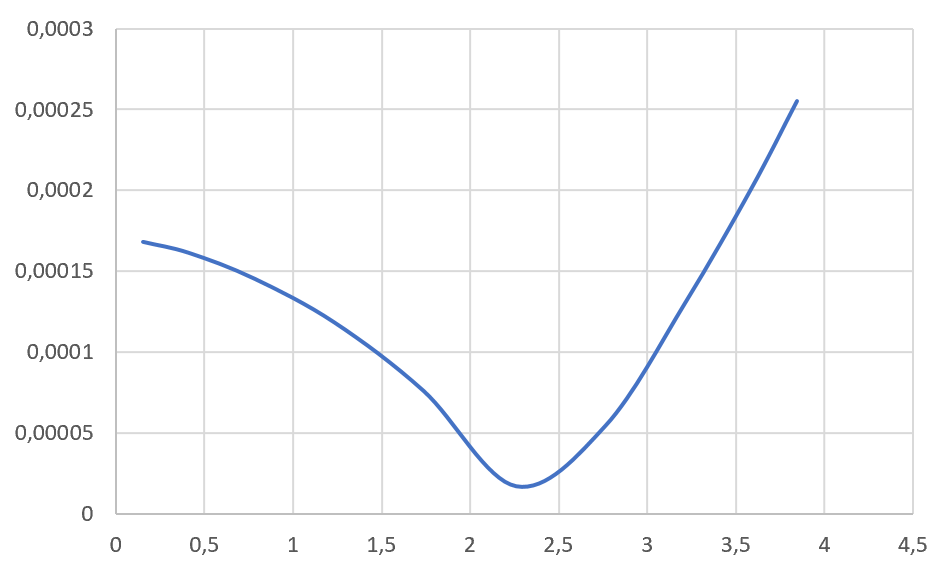


Рисунок 6. График погрешностей для 11 узлов Чебышева

Увеличим число узлов интерполяции и проследим, как меняется максимальная погрешность в случае узлов Чебышева. В таблице 6 в соответствии с числом узлов определена максимальная погрешность.

|  |  |  |  |
| --- | --- | --- | --- |
| 𝑛 | ε𝑛(𝑥) | 𝑛 | ε𝑛(𝑥) |
| 5 | 0,00528320 | 23 | 0,00000015 |
| 6 | 0,00004676 | 24 | 0,000000019 |
| 7 | 0,00011198 | 25 | 0,000000055 |
| 8 | 0,00000171 | 26 | 0,000000037 |
| 9 | 0,00000133 | 27 | 0,000000156 |
| 10 | 0,00000003 | 28 | 0,000000105 |
| 11 | 0,00000003 | 29 | 0,000000167 |
| 12 | 0,00000001 | 30 | 0,000000197 |
| 13 | 0,00000009 | 31 | 0,000000016 |
| 14 | 0,00000002 | 32 | 0,000000017 |
| 15 | 0,00000002 | 33 | 0,000000122 |
| 16 | 0,00000002 | 34 | 0,000000125 |
| 17 | 0,00000005 | 35 | 0,000000059 |
| 18 | 0,00000008 | 36 | 0,000000021 |
| 19 | 0,00000004 | 37 | 0,000000031 |
| 20 | 0,00000001 | 38 | 0,000000078 |
| 21 | 0,00000012 | 39 | 0,000000074 |
| 22 | 0,00000008 | 40 | 0,000000042 |

Таблица 6. Погрешности в различных узлах Чебышева.

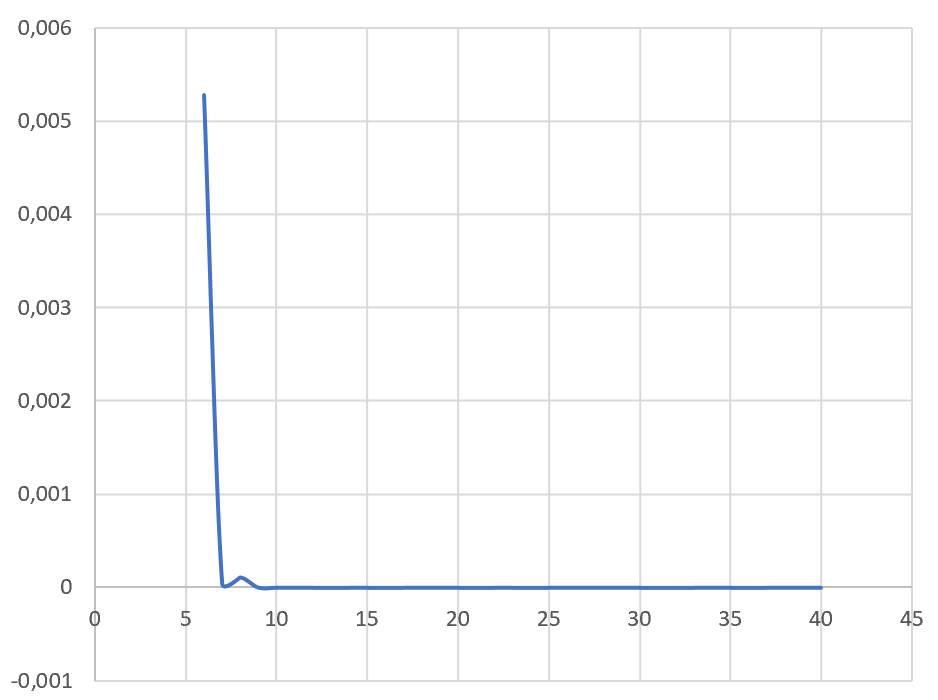


Рисунок 7. График погрешности для различного количества узлов Чебышева (от 6 до 40)

Можем заметить, что погрешности уменьшаются с увеличением количества узлов.

# ВЫВОДЫ

Итак, в ходе проделанной работы была протабулирована функция ошибок и построен полином Лагранжа на основе равномерно распределенных узлов и узлов Чебышева.

Вычисление и анализ значений функции и полинома Лагранжа привело нас к тому, что в точках интерполяции эти величины совпадают, погрешность в таком случае равна нулю.

В ходе эксперимента, в котором мы увеличивали количество равнораспределенных узлов, был сделан вывод о том, что максимальная погрешность начинает неравномерно возрастать при увеличении количества узлов. В нашем случае, этот процесс начинается с n ~ 30.

Повторение аналогичных действий с узлами Чебышева ведет к иному результату. При малом количестве таких узлов погрешность довольно большая, но с их увеличением она стремится к нулю.

Рисунок 8 наглядно демонстрирует различия между графиками погрешностей, относящимся к узлам, равномерно распределенным и узлам Чебышева.

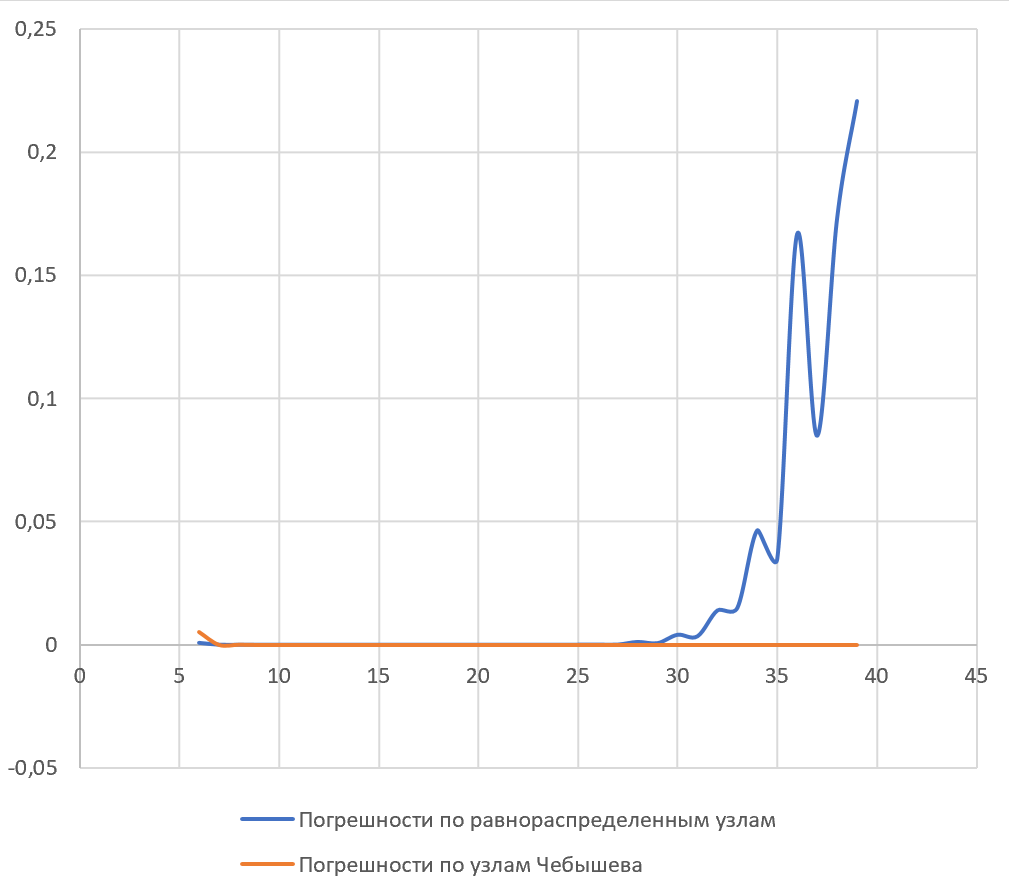


Рисунок 8. Графики погрешностей для различного количества узлов (равнораспределённых и чебышевских)

Так, можно сделать вывод о том, что узлы Чебышева намного лучше подходят для вычисления значения функции и построения полинома Лагранжа при любом количестве узлов поскольку значение погрешности уменьшает с ростом количества узлов. Однако, равнораспреденные узлы показывают хорошие результаты при небольшом количестве узлов, однако с их увеличением заметно возрастает.

# ЛИСТИНГ

#include <iostream>

#include <fstream>

#include <cmath>

#include <vector>

#include <iomanip>

void Sum(double x, double &s1) {

    double eps = 0.000001;

    double s = x;

    double dr = x;

    int i = 1;

    s1 = x;

    do {

        s = s1;

        dr = -1 \* dr \* x \* x / (2 \* i \* (2 \* i + 1));

        s1 = s + dr / (2 \* i + 1);

        i++;

    } while (std::abs(s1 - s) > eps);

}

void Interpolation(double x1, std::vector<double> &x, std::vector<double> &si, int n, double &l) {

    std::vector<double> x\_(n);

    std::vector<double> si\_(n);

    for (int i = 0; i < n; i++) {

        x\_[i] = x[i \* 2];

        si\_[i] = si[i \* 2];

    }

    l = 0;

    for (int i = 0; i < n; i++) {

        double p = 1;

        for (int j = 0; j < n; j++) {

            if (j != i) {

                p = p \* ((x1 - x\_[j]) / (x\_[i] - x\_[j]));

            }

        }

        l += si\_[i] \* p;

    }

}

void Error(std::vector<double> &a, std::vector<double> &b, int n, std::vector<double> &e) {

    for (int i = 0; i < n; i++) {

        e[i] = std::abs(a[i] - b[i]);

    }

}

void CalculateInterpolationAndError(int m, std::ofstream &file, std::vector<double> &max\_errors, std::vector<double> &max\_errors\_cb) {

    int a = 0;

    int b = 4;

    int n = 2 \* m + 1;

    std::vector<double> x(n);

    std::vector<double> si(n);

    std::vector<double> pol(n);

    x[0] = 0;

    si[0] = 0;

    pol[0] = 0;

    std::vector<double> err(n);

    std::vector<double> x\_cb(n);

    std::vector<double> si\_cb(n);

    std::vector<double> pol\_cb(n);

    x\_cb[0] = 0;

    si\_cb[0] = 0;

    for (int i = 1; i < n; i++) {

        double s, s\_cb;

        x[i] = i \* (b - a) / static\_cast<double>(2 \* m);

        x\_cb[i] = (b - a) / 2 \* std::cos(std::acos(-1.0) \* (2 \* i + 1) / (2 \* n + 2)) + (a + b) / 2;

        Sum(x[i], s);

        Sum(x\_cb[i], s\_cb);

        file << "Si(" << x[i] << ") = " << s << std::endl;

        file << "Si\_cb(" << x\_cb[i] << ") = " << s\_cb << std::endl;

        si[i] = s;

        si\_cb[i] = s\_cb;

    }

    file << std::endl;

    for (int i = 0; i < n; i++) {

        double l, l\_cb;

        Interpolation(x[i], x, si, m + 1, l);

        Interpolation(x\_cb[i], x\_cb, si\_cb, m + 1, l\_cb);

        file << "L(" << x[i] << ") = " << l << std::endl;

        file << "L\_cb(" << x\_cb[i] << ") = " << l\_cb << std::endl;

        pol[i] = l;

        pol\_cb[i] = l\_cb;

    }

    file << std::endl;

    Error(si, pol, n, err);

    double max = err[0];

    for (int i = 0; i < n; i++) {

        file << "S(" << x[i] << ") - L(" << x[i] << ") = " << err[i] << std::endl;

        if (err[i] > max) {

            max = err[i];

        }

    }

    file << std::fixed << std::setprecision(10) << max << "\t" << std::endl;

    Error(si\_cb, pol\_cb, n, err);

    double max\_cb = err[0];

    for (int i = 0; i < n; i++) {

        file << "S\_cb(" << x\_cb[i] << ") - L\_cb(" << x\_cb[i] << ") = " << err[i] << std::endl;

        if (err[i] > max\_cb) {

            max\_cb = err[i];

        }

    }

    file << std::fixed << std::setprecision(10) << max\_cb << "\t" << std::endl;

    max\_errors.push\_back(max);

    max\_errors\_cb.push\_back(max\_cb);

    if (m % 5 == 0) {

        file << m << "\t" << max << "\t" << n << "\t" << max\_cb << std::endl;

    }

}

void OutputResults(std::vector<double> &max\_errors, std::vector<double> &max\_errors\_cb) {

    std::cout << "Max Errors:" << std::endl;

    for (size\_t i = 0; i < max\_errors.size(); i++) {

        std::cout << "Regular " << i + 6 << ": " << std::fixed << std::setprecision(10) << max\_errors[i]

                  << "\tChebyshev " << i + 6 << ": " << std::fixed << std::setprecision(10) << max\_errors\_cb[i] << std::endl;

    }

}

int main() {

    std::ofstream f("Points.txt", std::ios::out);

    std::vector<double> max\_errors;

    std::vector<double> max\_errors\_cb;

    for (int m = 5; m < 36; m++) {

        CalculateInterpolationAndError(m, f, max\_errors, max\_errors\_cb);

    }

    OutputResults(max\_errors, max\_errors\_cb);

    f.close();

    return 0;