

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
Московский государственный технический университет им. Н.Э. Баумана
Факультет «Фундаментальные науки»

Лабораторная работа №4
по курсу «Вычислительная физика»
Тема: «Интерполяция сплайнами»
Вариант 6

Выполнили: студенты группы ФН4-72Б
Мистрюкова Л.А., Хижик А.И.
Проверил: доцент, к.физ.-мат.н.
Хасаншин Р.Х.

Москва, 2019

Оглавление

1.	Теоретическая часть	3
1.1.	Введение	3
1.2.	Кубический сплайн	3
1.3.	Способы задания наклонов интерполяционного кубического сплайна	7
1.4.	Погрешность приближения сплайном	8
2.	Алгоритм	9
2.1.	Способ А	9
2.2.	Способ В	9
3.	Постановка задачи	10
4.	Программа	11
5.	Результаты	14
5.1.	Способ А	14
5.2.	Способ В	15
6.	Вывод	16

1. Теоретическая часть

1.1. Введение

Пусть задана сетка $\{x_n, 0 \leq n \leq N\}$; точки x_n – узлы сплайна. Полиномиальным сплайном $S_p(x)$ дефекта q называется функция, удовлетворяющая следующим требованиям:

1. $S_p(x)$ на каждом интервале $[x_{n-1}, x_n]$ является полиномом степени p ;
2. Эти полиномы “склеены” во внутренних узлах так, что сплайн остаётся непрерывным вместе со своими $p - q$ производными: $S_p^{(k)}(x_n - 0) = S_p^{(k)}(x_n + 0)$, $0 \leq k \leq p - q$, $1 \leq n \leq N - 1$.

Чаще всего ограничиваются сплайнами дефекта $q = 1$, когда разрывна лишь старшая (p -я) производная, а все младшие производные непрерывны. В этом случае говорят просто о сплайне степени p , опуская упоминание о дефекте.

Если сплайн в заданных точках совпадает с табулированной функцией, то такой сплайн называется интерполяционным.

Примеры интерполяционных сплайнов: ломаная, проведённая через заданные точки, состоит из отрезков прямых. В узлах она непрерывна, но первая производная разрывна. Значит, это сплайн степени $p = 1$ дефекта $q = 1$. Кубический интерполяционный многочлен Эрмита склеен из кубических многочленов, а в узлах непрерывен вместе с первой производной. Это сплайн степени $p = 3$ дефекта $q = 2$.

1.2. Кубический сплайн

В этом случае на каждом интервале интерполяционный сплайн является многочленом третьей степени. Его удобно записать в следующем виде:

$$S_{3n}(x) = a_n + b_n(x - x_{n-1}) + c_n(x - x_{n-1})^2 + d_n(x - x_{n-1})^3, \quad x_{n-1} < x < x_n, \quad 1 \leq n \leq N, \quad (1)$$

где коэффициенты a_n, b_n, c_n, d_n свои для каждого интервала. Эти коэффициенты определяют из условий в узлах. Очевидно, каждый многочлен (1) в своем правом и левом узлах должен совпадать с табулированными значениями функции:

$$S_{3n}(x_{n-1}) \equiv a_n = u_{n-1}, \quad 1 \leq n \leq N, \quad (2)$$

$$S_{3n}(x_n) = a_n + b_n h_n + c_n h_n^2 + d_n h_n^3 = u_n. \quad (3)$$

Число этих уравнений вдвое меньше числа неизвестных коэффициентов, поэтому для определённости задачи нужны дополнительные условия. Для их получения вычислим первую и вторую производные многочлена (1):

$$S'_{3n}(x) = b_n + 2c_n(x - x_{n-1}) + 3d_n(x - x_{n-1})^2, \quad x_{n-1} < x < x_n, \quad 1 \leq n \leq N,$$

$$S''_{3n}(x) = 2c_n + 6d_n(x - x_{n-1}), \quad x_{n-1} < x < x_n, \quad 1 \leq n \leq N,$$

потребуем гладкости сплайна во всех точках, включая узлы. Приравняв во внутреннем узле x_n правые и левые пределы производных, получим

$$b_{n+1} = b_n + 2c_n h_n + 3d_n h_n^2, \quad 1 \leq n \leq N-1; \quad (4)$$

$$c_{n+1} = c_n + 3d_n h_n, \quad 1 \leq n \leq N. \quad (5)$$

Здесь введен дополнительный коэффициент c_{N+1} , имеющий смысл $0.5S_3''(x_N)$. Поэтому число уравнений (4) на одно больше, чем (5), а общее число неизвестных коэффициентов увеличилось на единицу.

Равенства (2-5) образуют систему $4N-1$ уравнений для $4N+1$ неизвестных коэффициентов.

Преобразование уравнений

Приведем систему уравнений к виду, содержащему только коэффициенты c_n . Уравнение (2) даёт нам коэффициенты a_n . Из уравнений (5) следует

$$d_n = \frac{c_{n+1} - c_n}{3h_n}, \quad 1 \leq n \leq N. \quad (6)$$

Подставим (6) в (3), одновременно исключив оттуда $a_n = u_{n-1}$; получим

$$b_n = \frac{u_n - u_{n-1}}{h_n} - \frac{h_n(c_{n+1} + 2c_n)}{3}, \quad 1 \leq n \leq N. \quad (7)$$

Исключим из (4) величины b_n и b_{n+1} с помощью (7), соответственно увеличивая во втором случае индекс на единицу, используя (6) для d_n . Остаётся система линейных уравнений для коэффициентов c_n , легко приводящаяся к следующему виду:

$$h_{n-1}c_{n-1} + 2(h_{n-1} + h_n)c_n + h_n c_{n+1} = 3 \left[\frac{u_n - u_{n-1}}{h_n} - \frac{u_{n-1} - u_{n-2}}{h_{n-1}} \right], \quad 2 \leq n \leq N. \quad (8)$$

Это система $N-1$ уравнения для $N+1$ неизвестных коэффициентов c_n . Уравнения имеют трехдиагональный вид.

Кубический сплайн похож на кубический интерполяционный многочлен, поэтому следует ожидать от него точности $O(h^4)$. Если табулированы только значения функции, то дополнительные условия приходится подбирать так, чтобы не испортить этого порядка точности. Здесь возникают две различные ситуации: $u(x)$ может быть периодической либо непериодической.

Периодические граничные условия

Рассмотрим периодическую и достаточно гладкую $u(x)$, для которой отрезок $[x_0, x_N]$ является периодом. Тогда на обоих концах отрезка функция и ее производные принимают одинаковые значения: $u^{(q)}(x_0) = u^{(q)}(x_N)$, $q = 0, 1, \dots$ (при этом обязательно надо проверять исходную информацию: если $u_0 \neq u_n$, то данные ошибочны). Естественно также требовать от сплайна, чтобы он

удовлетворял аналогичным условиям:

$$S_{31}^{(q)}(x_0 + 0) = S_{3N}^{(q)}(x_N - 0), \quad q = \overline{0, 2}. \quad (9)$$

Требовать непрерывности третьей производной мы уже не можем. При $q = 0$ условие (9) автоматически следует из условия интерполяции. Значения $q = 1, 2$ дают два недостающих уравнения

$$b_1 = b_N + 2c_N h_N + 3d_N h_N^2,$$

$$c_1 = c_{N+1}.$$

Исключив отсюда b_1, b_N, d_N с помощью (6), (7), получим граничные условия в следующем виде:

$$(2c_1 + c_2)h_1 + (c_N + 2c_{N+1})h_N = 3 \left[\frac{u_1 - u_0}{h_1} - \frac{u_N - u_{N-1}}{h_N} \right], \quad (10)$$

$$c_1 - c_{N+1} = 0.$$

Заметим, что первое из этих уравнений является циклическим замыканием цепочки (8). Дополнив условиями (10) уравнения (8), получим систему $N + 1$ линейных уравнений для определения $N + 1$ коэффициентов c_n . Структура матрицы этой системы приведена на рис. 1.

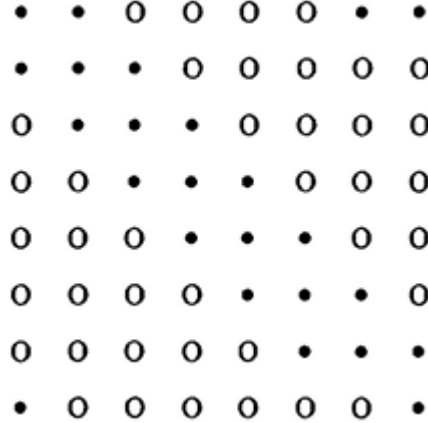


Рис. 1. Структура матрицы для нахождения коэффициентов сплайна для периодических граничных условий

Эта матрица имеет ненулевые три диагонали и три угловых элемента. Соответствующая система линейных уравнений легко решается методом Гаусса без выбора главного элемента с обходом нулей, или модификацией так называемой циклической прогонки. Диагональные элементы матрицы преобладают, поэтому алгоритм очень устойчив.

Найдя коэффициенты c_n , оставшиеся коэффициенты сплайна вычисляются по формулам (2), (3) и (7).

Если периодическая $u(x)$ имеет непрерывную четвертую производную, то на равномерной сетке можно получить строгую оценку погрешности:

$$|u(x) - S_3(x)| \leq \frac{1}{384} M_4 h^4, \quad M_4 = \max |u^{(IV)}(x)|. \quad (11)$$

Численный коэффициент в этой оценке очень мал. Точно такую же погрешность имеет локальный кубический интерполяционный многочлен Эрмита, но для его построения нужно также табулировать в узлах производные u'_n , а для сплайна этого не требуется. Это наглядно демонстрирует преимущество сплайна.

Естественные граничные условия

Пусть $u(x)$ непериодическая функция. Наиболее оптимальным дополнительным условием является вариационное условие. Надо минимизировать разрывы третьих производных сплайна во внутренних узлах в смысле метода наименьших квадратов. Разрыв $S_3'''(x)$ в узле x_n пропорционален $d_{n+1} - d_n$. Наилучшие результаты даёт вариационное условие

$$\sum_{n=1}^{N-1} \frac{(d_{n+1} - d_n)^2}{h_n + h_{n+1}} = \frac{1}{3} \sum_{n=1}^{N-1} \frac{\frac{c_{n+2} - c_{n+1}}{h_{n+1}} - \frac{c_{n+1} - c_n}{h_n}}{h_n + h_{n+1}} \rightarrow \min. \quad (12)$$

Условие (12) вместе с уравнениями (8) образует задачу на условный экстремум. Ее можно решить методом неопределённых множителей Лагранжа. Но это приводит к достаточно сложному алгоритму.

В частном случае, когда вместо всей суммы (12) оставляем только крайние слагаемые с $n = l$ и $n = N - 1$, что эквивалентно требованию непрерывности $S_3'''(x)$ в приграничных узлах x_1, x_{N-1} . Это даёт $d_1 = d_2$ и $d_{N-1} = d_N$. Выразив коэффициенты d_n через c_n по (6), получим соотношения

$$\frac{c_2 - c_1}{h_1} - \frac{c_3 - c_2}{h_2} = 0. \quad (13)$$

$$\frac{c_{N+1} - c_N}{h_N} - \frac{c_N - c_{N-1}}{h_{N-1}} = 0.$$

Соотношения (13) и (8) образуют систему линейных уравнений для коэффициентов c_n . Ее матрица отличается от трехдиагональной только первой и последней строкой, ее вид представлен на рис. 2. Такая система легко решается методом Гаусса для ленточной матрицы без выбора главного элемента, так как диагональные элементы преобладают. Этот алгоритм очень устойчив. После нахождения коэффициентов c_n остальные коэффициенты выражаются через них так же, как коэффициенты периодического сплайна. Погрешность естественного сплайна есть $O(h^4)$. На некотором расстоянии

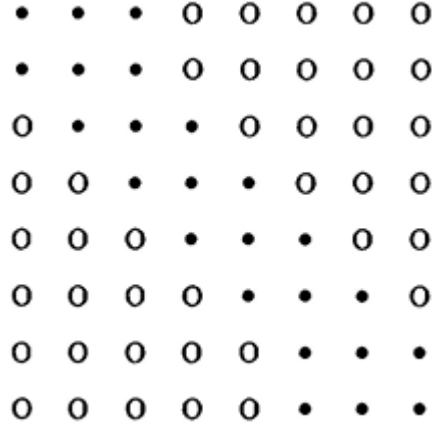


Рис. 2. Структура матрицы для нахождения коэффициентов сплайна для естественных граничных условий

от граничных интервалов при этом справедлива оценка (11). Вблизи границы оценка аналогична, но с несколько большим численным коэффициентом. Этот коэффициент быстро стремится к $\frac{1}{384}$ по мере удаления от границы.

1.3. Способы задания наклонов интерполяционного кубического сплайна

А. Полагают $m_i = \frac{f_{i+1}-f_{i-1}}{2h}$, $i = \overline{1, N-1}$, $m_0 = \frac{-3f_0+4f_1-f_2}{2h}$, $m_N = \frac{3f_N-4f_{N-1}+f_{N-2}}{2h}$.

В. Если известны значения $f'_i = f'(x_i)$, то полагают $m_i = f'_i$, $i = \overline{0, N}$.

С. Пусть $S''_3(x_i + 0)$ [$S''_3(x_i - 0)$] – значение $S''_3(x_i)$ в узле x_i справа [слева], найденное из $m_i = \frac{f_{i+1}-f_{i-1}}{2h}$, $i = \overline{0, N-2}$. Тогда

$$S''_3(x_i + 0) = -\frac{4}{h}m_i - \frac{2}{h}m_{i+1} + 6\frac{f_{i+1} - f_i}{h^2},$$

$$S''_3(x_i - 0) = \frac{4}{h}m_i + \frac{2}{h}m_{i-1} - 6\frac{f_i - f_{i-1}}{h^2}.$$

Потребуем непрерывность $S''_3(x)$ в узлах:

$$S''_3(x_i + 0) = S''_3(x_i - 0), \quad i = \overline{1, N-1}.$$

Получим следующую систему уравнений относительно наклонов:

$$m_{i-1} + 4m_i + m_{i+1} = \frac{3(f_{i+1} - f_{i-1})}{h}, \quad i = \overline{1, N-1}.$$

Для доопределения системы необходимы краевые условия. Рассмотрим некоторые из них.

- 1) Если известны $f'_0 = f'(a)$ и $f'_N = f'(b)$, то задают $m_0 = f'_0$ и $m_N = f'_N$.
- 2) Производные f'_0 и f'_N аппроксимируют формулами численного дифференцирования третьего порядка точности (без остаточных членов) и полагают

$$m_0 = \frac{-11f_0 + 18f_1 - 9f_2 + 2f_3}{6h},$$

$$m_N = \frac{11f_N - 18f_{N-1} + 9f_{N-2} - 2f_{N-3}}{6h}.$$

3) В некоторых случаях бывают известны значения второй производной на концах отрезка $[a, b]$, т.е. величины $f_0'' = f''(a)$ и $f_N'' = f''(b)$. Тогда требования $S_3''(a) = f''(a)$ и $S_3''(b) = f''(b)$ приводят к краевым условиям

$$m_0 = -\frac{m_1}{2} + \frac{3}{2} \frac{f_1 - f_0}{h} - \frac{h}{4} f_0'',$$

$$m_N = -\frac{m_{N-1}}{2} + \frac{3}{2} \frac{f_N - f_{N-1}}{h} + \frac{h}{4} f_N''.$$

Краевые условия (1-3) можно комбинировать, т.е. в левом и правом крайних узлах выбирать их независимо. Система уравнений относительно наклонов при всех рассмотренных краевых условиях имеет единственное решение, для нахождения которого могут быть применены методы прогонки и итераций. Решая эту систему при выбранных краевых условиях находим наклоны во всех узлах и задаём сплайн на каждом частичном отрезке $[x_i, x_{i+1}]$, $i = \overline{0, N-1}$. Построенный глобальным способом сплайн $S_3(x)$ имеет дефект не больше единицы, так как этот сплайн обладает на отрезке $[a, b]$ непрерывной второй производной $S_3''(x)$.

1.4. Погрешность приближения сплайном

Утверждение. Если $f \in C_{k+1}[a, b]$, $0 \leq k \leq 3$, то интерполяционный сплайн $S_3(x)$ с наклонами заданными способами В) или С), удовлетворяет неравенству

$$\max_{[x_i, x_{i+1}]} |f^{(m)}(x) - S_3^{(m)}(x)| \leq ch^{k+1-m} \max_{[a, b]} |f^{(k+1)}(x)|. \quad (14)$$

где $i = \overline{0, N-1}$, $m = \overline{0, k}$, с независимая от шага h_i и f постоянная.

Если $k = 3$, и наклоны сплайна $S_3(x)$ найдены глобальным способом, то на $[a, b]$ максимальные по модулю отклонения $S_3(x)$ от $f(x)$, $S_3'(x)$ от $f'(x)$ и $S_3''(x)$ от $f''(x)$ равны соответственно $O(h^4)$, $O(h^3)$, и $O(h^2)$. При задании наклонов способом В) имеющие в узлах скачки у $S_3''(x)$ не превышают $\frac{f_{i+1} - f_{i-1}}{h}$, $i = \overline{0, N-1}$, т.е. при $k = 3$ будут порядка $O(h^2)$. Эти скачки у второй производной на графике заметить трудно.

Замечание 1. Для кубического сплайна $S_3(x)$, наклоны которого заданы упрощённым способом А) утверждение справедливо при $0 \leq k \leq 2$.

Замечание 2. Если $f \in C_{k+1}(-\infty, +\infty)$, $0 \leq k \leq 3$ и имеет период $b - a$, то следует положить $m_0 = m_N$ и к системе уравнений относительно наклонов присоединить уравнение $m_{N-1} + 4m_0 + m_1 = \frac{3(f_1 - f_{N-1})}{h}$, отвечающее значению $i = 0$. Из расширенной системы, имеющей единственное решение, находятся наклоны сплайна $S_3(x) \in C_2[a, b]$; этот сплайн продолжается с отрезка $[a, b]$ на всю ось с сохранением непрерывности второй производной. При этом справедливо (14).

На больших промежутках (при больших N), сплайны являются более удобным средством для аппроксимации функций, чем интерполяционные многочлены.

Кубический сплайн $S_3(x)$, наклоны которого заданы глобальным способом, дважды непрерывно дифференцируем на всем отрезке $[a, b]$, т.е. имеет непрерывную кривизну.

Точность аппроксимации функции сплайном $S_3(x)$ управляется выбором N , т.е. шагом $h = \frac{b-a}{N}$.

2. Алгоритм

2.1. Способ А

1. Вводим на $[a, b]$ равномерную сетку с шагом h : $\{t_i\}_{i=0}^n$, где $t_i = t_0 + ih$, $t_i \in [a, b]$, $i = \overline{0, n}$.
2. Вычисляем: $\{f_i\}_{i=0}^n$, где $f_i = f(t_i)$.
3. Вычисляем значения наклонов интерполяционного кубического сплайна в узлах сетки:

$$m_i = \frac{f_{i+1} - f_{i-1}}{2h}, \quad i = \overline{1, n-1},$$

$$m_0 = \frac{4f_1 - f_2 - 3f_0}{2h}, \quad m_N = \frac{3f_n - 4f_{n-1} + f_{n-2}}{2h}.$$

4. Строим на каждом частичном отрезке $[x_i, x_{i+1}]$, $i = \overline{0, n-1}$ кубический сплайн:

$$\begin{aligned} s_i(x) = & \frac{(x_{i+1} - x)^2[2(x - x_i) + h]}{h^3} f_i + \frac{(x - x_i)^2[2(x_{i+1} - x) + h]}{h^3} f_{i+1} + \\ & + \frac{(x_{i+1} - x)^2(x - x_i)}{h^2} m_i + \frac{(x - x_i)^2(x - x_{i+1})}{h^2} m_{i+1}. \end{aligned}$$

2.2. Способ В

1. Вводим на $[a, b]$ равномерную сетку с шагом h : $\{t_i\}_{i=0}^n$, где $t_i = t_0 + ih$, $t_i \in [a, b]$, $i = \overline{0, n}$.
2. Вычисляем: $\{f_i\}_{i=0}^n$, где $f_i = f(t_i)$.
3. Вычисляем значения наклонов интерполяционного кубического сплайна в узлах сетки: $m_i = f'(x_i)$, $i = \overline{0, n}$.
4. Строим на каждом частичном отрезке $[x_i, x_{i+1}]$, $i = \overline{0, n-1}$ кубический сплайн:

$$\begin{aligned} s_i(x) = & \frac{(x_{i+1} - x)^2[2(x - x_i) + h]}{h^3} f_i + \frac{(x - x_i)^2[2(x_{i+1} - x) + h]}{h^3} f_{i+1} + \\ & + \frac{(x_{i+1} - x)^2(x - x_i)}{h^2} m_i + \frac{(x - x_i)^2(x - x_{i+1})}{h^2} m_{i+1}. \end{aligned}$$

3. Постановка задачи

- Построить графики интерполяционных многочленов приближающих функцию Рунге $f(x) = \frac{1}{1+25x^2}$ на отрезке $[-1, 1]$, для разного числа частичных отрезков $N = 5, 8, 15, 16$. Сравнить с результатами лаб. работы №2. Использовать локальные способы А) и В) задания наклонов сплайна.

4. Программа

```
#include <iostream>
#include <math.h>
using namespace std;

class Spline3{
private:
    double* data_x;
    double* data_y;
    double* data_m;
    double* data_m2;
    unsigned size;
    double h;
    double left_bound, right_bound;
public:
    Spline3(double* d1, double* d2, unsigned s, double l, double r){
        size = s;
        data_x = new double[size];
        data_y = new double[size];
        data_m = new double[size];
        data_m2 = new double[size];
        left_bound = l;
        right_bound = r;

        double h = (right_bound - left_bound)/(double)((size) - 1);
        for(unsigned i = 0; i < s; i++)
            data_x[i]=d1[i];
        for(unsigned i = 0; i < s; i++)
            data_y[i]= d2[i];
        data_m[0] = (-3*data_y[0] + 4*data_y[1] - data_y[2])/(2*h);
        data_m[size - 1] = (3*data_y[size - 1] - 4*data_y[size - 2] + data_y[size - 3])/(2*h);
        for (int i = 0; i <= size - 1; i++)
            data_m2[i] = Diff_Runge(left_bound + h * i);
        for (int i = 1; i <= size - 2; i++)
            data_m[i] = (data_y[i + 1] - data_y[i - 1]) / (2 * h);
    }

    double VariantA(double x, int index);
    double VariantB(double x, int index);
    void Spline();
    double Diff_Runge(double t);
};

double Spline3::Diff_Runge(double t){return -(50.0 * t )/ pow((1.0 + 25.0 * pow(t, 2)), 2);}

double Spline3::VariantA(double x, int index){
    double h = (right_bound - left_bound)/(double)((size) - 1);
    double s1, s2, s3, s4;
    double xi = data_x[index];
```

```

    double xii = data_x[index + 1];
    s1 = (pow((xii - x), 2) * (2 * (x - xi) + h)) * data_y[index] / (double)pow(h, 3);
    s2 = (pow((x - xi), 2) * (2 * (xii - x) + h)) * data_y[index + 1] / (double)pow(h, 3);
    s3 = pow((xii - x), 2) * (x - xi) * data_m[index] / (double)pow(h, 2);
    s4 = pow((x - xi), 2) * (x - xii) * data_m[index + 1] / (double)pow(h, 2);
    return s1 + s2 + s3 + s4;
}

double Spline3::VariantB(double x, int index){
    double h = (right_bound - left_bound)/(double)((size) - 1);
    double s1, s2, s3, s4;
    double xi = data_x[index];
    double xii = data_x[index + 1];
    s1 = (pow((xii - x), 2) * (2 * (x - xi) + h)) * data_y[index] / (double)pow(h, 3);
    s2 = (pow((x - xi), 2) * (2 * (xii - x) + h)) * data_y[index + 1] / (double)pow(h, 3);
    s3 = pow((xii - x), 2) * (x - xi) * data_m2[index] / (double)pow(h, 2);
    s4 = pow((x - xi), 2) * (x - xii) * data_m2[index + 1] / (double)pow(h, 2);
    return s1 + s2 + s3 + s4;
}

void Spline3::Spline(){
    cout << "Qubic Spline: " << endl;
    double h = (right_bound - left_bound)/(double)((size) - 1);
    cout << "Varriant A: " << endl;
    for(int i = 0; i <= size - 2; i++){
        for(int j = 0; j <= 15; j++){
            cout << VariantA(left_bound + h * i + h / 15.0 * j, i) << ", ";
        }
    }
    cout << endl;
    cout << "Variant B: " << endl;
    for(int i = 0; i <= size - 2; i++){
        for(int j = 0; j <= 15; j++){
            cout << VariantB(left_bound + h * i + h / 15.0 * j, i) << ", ";
        }
    }
    delete data_x;
    delete data_y;
    delete data_m;
    delete data_m2;
}

double Runge_Function(double t){return 1.0 / (1.0 + 25.0 * pow(t, 2));}

int main() {
    int n = 5;
    double left_bound = -1, right_bound = 1, dat_x[n], dat_y[n];
    for (int i = 0; i < n + 1; i++){
        dat_x[i] = left_bound + (double)i * (double)(right_bound - left_bound) / (double)(n);
    }
}

```

```
        dat_y[i] = Runge_Function(dat_x[i]);
    };
    Spline3 spline(dat_x, dat_y, n + 1, left_bound, right_bound);
    spline.Spline();

    return 0;
}
```

5. Результаты

5.1. Способ А

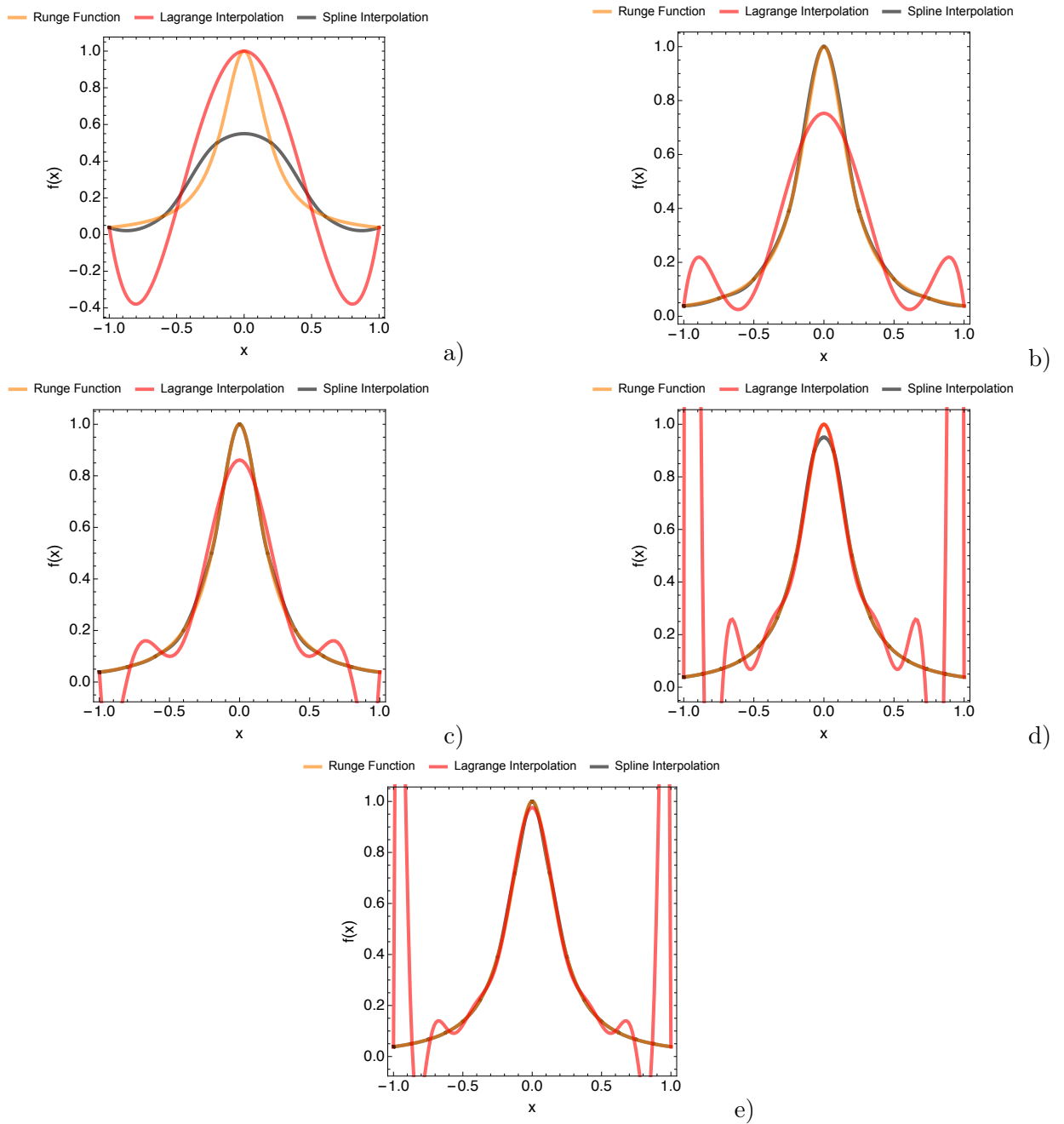


Рис. 3. Сравнение точности интерполяции интерполяционными многочленами Лагранжа и сплайнами для а) $N = 5$, б) $N = 8$, в) $N = 10$, г) $N = 15$, е) $N = 16$ частичных отрезков.

5.2. Способ В

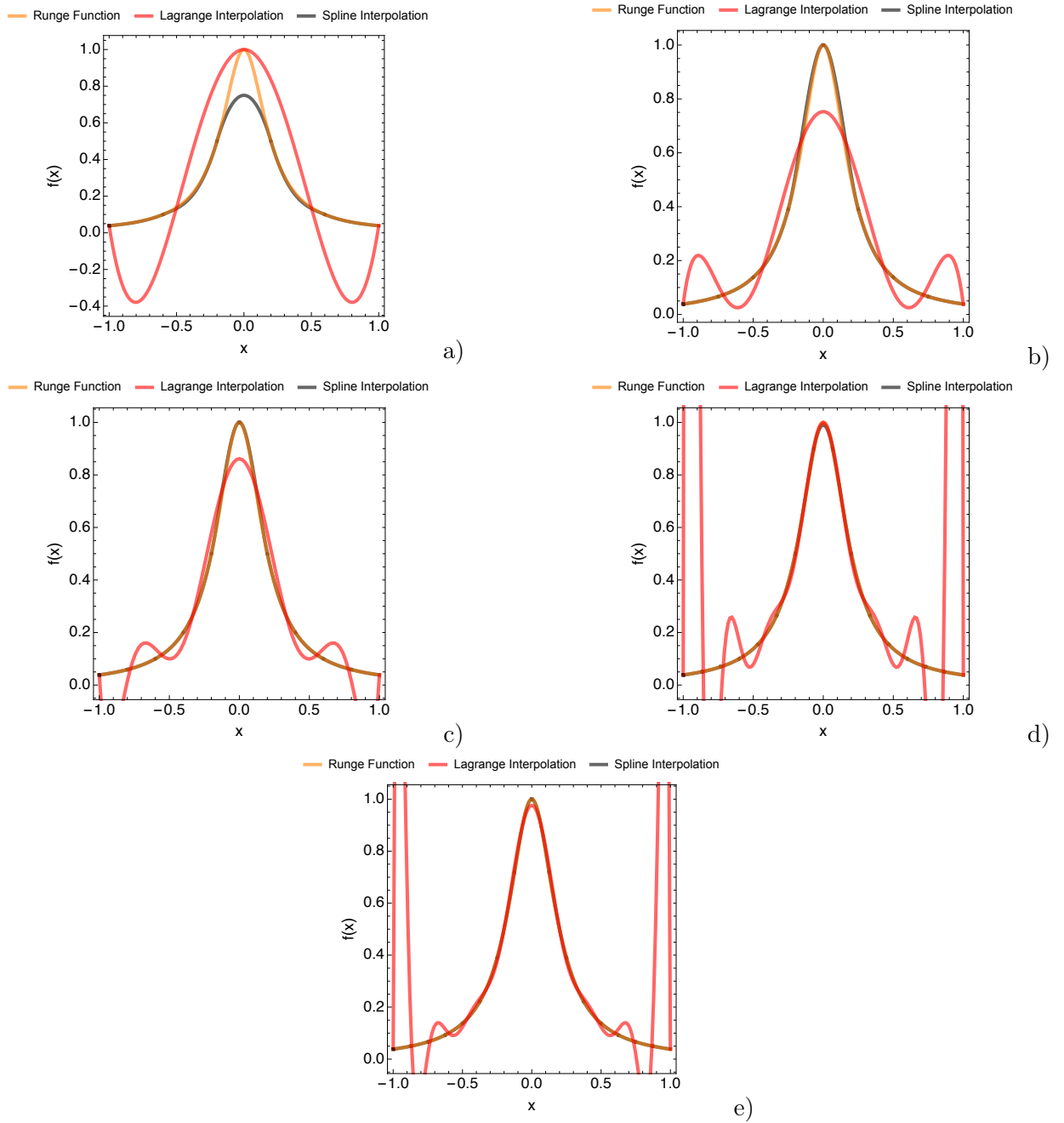


Рис. 4. Сравнение точности интерполяции интерполяционными многочленами Лагранжа и сплайнами для а) $N = 5$, б) $N = 8$, в) $N = 10$, г) $N = 15$, д) $N = 16$ частичных отрезков.

6. Вывод

Интерполяция сплайнами значительно точнее интерполяции полиномами Лагранжа, в случае сплайн-интерполяции наблюдается тенденция к уменьшения ошибки приближения с увеличением числа частичных отрезков.

Способ В) задания наклона сплайна показал большую точность аппроксимации, чем способ А).