

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по практической работе №5**  
**по дисциплине «Вычислительная математика»**  
**Тема: Аппроксимация функции с помощью кубического сплайна**

Студент гр. 8301

Готовский К.В.

Преподаватель

Сучков А.И

Колоницкий С.Б.

Санкт-Петербург

2020

## Цель работы.

Изучение и сравнение различных методов численного интегрирования на примере составных формул прямоугольников, трапеций и Симпсона.

## Основные теоретические положения.

Типичной задачей приближения функций является задача интерполяции: по заданной таблице чисел  $(x_i, f(x_i)), x_i \in [a, b], i = 1, 2, \dots, N$ , вычислить функцию  $f(x)$  с той или иной точностью на отрезке  $[a, b]$  действительной оси.

Классический метод решения этой задачи основан на построении интерполяционного многочлена Лагранжа, определяемого равенством:

$$L_N(x) = \sum_{i=1}^N \frac{f(x_i) l_N(x)}{(x-x_i) l'_N(x_i)}, l_N(x) = \prod_{j=1}^N (x - x_j). \quad (1)$$

Хотя, согласно теореме Вейерштрасса, всякая непрерывная на отрезке  $[a, b]$  функция  $f(x)$  может быть как угодно точно приближена многочленом, практическая возможность применения многочленов ограничена.

Ряд примеров показывает, что даже при большом числе узлов интерполяционный многочлен Лагранжа не гарантирует хорошее приближение функции. Например, С. Н. Бернштейном было установлено, что последовательность интерполяционных многочленов Лагранжа, построенных для непрерывной на отрезке  $[-1, 1]$  функции  $f(x) = |x|$  по равноотстоящим узлам, не сходится при возрастании числа узлов  $N$  к  $f(x)$ .

Еще более замечательный пример построен Рунге, который в качестве тестовой функции взял бесконечно дифференцируемую функцию

$$f(x) = \frac{1}{1+25x^2}$$

Однако, несмотря на ее гладкость, последовательность интерполяционных многочленов Лагранжа не сходилась к этой функции в равномерной норме. А именно, в обоих случаях было получено, что

$$\lim_{N \rightarrow \infty} \max_{x \in [-1, 1]} |f(x) - L_{20}(x)| > \infty$$

В частности, в примере Рунге установлено, что

$$\max_{x \in [-1, 1]} |f(x) - L_{20}(x)| > 58$$

Относительная погрешность такой интерполяции составляет величину около 10000%.

Иногда трудности построения хороших интерполяционных многочленов удается преодолеть за счет перехода к специальным многочленам или выбора подходящей системы узлов интерполяции.

Однако на практике, для того чтобы аппроксимировать функцию, вместо построения глобального интерполяционного многочлена на всем промежутке используют интерполяцию кусочными многочленами.

Простейшим примером такого рода является кусочно-линейная интерполяция. В общем случае отрезок  $[a, b]$  точками  $a = x_1 < x_2 < \dots < x_N = b$  разбивают на части и на каждом промежутке  $[x_i, x_{i+1}]$ ,  $i = 1, \dots, N$ , строят свой интерполяционный многочлен. Полученные таким образом многочлены (обычно одной степени) дают интерполяцию функции на всем отрезке. Такой интерполяционный многочлен-многозвенник может обладать различными свойствами, которые определяются степенью используемых многочленов на каждом отрезке  $[x_i, x_{i+1}]$  и условиями сопряжения на концах соседних отрезков. Полученные при этом гладкие кусочно-многочленные функции (составленные из многочленов одной и той же степени) называются сплайнами. Сплайн называется кубическим, если он составлен из многочленов третьей степени.

Гладкие сплайны обладают многими замечательными свойствами, которые и обеспечили им успех в приложениях. Так, для функции, рассмотренной в примере Рунге, кубический сплайн на сетке с шестью узлами дает погрешность настолько малую, что она не может быть изображена на одном графике с  $f(x)$ . Алгоритмы построения кубических сплайнов весьма просты и легко реализуются на ЭВМ.

Перейдем к построению кубических сплайнов класса  $C^2$  (являющихся дважды непрерывно дифференцируемыми функциями). Пусть на отрезке  $[a, b]$  в узлах сетки  $X = \{a = x_0 < x_1 < \dots < x_N = b\}$  заданы значения некоторой

функции  $f(x): f_i = f(x_i), i = 0, 1, 2, \dots, N$ . Интерполяционным кубическим сплайном  $S(x)$  называется сплайн, удовлетворяющий условиям:

$$S(x_i) = f_i, S'(x_i) = f'(x_i), S''(x_i) = f''(x_i). \quad (1)$$

Так как  $S(x)$  – кубический сплайн, то на каждом из отрезков  $[x_i, x_{i+1}]$  он определяется четырьмя коэффициентами, и поэтому для его построения на всем промежутке  $[a, b]$  необходимо определить  $4N$  коэффициентов. Условие принадлежности сплайна классу  $C^2[a, b]$  требует непрерывности во всех внутренних узлах интерполяции  $x_i, i = 2, \dots, N-1$ , не только сплайна  $S(x)$ , но и его производных  $S'(x)$  и  $S''(x)$ , что дает  $3(N-1)$  уравнений для определения неизвестных коэффициентов сплайна. Добавив  $N+1$  уравнение из (1), получим  $4N-2$  уравнения. Два недостающих уравнения обычно получают из ограничений на значение сплайна и его производных на концах промежутка  $[a, b]$  их называют краевыми условиями. Из множества различных видов краевых условий наиболее употребительными являются следующие:

$$S'(x_i) = f'(a), S'(b) = f'(b)$$

$$S''(a) = f''(a), S''(b) = f''(b).$$

$$S^{(k)}(a) = S^{(k)}(b), k = 1, 2.$$

$$S'''(x_p + 0) = S'''(x_p - 0), p = N-1.$$

Воспользуемся так называемыми «естественными» условиями, которые имеют вид

$$S''(a) = 0, S''(b) = 0.$$

Рассмотрим алгоритм построения интерполяционных кубических сплайнов. На каждом из промежутков  $[x_i, x_{i+1}]$  будем записывать сплайн не в общем виде

$$S(x) = a_i + b_i x^2 + d_i x^3,$$

а в некотором специальном виде, позволяющем уменьшить число неизвестных коэффициентов сплайна. Для этого введём обозначения

$$S'(x_i) = m_i, i = 0, 1, \dots, N,$$

где  $h_i = x_{i+1} - x_i$  и  $t = (x - x_i)/h_i$ .

На отрезке  $[x_i, x_{i+1}]$  кубический сплайн можно записать в виде

$$S(x) = f_i(1-t)^2(1+2t) + f_{i+1}t^2(3-2t) + m_i h_i t(1-t)^2 - m_{i+1} t^2(1+t) h_i. \quad (2)$$

Кубический сплайн, записанный в таком виде, на каждом из промежутков  $[x_i, x_{i+1}]$  автоматически непрерывен вместе со своей первой производной всюду на  $[a, b]$ . Выберем величины  $m_i$  таким образом, чтобы и вторая производная была непрерывна во всех внутренних узлах. Из этого условия получим следующую систему уравнений:

$$\lambda_i m_{i-1} + 2m_i + \mu_i m_{i+1} = 3 \left( \frac{\mu_i (f_{i+1} - f_i)}{h_i} + \frac{\lambda_i (f_i - f_{i-1})}{h_{i-1}} \right), \quad (3)$$

$$\text{где } \mu_i = \frac{h_{i-1}}{h_{i-1} + h_i}, \lambda_i = 1 - \mu_i = \frac{h_i}{h_{i-1} + h_i}.$$

К уравнениям (3) следует добавить уравнения, полученные из граничных условий. В рассматриваемом случае эти уравнения имеют вид

$$2m_0 + m_1 = \frac{3(f_1 - f_0)}{h_0}, m_{N-1} + m_N = \frac{3(f_N - f_{N-1})}{h_{N-1}}. \quad (4)$$

Таким образом, задача построения кубического сплайна свелась к решению линейной системы для неизвестных коэффициентов  $m_i$ :

$$\begin{aligned} 2m_0 + m_1 &= \frac{3(f_1 - f_0)}{h_0}, \\ \lambda_i m_{i-1} + 2m_i + \mu_i m_{i+1} &= 3 \left( \frac{\mu_i (f_{i+1} - f_i)}{h_i} + \frac{\lambda_i (f_i - f_{i-1})}{h_{i-1}} \right), \\ m_{N-1} + m_N &= \frac{3(f_N - f_{N-1})}{h_{N-1}}. \end{aligned}$$

Матрица полученной системы трехдиагональная, поэтому эту систему можно решить методом прогонки.

### Постановка задачи.

Вычислить значения заданной функции  $f(x)$  в узлах интерполяции  $x_i = a + h(i-1), i = 1, 2, \dots, N$ , на отрезке  $[a, b]$ . По вычисленной таблице  $\{x_i, f(x_i)\}$  построить интерполяционный кубический сплайн  $S(x)$ , вычислить его значения в промежуточных точках  $x_j = a + h/2 + h(j-1)$ .

Сравнить вычисленные значения с точными значениями функции в точках  $x_j$ . Построить график  $f(x)$  и  $S(x)$ .

1. Составить подпрограмму-функцию для вычисления значений  $f(x)$ .

2. Составить функцию SPLINE для построения кубического сплайна для интерполирования таблично заданной функции.
3. Составить главную программу. Главная программа должна: вычислить координаты узлов  $x_i$  и значения функции  $f(x)$  в них; с помощью программы SPLINE вычислить коэффициенты сплайна; вычислить значения функции  $f(x)$  и сплайна  $S(x)$  в промежуточных точках.
4. Произвести вычисления на ЭВМ.
5. Выбрав подходящий масштаб, изобразить графики  $S(x)$  и  $f(x)$ .

### Необходимые таблицы и графики расчётов программы.

Таблица 1 – Таблица значений для второго варианта.

x	y(x)	S(x)
-1.900000	-3.4079	1.4111
-1.700000	0.83821	1.2545
-1.500000	0.20828	0.40405
-1.300000	0.0054416	-0.027706
-1.100000	-0.12146	-0.10723
-0.900000	-0.21591	-0.21565
-0.700000	-0.27428	-0.26906
-0.500000	-0.27975	-0.27189
-0.300000	-0.21934	-0.21823
-0.100000	-0.090533	-0.061779
0.100000	0.11084	0.03797
0.300000	0.42655	0.75277
0.500000	1.0539	-0.005309
0.700000	3.3604	9.3239
0.900000	-13.246	0.81028

1.100000	-3.7859	-3.4158
1.300000	-3.4413	-2.8269
1.500000	-5.3755	-10.104
1.700000	-45.893	9.4881
1.900000	9.7936	11.796

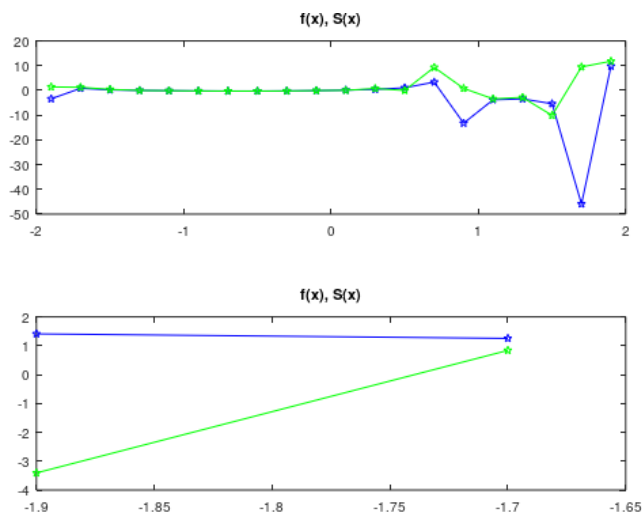


Рисунок 1 – Графики функций для второго варианта.

### Выводы.

Я изучил и сравнил различные методы численного интегрирования на примере составных формул прямоугольников, трапеций и Симпсона.

## ПРИЛОЖЕНИЕ А

### Программа

```
function [Xi,Yi,S] = Yrok_5

n = 20;
Left = -2;
Right = 2;
f = @(x) (e^x - cos(x))/(cos(x + x^2));
x = zeros(1, n + 1);
y = zeros(1, n + 1);

function [x, y, step] = CountFunction(f, x, y, Left, Right, n)
    step = (Right-Left)/n;
    k = 1;
    for i = Left:step:Right
        x(k) = i;
        y(k) = f(i);
        k++;
    endfor
endfunction

function [S, Xi, Yi, h, F, m] = Spline(x, y, n, step, Left, f)
    F = zeros(1, n + 1);
    h = gallery("tridiag", n + 1, 1/2, 2, 1/2);
    h(1, 2) = 1;
    h(n+1, n) = 1;
    for i = 2:n
        F(i) = 3*((y(i+1) - y(i))/2) - ((y(i) - y(i-1))/2));
    endfor
    m = F/h;
    for i = 1:n
        X = Left + step/2 + step*(i-1);
        Xi(i) = X;
        t = (X - x(i))/step;
        Yi(i) = f(X);
        S(i) = y(i)*(1-t)^2*(1+2*t) + y(i+1)*t^2*(3-2*t) +
m(i)*step*t*(1-t)^2 ...
        - m(i+1)*t^2*(1+t)*step;
    endfor
endfunction

[x, y, step] = CountFunction(f, x, y, Left, Right, n);
```



```

[S, Xi, Yi, h, F, m] = Spline(x, y, n, step, Left, f);

subplot (2, 1, 1)
plot (Xi, Yi, 'p-b');
hold on;
plot (Xi, S, 'p-g');
hold off;
title ("f(x), S(x)");
subplot (2, 1, 2)
plot (Xi(1:2), Yi(1:2), 'p-g');
hold on;
plot (Xi(1:2), S(1:2), 'p-b');
hold off;
title ("f(x), S(x)");
endfunction

```