Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский Томский политехнический Университет»

Инженерная школа ядерных технологий

01.03.02 «Прикладная математика и информатика»

**Лабораторная работа № 2**

## Вариант 24

по дисциплине:

**Численные методы**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Исполнитель:** |  | | | | |
| студентка группы | 0В01 |  | Саматов Денис Сергеевич |  |  |
|  |  |  |  |  |  |
| **Руководитель:** |  | | | | |
| преподаватель |  |  | Крицкий Олег Леонидович |  |  |
|  |  |  |  |  |  |

Томск – 2022

**Задание:**

Используя значения заданной функции , построить для нее сплайн-функцию в соответствующих узлах многочлена Чебышева на заданном интервале. Использовать дополнительное краевое условие как указано в варианте задания, самостоятельно определив недостающие параметры. Вычислить достигаемое практическое значение погрешности интерполяции в точке . Построить график исходной функции и ее сплайна, выделив графически узлы интерполяции.

**Краткое теоретическое содержание**

*Сплайн-интерполяция*

Пусть в известных узлах функции *xi*, *a* = *x*0 < *x*1 < *x*2… < *xn*= *b* вычислены (заданы) значения функции *fi = f*(*xi*), *i* = 0, 1, …, *n*. На каждом промежутке [*xi*-1, *xi*] будем строить сплайн:

, *i*=1, …, *n*. (1)

Потребуем дополнительно выполнения следующих равенств:

1. , *i*=0, 1,…, *n* (значения сплайнов в узлах интерполяции совпадают со значениями функции);
2. , *i*=1,2,…,(*n*-1) (непрерывность сплайна на всем промежутке интерполяции);
3. , *i*=1,2,…,(*n*-1) (непрерывность производных сплайна на всем промежутке интерполяции);
4. , *i*=1,2,…,(*n*-1) (гладкость сплайна на всем промежутке интерполяции).

Построенный на промежутке [*a*, *b*] сплайн *S*(*x*) как объединение всех кусков имеет в узлах интерполяции легко вычисляемую первую и вторую производную. Поэтому сплайны незаменимы, если нужно достаточно точно вычислить первую и вторую производную для таблично-заданной функции *fi*.

Для определения всех 4n коэффициентов в равенстве (1) мы имеем (*n*+1)+3(*n*-1) = 4*n*-2 условий. Для однозначного задания всех коэффициентов выберем два недостающих краевых условия:

А. если вторая производная *fʹʹ*(*a*) = 0, *fʹʹ*(*b*) = 0, то *Sʹʹ*(*a*) = 0, *Sʹʹ*(*b*) = 0;

B. если первая производная *fʹ*(*a*) = α, *fʹ*(*b*) =β, то *Sʹʹ*(*a*) = α, *Sʹʹ*(*b*) = β;

С. если вторая производная *fʹʹ*(*a*) = α, *fʹʹ*(*b*) = β, то *Sʹʹ*(*a*) = α, *Sʹʹ*(*b*) = β и тд.

Найдем все неизвестные коэффициенты. Из условия получаем:

.

Из условия тогда имеем:

.

Пусть далее . Поэтому и

,

или, сдвигая индекс и умножая на (-1), имеем

|  |  |
| --- | --- |
| . | (2) |

Из условия имеем (берем первую производную в (1)):

,

или, сдвигая индекс и умножая на (-1), имеем

|  |  |
| --- | --- |
| . | (3) |

Наконец, из условия имеем (берем вторую производную в (1)):

,

Или

|  |  |
| --- | --- |
|  | (4) |

Два последних равенства системы выбираем из условия *Sʹʹ*(*X\**) = f*ʹʹ(X\**)

Таким образом, получили СЛАУ (2) – (4) для определения всех неизвестных коэффициентов в (1).

Упростим систему, для чего выразим *bi* из (2):

.

При индексе (*i*-1) из нее получаем:

,

откуда

.

Эту разность используем в (3):

,

или, приводя подобные,

, *i* = 2, 3, …, *n*. (5)

Из равенств (4) , а . Значит, подставляя их в (5), получаем:

.

Приводя подобные и заменяя индекс *i* на (*i*+1), окончательно имеем:

, *i* = 1, 2, …, (*n*-1), (6)

.

Матрица системы (5) имеет диагональное преобладание. По теореме Таусски (если матрица не имеет блочного вида , где – ненулевые матрицы, и обладает свойством диагонального преобладания, то она невырождена) заключаем, что система имеет единственное решение.

Найдя решение (6), возвращаемся к системе равенств (4), затем к (2). Записываем сплайн в виде (1).

**Замечание**: в случае равномерной сетки, когда , система (6) заметно упрощается:

, , *i* = 1, 2, …, (*n*-1).

*Решение СЛАУ методом Гаусса*

Для системы линейных алгебраических уравнений составим расширенную матрицу С, которая будет включать в себя как матрицу A (первые n строк и столбцов), так и столбец свободных членов b (последний столбец). Далее для реализации алгоритма решения [СЛАУ](https://ru.wikipedia.org/wiki/%D0%A1%D0%9B%D0%90%D0%A3) методом Гаусса необходимо реализовать два этапа.

На первом этапе осуществляем прямой ход, путём [элементарных преобразований](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%80%D0%BD%D1%8B%D0%B5_%D0%BF%D1%80%D0%B5%D0%BE%D0%B1%D1%80%D0%B0%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%BC%D0%B0%D1%82%D1%80%D0%B8%D1%86%D1%8B) над строками систему приводим к [треугольной форме](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B5%D1%83%D0%B3%D0%BE%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%BC%D0%B0%D1%82%D1%80%D0%B8%D1%86%D0%B0), либо устанавливаем, что система несовместна. Для этого среди элементов первого столбца матрицы выбираем наибольший ненулевой, перемещаем содержащую его строку в крайнее верхнее положение, делая эту строку первой. Далее ненулевые элементы первого столбца всех нижележащих строк обнуляем путём [вычитания](https://ru.wikipedia.org/wiki/%D0%92%D1%8B%D1%87%D0%B8%D1%82%D0%B0%D0%BD%D0%B8%D0%B5) из каждой строки первой строки, домноженной на отношение первого элемента этих строк к первому элементу первой строки. После того, как указанные преобразования были совершены, первую строку и первый столбец «мысленно вычёркиваем» и продолжаем, пока не останется матрица нулевого размера. Если на какой-то из итераций среди элементов первого столбца не нашёлся ненулевой, то переходим к следующему столбцу и проделываем аналогичную операцию.

На втором этапе осуществляем обратный ход, суть которого заключается в том, чтобы выразить все получившиеся базисные переменные через небазисные и построить [фундаментальную систему решений](https://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%B4%D0%B0%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%80%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D0%B9), либо, если все переменные являются базисными, то выразить в численном виде единственное решение системы линейных уравнений. Эта процедура начинается с последнего уравнения, из которого выражается соответствующая базисная переменная и подставляется в предыдущие уравнения, и так далее, поднимаясь по «ступенькам» наверх. Каждой строчке соответствует ровно одна базисная переменная, поэтому на каждом шаге, кроме последнего (самого верхнего), ситуация в точности повторяет случай последней строки.

**Ход работы**

Заданная функция:

Интервал: [-0.9, 0.9]

Краевое условие: В

Значение точки, в которой вычисляется погрешность: .

Для построения сплайна было реализовано 6-ть функций: chebyshev(), find\_c(), find\_d(), find\_b(), spline(), create\_spline(). Функция chebyshev() находит значения узлов многочлена Чебышева на заданном интервале. На вход данная функция принимает три параметра: число узлов, и границы интервалов. Функции find\_c(), find\_d(), find\_b() находят соответствующие коэффициенты сплайн-функции. Для вычисления сi использовалась формула (6), на основе которой была составлена система уравнений, для решения которой использовался метод Гаусса, описанный выше. Для нахождения di применялась формула (4), для bi – формула (2). Стоит отметить, что благодаря условию B были найдены недостающие коэффициенты b0 и bn. Функцияspline() была реализована следующим образом. На каждом интервале [*xi*-1, *xi*], где *xi*-1, *xi* – это узлы многочлена Чебышева, задавалась равномерная сетка из ста значений (с помощью функции linspace()), на которой строился сплайн с соответствующими этому интервалу найденными коэффициентами ai, bi, ci, di. create\_spline() строит графики исходной функции и найденной сплайн-функции. Результат работы программы представлен на рисунке 1.

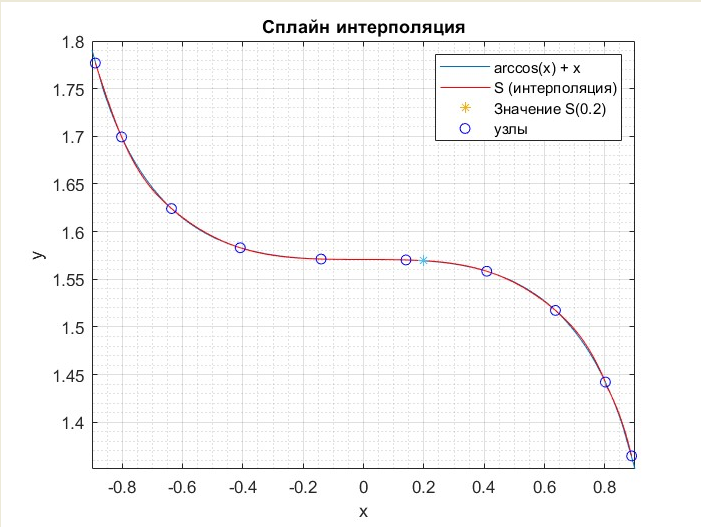


Рисунок 1 – Сплайн интерполяция.

Вычислим достигаемое практическое значение погрешности интерполяции в точке .

В результате расчета получим, что достигаемое практическое значение погрешности интерполяции в точке равняется 0.0001.

# **Вывод:**

В ходе лабораторной работы был реализован алгоритм построения сплайн-функции в соответствующих узлах многочлена Чебышевана определенном интервале для заданной функции . Также реализован алгоритм решения СЛАУ методом Гаусса. Было вычислено достигаемое практическое значение погрешности интерполяции в точке и построены графики исходной функции и ее сплайна.

Реализация алгоритмов и построение графиков было проведено в среде Matlab.

**Приложение 1**

Файл *main.m*:

clc; clear All;

left\_board = -0.9;

right\_board = 0.9;

b0 = -1 / sqrt(1-left\_board^2) + 1;

bn = -1 / sqrt(1-right\_board^2) + 1;

n = 10;

X = 0.2;

x\_ch = chebyshev(n, left\_board, right\_board);

x\_ch = sort(x\_ch);

h = [0, diff(x\_ch)];

y = @(x) acos(x) + x;

a = y(x\_ch);

c = find\_c(n, h, a);

d = find\_d(n, h, c);

b = find\_b(n, h, a, c, d, b0, bn);

[splines, f\_X] = spline(n, x\_ch,a, b, c, d, X);

create\_spline(n, x\_ch, y, splines, X, f\_X)

error = abs(y(0.2) - f\_X);

Файл *chebyshev.m*:

function x\_ch = chebyshev(n, a, b)

x\_ch = zeros(size(n));

for m = 0:n-1

x\_ch(m+1) = (b + a) / 2 + (b - a) / 2 \* cos((2 \* m + 1) \* pi / (2 \* n));

end

Файл *find\_c.m*:

function c = find\_c(n, h, y)

A = zeros(n-2, n-2);

for i = 2:(n-1)

A(i-1, i-1) = h(i);

A(i-1, i) = 2 \* (h(i) + h(i+1));

A(i-1, i+1) = h(i+1);

end

A(:, [1, 10]) = [];

B = zeros(n-2, 1);

for i = 2:(n-1)

B(i-1) = 6 \* ((y(i+1) - y(i)) / h(i+1) - (y(i) - y(i-1)) / h(i));

end

C = horzcat(A,B);

for column = 1:7

[el\_max, index\_max] = max(abs(C(column:8, column)));

index\_max = index\_max + column - 1;

C([column, index\_max],:) = C([index\_max, column],:);

for row = column+1:8

C(row, :) = C(row, :) - C(column, :) \* C(row, column) / C(column, column);

end

end

c = zeros(n-2, 1);

for i = 8:-1:1

sum = 0;

for j = i+1:8

sum = sum + c(j) \* C(i,j);

end

c(i) = (C(i,9) - sum) / C(i,i);

end

cn = 1 / h(n-1) \* (6 \* ((y(n) - y(n-1)) / h(n) - (y(n-1) - y(n-2)) / h(n-1)) - c(n-4) \* h(n-1) - 2 \* (h(n-1) + h(n)) \* c(n-3));

c = vertcat(0, c, cn);

end

Файл *find\_d.m*:

function d = find\_d(n, h, c)

d = zeros(n, 1);

for i = 2:(n-1)

d(i) = (c(i) - c(i-1)) / h(i);

end

dn = (c(n) - c(n-1)) / h(n);

d(n) = dn;

end

Файл *find\_b.m*:

function b = find\_b(n, h, a, c, d, b0, bn)

b = zeros(n, 1);

for i = 2:(n-1)

b(i) = (a(i) - a(i-1)) / h(i) + (c(i) \* h(i)) / 2 - (d(i) \* h(i) \* h(i) / 6);

end

b(1) = b0;

b(n) = bn;

end

Файл *spline.m*:

function [splines, f\_X] = spline(n, x\_ch, a, b, c, d, X)

splines = zeros(n-1, 100);

for i = 2:n

xx = linspace(x\_ch(i-1), x\_ch(i), 100);

for j = 1:length(xx)

splines(i, j) = a(i) + b(i) \* (xx(j) - x\_ch(i)) + c(i) \* (xx(j) - x\_ch(i))^2 / 2 + d(i) \* (xx(j) - x\_ch(i))^3 / 6;

if (x\_ch(i-1) <= X) && (X < x\_ch(i))

f\_X = a(i) + b(i) \* (X - x\_ch(i)) + c(i) \* (X - x\_ch(i))^2 / 2 + d(i) \* (X - x\_ch(i))^3 / 6;

end

end

end

end

Файл *create\_spline.m*:

function create\_spline(n, x\_ch, y, splines, X, f\_X)

figure('Color', 'w')

fplot(y, [-0.9 0.9])

for i = 2:n

xx = linspace(x\_ch(i-1), x\_ch(i), 100);

hold on

plot(xx, splines(i, :), 'r')

plot(X, f\_X, '\*')

plot(x\_ch, y(x\_ch), 'bo')

hold off

grid on

grid minor

end

title('Сплайн интерполяция');

legend('arccos(x) + x', 'S (интерполяция)', 'Значение S(0.2)', 'узлы')

xlabel('x')

ylabel('y')

end