Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Российский химико-технологический университет имени Д.И. Менделеева»

Факультет цифровых технологий и химического инжиниринга

Кафедра информационных компьютерных технологий

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 10**

**ПО КУРСУ**

**«ЧИСЛЕННЫЕ МЕТОДЫ В СРЕДЕ MATLAB»:**

**«Аппроксимация экспериментальных данных»**

Ведущий преподаватель

Доцент кафедры ИКТ Филиппова Е.Б.

**СТУДЕНТ группы КС-20** Мелехин А.А.

**Москва**

**2024**

# **Задание**

Построить функцию, интерполирующую данные зависимости второй переменной (плотности, вязкости, теплоёмкости...) от первой переменной (температуры, давления) и третьей переменной (весового коэффициента). По ней определить значение второй переменной при значении первой и третьей переменных, соответствующем середине интервалов между первой и второй и между 14-ой и 15-ой узловыми точками. Оценить погрешность интерполяции в этих точках. С помощью разделённых разностей определить степень полинома, наиболее точно интерполирующего заданную функцию. Найти значение второй переменной в указанных точках с помощью сплайн-интерполяции. Изобразить график построенных интерполяционных зависимостей, отметив на нём экспериментальные и рассчитанные точки.

|  |  |  |
| --- | --- | --- |
| Вариант 14 |  |  |
| Давление (кПа) | Mass Heat Capacity (kJ/kg-C) | вес.коэфф. |
| 4865 | 2,799505654 | 0,5 |
| 5065 | 2,865976811 | 0,2 |
| 5265 | 2,936424531 | 0,9 |
| 5465 | 3,010995579 | 0,5 |
| 5665 | 3,089799019 | 0,8 |
| 5865 | 3,172890855 | 0,8 |
| 6065 | 3,260255665 | 0,2 |
| 6265 | 3,351785366 | 0,5 |
| 6465 | 3,44725572 | 0,4 |
| 6665 | 3,546301795 | 0,2 |
| 6865 | 3,648394432 | 0,6 |
| 7065 | 3,752820666 | 0,5 |
| 7265 | 3,858671918 | 0,2 |
| 7465 | 3,964844364 | 0,7 |
| 7665 | 4,070055767 | 0,1 |
| 7865 | 4,172881981 | 0,8 |
| 8065 | 4,27181405 | 0,6 |
| 8265 | 4,365333422 | 0,9 |
| 8465 | 4,451998938 | 0,6 |
| 8665 | 4,530535861 | 0,9 |
| 8865 | 4,599915455 | 0,4 |

**Программа lab.m**

clear; clc;

mass\_heatCapacity = [2.799505654 2.865976811 2.936424531 3.010995579 3.089799019 3.172890855 3.260255665 3.351785366 3.44725572 3.546301795 3.648394432 3.752820666 3.858671918 3.964844364 4.070055767 4.172881981 4.27181405 4.365333422 4.451998938 4.530535861 4.599915455];

pressure = [4865 5065 5265 5465 5665 5865 6065 6265 6465 6665 6865 7065 7265 7465 7665 7865 8065 8265 8465 8665 8865];

weights = [0.5 0.2 0.9 0.5 0.8 0.8 0.2 0.5 0.4 0.2 0.6 0.5 0.2 0.7 0.1 0.8 0.6 0.9 0.6 0.9 0.4];

% 1

% Построение таблицы конечных разностей

dt = diff(mass\_heatCapacity);

disp(['Таблица конечных разностей: ',dt]);

disp(dt);

diff\_table = diffTable(pressure, mass\_heatCapacity, transpose(weights)); % Вызов функции diffTable для вычисления таблицы конечных разностей

% Максимальная степень полинома (n-2), где n - количество узловых точек

max\_degree = length(pressure) - 2;

% Определение степени полинома на основе максимальной разности в таблице

degree = find(max(abs(diff\_table)) > 1e-6, 1, 'last');

degree = min(degree, max\_degree); % Ограничиваем степень полинома максимальным значением

% Вывод степени полинома

disp(['Рекомендуемая степень полинома: ', num2str(degree)]);

% 2

% Построение аппроксимационного полинома

% без учёта весовых коэффициентов с использованием

% определителя Вандермонда

% Создание матрицы Вандермонда

W = vander(pressure);

W = W(:, end-4:end); % Выбор последних столбцов, соответствующих степеням полинома

% Решение системы линейных уравнений

coefficients = W \ mass\_heatCapacity';

% Вычисление значений полинома для построения графика

p\_density\_w = polyval(flip(coefficients'), pressure);

% Построение графика

figure;

plot(pressure, mass\_heatCapacity, 'o', pressure, p\_density\_w, '-');

xlabel('Давление в кПа');

ylabel('Теплоёмкость');

title(['Аппроксимационный полином(Вандермонд) степени ', num2str(degree)]);

legend('Экспериментальные данные', 'Аппроксимационный полином (Вандермонд)');

grid on;

% 3

% Построить аппроксимационный полином

% без учёта весовых коэффициентов с использованием

% стандартных операторов MATLAB

% Аппроксимация полиномом

p = polyfit(pressure, mass\_heatCapacity, degree);

% Вычисление значений полинома для построения графика

p\_density\_s = polyval(p, pressure);

% Построение графика

figure;

plot(pressure, mass\_heatCapacity, 'o', pressure, p\_density\_s, 'r-');

xlabel('Давление в кПа');

ylabel('Теплоёмкость');

title(['Аппроксимационный полином степени ', num2str(degree)]);

legend('Экспериментальные данные', 'Аппроксимационный полином стандарт');

grid on;

% 4

% Построить аппроксимационный полином

% с учётом весовых коэффициентов с использованием

% функции spap2

% Создание объекта сплайна

sp = spap2(pressure, 1, pressure, mass\_heatCapacity, weights);

% Оценка значений плотности для построения графика

p\_density2 = fnval(sp, pressure);

% Построение графика

figure;

plot(pressure, mass\_heatCapacity, 'o', pressure, p\_density2, 'g-');

xlabel('Давление в кПа');

ylabel('Теплоёмкость');

title(['Аппроксимационный полином(spap2) степени ', num2str(degree)]);

legend('Экспериментальные данные', 'Аппроксимационный полином spap2');

grid on;

% 5 Построить аппроксимационный полином

% с учётом весовых коэффициентов с использованием

% функции fminsearch

% Функция для минимизации (сумма квадратов разностей)

fun = @(coefficients) sum(weights .\* (polyval(fliplr(coefficients'), pressure) - mass\_heatCapacity).^2);

% Начальное приближение для коэффициентов полинома

initial\_guess = rand(1, degree+1); % Случайное начальное приближение

% Минимизация функции с использованием fminsearch

optimal\_coefficients = fminsearch(fun, initial\_guess);

% Вычисление значений полинома для построения графика

p\_density\_f = polyval(flip(optimal\_coefficients'), pressure);

% Построение графика

figure;

plot(pressure, mass\_heatCapacity, 'o', pressure, p\_density\_f, 'b-');

xlabel('Давление в кПа');

ylabel('Теплоёмкость');

title(['Аппроксимационный полином(fminsearch) степени ', num2str(degree)]);

legend('Экспериментальные данные', 'Аппроксимационный полином fminsearch');

grid on;

% 6

% Оценить точность аппроксимации

% Вычисление MSE для каждого метода

mse\_polyfit = mean((polyval(p, pressure) - mass\_heatCapacity).^2);

mse\_vander = mean((polyval(flip(coefficients'), pressure) - mass\_heatCapacity).^2);

mse\_spap2 = mean((fnval(sp, pressure) - mass\_heatCapacity).^2);

mse\_fminsearch = mean((polyval(flip(optimal\_coefficients'), pressure) - mass\_heatCapacity).^2);

% Вывод результатов

disp('Среднеквадратичная ошибка (MSE) для каждого метода:');

disp(['1. Полином методом наименьших квадратов (без учета весовых коэффициентов): ', num2str(mse\_polyfit)]);

disp(['2. Полином с использованием определителя Вандермонда: ', num2str(mse\_vander)]);

disp(['3. Сплайны с использованием функции spap2: ', num2str(mse\_spap2)]);

disp(['4. Аппроксимация с использованием функции fminsearch: ', num2str(mse\_fminsearch)]);

% 1 и 2, 14 и 15 из 3 пункта

% Вычисление абсолютной погрешности для методов 1 и 2

abs\_error\_1\_2\_polyfit = abs(polyval(p, pressure(2)) - mass\_heatCapacity(2));

abs\_error\_1\_2\_vander = abs(polyval(flip(coefficients'), pressure(2)) - mass\_heatCapacity(2));

% Вычисление абсолютной погрешности для методов 14 и 15

abs\_error\_14\_15\_polyfit = abs(polyval(p, pressure(15)) - mass\_heatCapacity(15));

abs\_error\_14\_15\_vander = abs(polyval(flip(coefficients'), pressure(15)) - mass\_heatCapacity(15));

% Вычисление среднего абсолютного значения погрешности для каждого метода

mean\_abs\_error\_1\_2\_polyfit = mean(abs\_error\_1\_2\_polyfit);

mean\_abs\_error\_1\_2\_vander = mean(abs\_error\_1\_2\_vander);

mean\_abs\_error\_14\_15\_polyfit = mean(abs\_error\_14\_15\_polyfit);

mean\_abs\_error\_14\_15\_vander = mean(abs\_error\_14\_15\_vander);

% Вывод результатов

disp('Среднее абсолютное значение погрешности для каждого метода:');

disp(['1. Полином методом наименьших квадратов (без учета весовых коэффициентов), между 1 и 2 точками: ', num2str(mean\_abs\_error\_1\_2\_polyfit)]);

disp(['2. Полином с использованием определителя Вандермонда, между 1 и 2 точками: ', num2str(mean\_abs\_error\_1\_2\_vander)]);

disp(['3. Полином методом наименьших квадратов (без учета весовых коэффициентов), между 14 и 15 точками: ', num2str(mean\_abs\_error\_14\_15\_polyfit)]);

disp(['4. Полином с использованием определителя Вандермонда, между 14 и 15 точками: ', num2str(mean\_abs\_error\_14\_15\_vander)]);

% 7

% Построить на одном графике полученные функций

% с вынесенными на них узловыми точками в виде звёздочек, разными цветами,

% добавив легенду.

figure;

hold on

grid on

% точки и Вандермонд

plot(pressure, mass\_heatCapacity, 'o', pressure, p\_density\_w, '-');

% стандарт

plot(pressure, p\_density\_s, '-');

% спап2

plot(pressure, p\_density2, '-');

% фминсерч

plot(pressure, p\_density\_f, '-');

xlabel('Давление в кПа');

ylabel('Теплоёмкость');

title('Аппроксимационные полиномы');

legend('Экспериментальные данные', 'Полином + Вандермонд', 'стандартные методы', 'spap2', 'fminsearch');

hold off

% 8

% Построить функцию, аппроксимирующую экспериментальные данные,

% не в виде полинома, в виде другой функции

% с учётом весовых коэффициентов с использованием функции fminsearch

% функция к минимизации(мнк)

obj\_function = @(params) sum(weights .\* (mass\_heatCapacity - (params(1) .\* pressure + params(2)) ./ (params(3) .\* pressure + params(4))).^2);

% Догадка

initial\_guess = [1, 1, 1, 1];

% оптимизация fminsearch

params = fminsearch(obj\_function, initial\_guess);

% Инициализация параметров

a = params(1);

b = params(2);

c = params(3);

d = params(4);

% Вывод подобранных параметров

disp(['Optimized Parameters: a = ', num2str(a), ', b = ', num2str(b), ', c = ', num2str(c), ', d = ', num2str(d)]);

% График данных и кривой

p\_range = linspace(min(pressure), max(pressure), 100);

fitted\_density = (a .\* p\_range + b) ./ (c .\* p\_range + d);

figure;

plot(pressure, mass\_heatCapacity, 'bo', p\_range, fitted\_density, 'm-');

grid on

xlabel('Давление (кПа)');

ylabel('Теплоёмкость');

legend('Экспериментальные данный', 'Кривая');

title('Аппроксимационный полином без полинома');

% 9

% Построить функцию, аппроксимирующую экспериментальные данные,

% в виде полинома Чебышёва, с учётом весовых коэффициентов

% Определим степень полинома Чебышева

degree = 5;

% Вычисляем узлы

x = 2 \* ((pressure - min(pressure)) / (max(pressure) - min(pressure))) - 1;

% Вычисление многочлена с помощью нормализованных мнк

p\_coefficients = polyfit(x, mass\_heatCapacity, degree);

% многочлен в исходных точках данных

fitted\_density = polyval(p\_coefficients, x);

% Построение исходных данных и подогнанной кривой

figure;

plot(pressure, mass\_heatCapacity, 'bo', pressure, fitted\_density, 'c-');

grid on

xlabel('Давление (кПа)');

ylabel('Теплоёмкость');

legend('Экспериментальные данный', 'Полином');

title('Аппроксимационный полином Чебышева');

function diff\_table = diffTable(x, y, w)

% Инициализация таблицы конечных разностей

n = length(x);

diff\_table = zeros(n, n);

diff\_table(:, 1) = w .\* y';

% Вычисление конечных разностей

for j = 2:n

for i = 1:n-j+1

w\_factor = w(i) / w(i+j-1); % Коэффициент весов для текущей разности

diff\_table(i, j) = w\_factor \* (diff\_table(i+1, j-1) - diff\_table(i, j-1));

end

end

end

**Результаты расчётов**

Таблица конечных разностей:

Columns 1 through 16

0.0665 0.0704 0.0746 0.0788 0.0831 0.0874 0.0915 0.0955 0.0990 0.1021 0.1044 0.1059 0.1062 0.1052 0.1028 0.0989

Columns 17 through 20

0.0935 0.0867 0.0785 0.0694

Рекомендуемая степень полинома: 19

Exiting: Maximum number of function evaluations has been exceeded

- increase MaxFunEvals option.

Current function value: 34933019998953410443245567526911274885257557440546950061646308504220556525544472282290752979538097267690921803297029371929296896.000000

Среднеквадратичная ошибка (MSE) для каждого метода:

1. Полином методом наименьших квадратов (без учета весовых коэффициентов): 1.3173e-16

2. Полином с использованием определителя Вандермонда: 4.556829903691938e+17

3. Сплайны с использованием функции spap2: 0.00013156

4. Аппроксимация с использованием функции fminsearch: 3.955328862803589e+148

Среднее абсолютное значение погрешности для каждого метода:

1. Полином методом наименьших квадратов (без учета весовых коэффициентов), между 1 и 2 точками: 8.4866e-09

2. Полином с использованием определителя Вандермонда, между 1 и 2 точками: 221880247.8152

3. Полином методом наименьших квадратов (без учета весовых коэффициентов), между 14 и 15 точками: 6.1407e-09

4. Полином с использованием определителя Вандермонда, между 14 и 15 точками: 768983076.7925

Optimized Parameters: a = 0.00093317, b = 1.1015, c = -1.0804e-05, d = 2.1195