МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Лабораторная работа №1

МЕТОДЫ ВОССТАНОВЛЕНИЯ ФУНКЦИИ ОДНОЙ ПЕРЕМЕННОЙ

Учебно-методическое пособие к выполнению лабораторных работ по дисциплине «Математическое моделирование физических процессов» для студентов ИЯТШ по направлению 14.03.02 «Ядерные физика и технологии»

Томск 2020

1. Цель работы

Цель работы заключается в освоение методов восстановления эмпирических зависимостей (аппроксимация, интерполяция, экстраполяция) с применением программного комплекса на примере пакета Matlab.

2. Содержание работы

1) Изучение теоретического материала (задача интерполяции, интерполяция сплайнами, задача аппроксимации, метод наименьших квадратов).

2) Выполнение индивидуального задания с применением программного комплекса на примере пакета Matlab.

3) Подготовка отчета по лабораторной работе

3. Задача интерполяции

Пусть функция f на отрезке $[a, b]$ задана таблицей своих значений:

$$y_i = f(x_i), \ (i = 0,1,2,...,n). \tag{1}$$

Задача интерполяции состоит в построении функции g, удовлетворяющей условию:

$$g(x_i) = y_i, \quad (i = 0,1,2,...,n), \tag{2}$$

т.е. график функции g будет проходить через заданные точки (x_i, y_i). Точки x_i называют **узлами интерполяции**. Интерполяция позволяет восстанавливать точки между узлами интерполяции.

Интерполяционным многочленом степени n для заданных табличных данных (1) называется многочлен $P_n(x) = \sum_{k=0}^{n} a_k \cdot x^k$, если он удовлетворяет условиям:

$$P_n(x_i) = y_i, \quad (i = 0,1,2,...,n). \tag{3}$$

Одна из форм записи интерполяционного многочлена – **многочлен Лагранжа**:

$$\begin{gathered} L_n(x) = \sum_{j=0}^{n} y_j \cdot l_{nj}(x), \text{ где} \\ l_{nj}(x) = \prod_{\substack{k=1 \\ k \neq j}}^{n} \frac{x - x_k}{x_j - x_k} \end{gathered} \tag{4}$$

В выражении (4) параметр $l_{nj}(x)$ представляет собой многочлен степени n, удовлетворяющий условию:

$$l_{nj}(x_i)=\begin{cases}1 \text{ при } i=j\\ 0 \text{ при } i\neq j\end{cases} \tag{5}$$

Таким образом, степень многочлена L_n равна n и при $x = x_i$ в сумме обращаются в ноль все слагаемые, кроме слагаемого с номером $j = i$, равного y_i.

В инженерной практике наиболее часто используется интерполяция многочленами 1-й, 2-й и 3-й степени (линейная, квадратичная и кубическая интерполяции). Соответствующие формулы для записи многочлена Лагранжа 1-й и 2-й степени следующие:

$$L_1(x)=y_0\cdot\frac{x-x_1}{x_0-x_1}+y_1\cdot\frac{x-x_0}{x_1-x_0}, \tag{6}$$

$$\begin{aligned}L_2(x)=y_0\cdot\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}+y_1\cdot\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}+\\+y_2\cdot\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}.\end{aligned} \tag{7}$$

4. Интерполяция сплайнами

Определение сплайна. Пусть отрезок $[a, b]$ разбит точками $a = x_0 < x_1 < \ldots < x_n = b$ на n частичных отрезков $[x_{i-1}, x_i]$. **Сплайном степени** m называется функция $S_m(x)$ обладающая следующими свойствами:

1) Функция $S_m(x)$ непрерывна на отрезке $[a, b]$ вместе со всеми своими производными $S_m^{(1)}(x)$, $S_m^{(2)}(x)$, ..., $S_m^{(p)}(x)$ до некоторого порядка p;

2) На каждом частичном отрезке $[x_{i-1}, x_i]$ функция $S_m(x)$ совпадает с некоторым алгебраическим многочленом $P_{m,i}(x)$ степени m.

Разность $m - p$ между степенью сплайна и наивысшим порядком непрерывной на отрезке $[a, b]$ производной называется **дефектом сплайна**.

Наибольшее распространение на практике получили сплайны $S_3(x)$ 3-й степени (кубические сплайны) с дефектом 1 или 2. Такие сплайны на каждом из частичных отрезков $[x_{i-1}, x_i]$ совпадают с кубическим многочленом:

$$S_3(x) = P_{3,i}(x) = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2 + d_i(x - x_{i-1})^3 \tag{8}$$

и имеют на отрезке $[a, b]$ по крайней мере одну непрерывную производную $S_3'(x)$.

Пусть функция $y = f(x)$ задана таблицей своих значений (1). Сплайн $S_m(x)$ называется интерполяционным, если $S_m(x_i) = y_i$ для всех $i = 0,1, \ldots, n$. Значение $s_i = S_m'(x_i)$ называется **наклоном сплайна** в точке x_i. На отрезке $[x_{i-1}, x_i]$ интерполяционный кубический сплайн однозначно определяется заданием значений y_{i-1}, y_i, s_{i-1}, s_i и справедлива следующая формула:

$$\begin{aligned} S_3(x) = P_{3,i}(x) = {} & \frac{(x - x_i)^2 \left(2(x - x_{i-1}) + h_i\right)}{h_i^3} y_{i-1} + \\ & + \frac{(x - x_{i-1})^2 \left(2(x_i - x) + h_i\right)}{h_i^3} y_i + \\ & + \frac{(x - x_i)^2 (x - x_{i-1})}{h_i^2} s_{i-1} + \\ & + \frac{(x - x_{i-1})^2 (x - x_i)}{h_i^2} s_i. \end{aligned} \tag{9}$$

где $h_i = x_i - x_{i-1}$.

Если в точках x_i известны значения производной $y_i^{'} = f^{'}\left(x_i\right)$, то естественно положить $s_i = y_i^{'}$ для всех i = 0,1, …, n. Тогда на каждом частичном отрезке [x_{i-1}, x_i] в соответствии с формулой (9) сплайн однозначно определяется значениями y_{i-1}, y_i, $y_{i-1}^{'}$, $y_i^{'}$ (из-за этого его называют локальным сплайном).

Глобальны способы построения сплайнов. Для того чтобы сплайн $S_3(x)$ имел непрерывную на отрезке [a, b] 2-ю производную $S_3^{''}\left(x\right)$, необходимо выбирать наклоны s_i так, чтобы в точках x_i «стыка» многочленов $P_{3,i}$ и $P_{3,i+1}$ совпадали значения их 2-х производных:

$$P_{3,i}^{''}\left(x_i\right) = P_{3,i+1}^{''}\left(x_i\right),\ \ i = 1,2,...,n-1. \tag{10}$$

Используя выражение (9) и равенство (10) можно получить следующую систему уравнений относительно коэффициентов $s_i^{'}$:

$$\begin{aligned} &h_i^{-1} s_{i-1} + 2\left(h_i^{-1} + h_{i+1}^{-1}\right) s_i + h_{i+1}^{-1} s_{i+1} = 3\left[h_i^{-2}\left(y_i - y_{i-1}\right) + h_{i+1}^{-2}\left(y_{i+1} - y_i\right)\right], \\ &i = 1,2,...,n-1. \end{aligned} \tag{11}$$

Система уравнений (11) не доопределена, так как число уравнений системы n – 1 меньше числа неизвестных n + 1. Выбор двух оставшихся уравнений обычно связывают с некоторыми дополнительными условиями, накладываемыми на сплайн в граничных точках a, b (**граничными условиями**).

1) Если в граничных точках известны значения 1-й производной $f^{'}\left(a\right)$ и $f^{'}\left(b\right)$, то естественно положить:

$$s_0 = f^{'}\left(a\right),\ \ s_n = f^{'}\left(b\right). \tag{12}$$

Дополняя систему (11) уравнениями (12) получим систему уравнений с трёх диагональной матрицей. Полученный таким образом сплайн называется **фундаментальным кубическим сплайном**.

2) Если в граничных точках известны значения 2-й производной $f''(a), f''(b)$, то можно наложить на сплайн граничные условия $S_3''(a) = P_{3,1}''(x_0) = f''(a)$, $S_3''(b) = P_{3,1}''(x_n) = f''(b)$, что приводит к следующим уравнениям:

$$
\begin{aligned}
&-\frac{4s_0}{h_1} - \frac{2s_1}{h_1} + 6\frac{y_1 - y_0}{h_1^2} = f''(a), \\
&\frac{2s_{n-1}}{h_n} + \frac{4s_n}{h_n} - 6\frac{y_n - y_{n-1}}{h_n^2} = f''(b).
\end{aligned}
\tag{13}
$$

3) Полагая в уравнениях (13) $f''(a) = 0$, $f''(b) = 0$ придем к системе уравнений, определяющих так называемый естественный кубический сплайн.

5. Метод наименьших квадратов

Линейная задача наименьших квадратов. Пусть функция $y = f(x)$ задана таблицей приближенных значений:

$$y_i \approx f(x_i), \; i = 0,1,\ldots,n, \tag{14}$$

полученных с ошибками $\varepsilon_i = y_i^0 - y_i$, где $y_i^0 = f(x_i)$.

Предположим, что для аппроксимации функции f используется линейная модель:

$$y = \Phi_m(x) = a_0\varphi_0(x) + a_1\varphi_1(x) + \ldots + a_m\varphi_m(x), \tag{15}$$

Где $\varphi_0(x),\ \varphi_1(x),...,\varphi_m(x)$ – заданные базисные функции, a_0, a_1, …, a_m – параметры модели, являющиеся коэффициентами обобщенного многочлена $\Phi_m(x)$. При $\varphi_k(x)=x^k$ получим наиболее часто используемую на практике полиномиальную модель:

$$y=P_m(x)=a_0+a_1x+a_2x^2+...+a_mx^m. \tag{16}$$

В случае, когда уровень неопределенности исходных данных высок, неестественно требовать от модели (16) выполнения условий совпадения значений обобщенного многочлена $\Phi_m(x)$ в точках x_i с заданными значениями y_i, т.е. использовать интерполяцию, так как при интерполировании происходит повторение ошибок наблюдений, в то время как при обработке экспериментальных данных желательно, напротив, их сглаживание. В то же время следует все же стремиться к тому, чтобы в этих точках (x_i) выполнялись соответствующие приближенные равенства:

$$\begin{array}{l} a_0\varphi_0(x_0)+a_1\varphi_1(x_0)+...+a_m\varphi_m(x_0)\approx y_0, \\ a_0\varphi_0(x_1)+a_1\varphi_1(x_1)+...+a_m\varphi_m(x_1)\approx y_1, \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ a_0\varphi_0(x_n)+a_1\varphi_1(x_n)+...+a_m\varphi_m(x_n)\approx y_n. \end{array} \tag{17}$$

Из различных критериев, позволяющих выбрать параметры a_0, a_1, …, a_m модели (15) так, чтобы приближенные равенства (17) удовлетворялись наилучшим в некотором смысле образом, наиболее часто используется критерий наименьших квадратов:

$$s(a,y)=\sum_{i=0}^{n}\left(\sum_{j=0}^{m}\left(a_i\varphi_i(x_i)-y_i\right)\right)^2. \tag{18}$$

Для решения поставленной задачи можно воспользоваться необходимым условием экстремума функции s:

$$\frac{\partial s}{\partial a_k} = 0, \quad k = 0,1,\ldots,m. \tag{19}$$

Вычисляя частные производные функции s и изменяя порядок суммирования, от равенства (19) переходим к системе линейных алгебраических уравнений, называемой нормальной системой метода наименьших квадратов:

$$\sum_{j=0}^{m}\left(\sum_{i=0}^{n}\varphi_j\left(x_i\right)\varphi_k\left(x_i\right)\right)a_j = \sum_{i=0}^{n} y_i\varphi_k\left(x_i\right), \quad \left(k = 0,1,\ldots,m\right). \tag{20}$$

6. Выполнение индивидуального задания с применением программного комплекса на примере пакета Matlab.

1) Получить исходную таблицу данных (xi, yi) в пакете Matlab

Варианты:

1) y = Log[x] + 0.2*rand, {x от 1 до 10, шаг 1}

2) y = exp[x] + 0.2*rand, {x от 0 до 5, шаг 0.5}

3) y = 1/x + 0.1*rand, {x от 0.2 до 5.2, шаг 1}

4) y = x^2 + rand, {x от 0 до 10, шаг 1}

5) y = x + rand, {x от 0, до 5, шаг 1}

6) y = x + x^2 + rand, {x от 0 до 5, шаг 1}

7) y = sin[x] + 0.1*rand, {x от 0 до pi, шаг pi/10}

8) y =(sin[x])^2 + 0.1*rand, {x от 0 до pi, шаг pi/10}

9) y = sqrt[x] + 0.1*rand, {x от 0 до 10, шаг 1}

10) y = cos[x] + 0.1*rand, {x от 0 до pi, шаг pi/10}

11) y = log[x] – 0.2*rand, {x от 1 до 10, шаг 1}

12) y = exp[x] – 0.2*rand, {x от 0 до 5, шаг 0.5}

13) y = 1/x – 0.1*rand, {x от 0.2 до 5.2, шаг 1}

14) y = x^2 - rand, {x от 0 до 10, шаг 1}

15) y = x - rand, {x от 0 до 5, шаг 0.5}

16) y = x + x^2 - rand, {x от 0 до 5, шаг 0.5}

17) y = sin[x] – 0.1*rand, {x от 0 до pi, шаг pi/10}

18) y = (sin[x])^2 – 0.1*rand, {x от 0 до pi, шаг pi/10}

19) y = sqrt[x] – 0.1*rand, {x от 0 до 10, шаг 1}

20) y = cos[x] – 0.1*rand, { x от 0 до pi, шаг pi/10}

21) y = sqrt[x] + 0.5*rand, {x от 0 до 20, шаг 2}

22) y = exp[x] + 0.4*rand, {x от 0 до 2.5, шаг 0.25}

23) y = x + x^2 + 0.5*rand, {x от 0 до 5, шаг 0.5}

24) y = (sin[x])^3 + 0.2*rand, {x от 0 до 2*pi, шаг pi/5}

25) y =(cos[x])^2 + 0.1*rand, {x от 0 до 2*pi, шаг pi/5}

2) Построить график исходных данных с помощью функции plot.

3) Построить интерполяционный многочлен Лагранжа:

$$L_n(x) = \sum_{j=0}^{n} y_j \cdot l_{nj}(x), \text{ где}$$

$$l_{nj}(x) = \prod_{\substack{k=1 \\ k \neq j}}^{n} \frac{x - x_k}{x_j - x_k}$$

Построить график.

4) Построить интерполяционную функцию с помощью interp1. Вывести график.

5) Построить интерполяционный естественный кубический сплайн для исходных данных. Провести кубическую сплайн-интерполяцию с помощью interp1. Вывести графики.

6) Провести аппроксимацию методом наименьших квадратов исходных данных с помощью функции polyfit. В качестве базисных функций использовать обычные степенные полиномы 1-го, 2-го и 3-го порядка. Построить графики, сравнить результаты.

7) Подготовить отчет. Сделать выводы.