

11. Обработка экспериментальных данных

11.1. Метод наименьших квадратов

Метод наименьших квадратов позволяет по экспериментальным данным подобрать такую аналитическую функцию, которая проходит настолько близко к экспериментальным точкам, насколько это возможно.

Пусть в результате эксперимента были получены некоторые данные, отображенные в виде таблицы (табл. 11.1). Требуется построить аналитическую зависимость, наиболее точно описывающую результаты эксперимента.

Таблица 11.1. Экспериментальные данные

x_i	x_1	x_2	x_3	x_4	x_5	x_6	x_7	...	x_n
y_i	y_1	y_2	y_3	y_4	y_5	y_6	y_7	...	y_n

Идея метода наименьших квадратов заключается в том, что функцию

$$Y = f(x, a_0, a_1, \dots, a_k)$$

необходимо подобрать таким образом, чтобы сумма квадратов отклонений измеренных значений y_i от расчетных Y_i была наименьшей:

$$S = \sum_{i=1}^n (y_i - f(x_i, a_0, a_1, \dots, a_k))^2 \rightarrow \min \quad (11.1)$$

Задача сводится к определению коэффициентов a_i из условия (11.1). Для реализации этой задачи в Scilab предусмотрена функция $[a, S] = \text{datafit}(F, z, a_0)$,

где

F - функция параметра, которой необходимо подобрать;

z - матрица исходных данных;

c - вектор начальных приближений;

a - вектор коэффициентов;

S - сумма квадратов отклонений измеренных значений от расчетных.

Рассмотрим использование функции `datafit` на примере.

ЗАДАЧА 11.1. В результате опыта холостого хода определена зависимость потребляемой из сети мощности (P_0 , Вт) от входного напряжения (U_1 , В) для асинхронного двигателя.

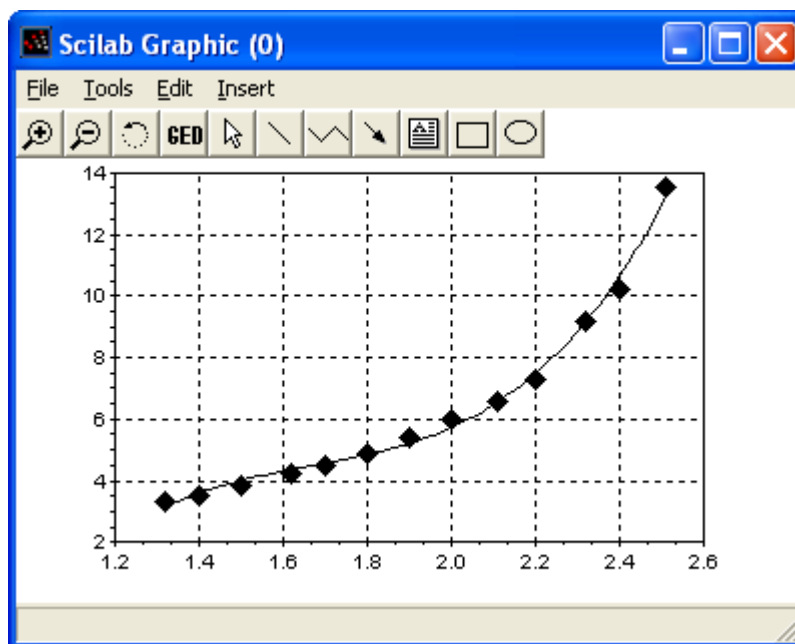
$U_1, \text{В}$	132	140	150	162	170	180	190	200	211	220	232	240	251
$P_0, \text{Вт}$	330	350	385	425	450	485	540	600	660	730	920	1020	1350

Методом наименьших квадратов подобрать зависимость вида

$$P_0 = c_1 + c_2 U_1 + c_3 U_1^2 + c_4 U_1^3 \quad (11.2).$$

Решение задачи показано в листинге 11.1. Геометрическая интерпретация на рис. 11.1.

```
//Функция (11.2)
function [zr]=G(c,z)
zr=z(2)-c(1)-c(2)*z(1)-c(3)*z(1)^2-c(4)*z(1)^3
endfunction
//Исходные данные
x=[1.32 1.40 1.50 1.62 1.70 1.80 1.90
    2.00 2.11 2.20 2.32 2.40 2.51];
y=[3.30 3.50 3.85 4.25 4.50 4.85 5.40
    6.00 6.60 7.30 9.20 10.20 13.50];
//Построение графика экспериментальных данных
plot2d(x,y,-4);
//Вектор начальных приближений
c=[0;0;0;0];
//Формирование матрицы исходных данных
z=[x;y];
//Решение задачи
[c,err]=datafit(G,z,c);
// Построение графика подобранной функции
t=1.32:0.01:2.51;
Ptc=c(1)+c(2)*t+c(3)*t^2+c(4)*t^3;
plot2d(t,Ptc);
```

Листинг 11.1*Рис. 11.1. Решение задачи 11.1*

11.2. Интерполяция функций

Простейшая задача *интерполирования* заключается в следующем. На отрезке $[a; b]$ заданы точки $x_0, x_1, x_2, \dots, x_n$ (всего $n + 1$ точка), которые называют *узлами интерполяции*, и значения некоторой функции $f(x)$ в этих точках:

$$f(x_0)=y_0, f(x_1)=y_1, f(x_2)=y_2, \dots, f(x_n)=y_n. \quad (11.2)$$

Требуется построить *интерполирующую функцию* $F(x)$, принадлежащую известному классу и принимающую в узлах интерполяции те же значения, что и $f(x)$:

$$F(x_0)=y_0, F(x_1)=y_1, F(x_2)=y_2, \dots, F(x_n)=y_n. \quad (11.3)$$

Для решения подобной задачи довольно часто используют сплайн-интерполяцию (от английского слова spline – рейка, линейка). Один из наиболее распространенных вариантов интерполяции – интерполяция кубическими сплайнами. Кроме того существуют квадратичные и линейные сплайны.

В Scilab для построения линейной интерполяции служит функция `y=interp(z, x)`, где

`z` - матрица исходных данных;

`x` - вектор абсцисс;

`y` - вектор значений линейного сплайна в точка `x`;

Пример использования функции `interp` показан в листинге 11.2. Здесь линейный сплайн применяется для решения задачи 11.1. Графическое решение задачи показано на рис. 11.2.

```
x=[132 140 150 162 170 180 190 200 211 220 232 240 251];
y=[330 350 385 425 450 485 540 600 660 730 920 1020 1350];
plot2d(x, y, -4);
z=[x; y];
t=132:5:252;
ptd=interp(z, t)
plot2d(t, ptd);
xgrid();
```

Листинг 11.2

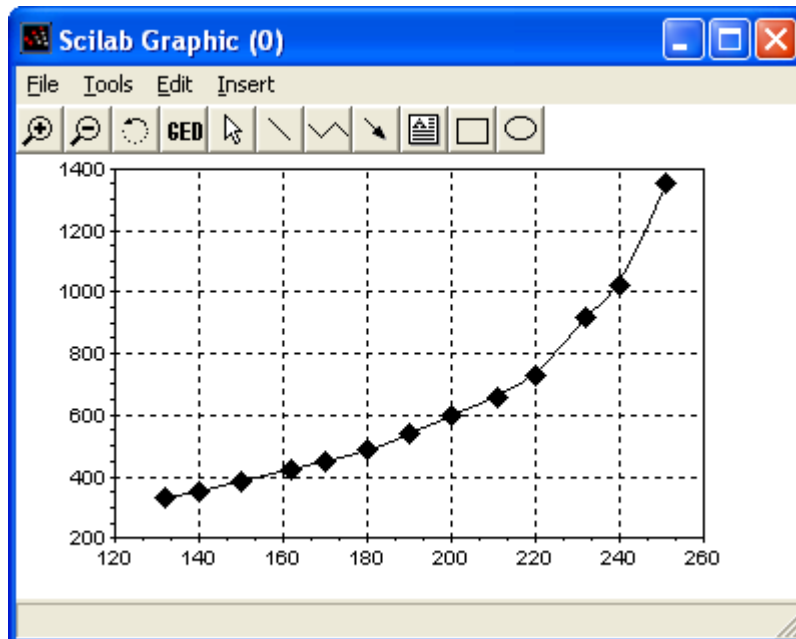


Рис. 11.2. Геометрическая интерпретация линейного сплайна

Построение кубического сплайна в Scilab состоит из двух этапов: вначале вычисляются коэффициенты сплайна с помощью функции `d=splin(x, y)`, а затем рассчитываются значения интерполяционного полинома в точке `y=interp(t, x, y, d)`.

Функция `d=splin(x, y)` имеет следующие параметры:

`x` – строго возрастающий вектор, состоящий минимум из двух компонент;

`y` – вектор того же формата, что и `x`;

`d` – результат работы функции, коэффициенты кубического сплайна.

Для функции `y=interp(t, x, y, k)` параметры `x`, `y` и `d` имеют те же значения, параметр `t` это вектор абсцисс, а `y` - вектор ординат, являющихся значениями кубического сплайна в точках `x`.

Листинг 11.3 содержит пример решения задачи 11.1 с помощью интерполяции кубическими сплайнами.

```
x=[132 140 150 162 170 180 190 200 211 220 232 240 251];  
y=[330 350 385 425 450 485 540 600 660 730 920 1020 1350];  
plot2d(x, y, -4);  
coeff=splin(x, y);  
t=132:5:252;  
ptd=interp(t, x, y, coeff);  
plot2d(t, ptd);  
xgrid();
```

Листинг 11.3