

Рис. 1. Обработка экспериментальных данных различными вычислительными методами

# Аппроксимация экспериментальных данных

В результате проведения эксперимента получена табличная функция:

Где N-количество узловых точек в таблице,  $n=N-1$ .

Задача аппроксимации заключается в отыскании аналитической зависимости  $y=f(x)$  полученной табличной функции. Существует 2 способа аппроксимации опытных данных.

<u>i</u>	X	Y
0	<u><math>x_0</math></u>	<u><math>y_0</math></u>
1	$x_1$	$y_1$
2	$x_2$	$y_2$
3	$x_3$	$y_3$
:	:	:
<u>n</u>	<u><math>x_n</math></u>	<u><math>y_n</math></u>

**Первый способ – интерполяция** требует, чтобы аппроксимирующая кривая  $\varphi(x)$ , аналитический вид которой необходимо найти, проходила через все узловые точки таблицы. Эту задачу можно решить с помощью построения интерполяционного многочлена степени  $n$ :

$$P_n(x) = a_0x^n + a_1x^{n-1} + a_2x^{n-2} + \dots + a_{n-1}x^1 + a_n.$$

Однако этот способ аппроксимации опытных данных имеет недостатки:

точность аппроксимации гарантируется в небольшом интервале  $[x_0, x_n]$  при количестве узловых точек не более 7-8; значения табличной функции в узловых точках должны быть заданы с большой точностью; вычисление значения функции за пределами отрезка  $[x_0, x_n]$  обычно производят не далее, чем на один шаг  $h$ , равный наименьшей величине  $\Delta x$ .

Известно, что как бы точно не проводился эксперимент, результаты эксперимента содержат погрешности. Исследуемая величина зависит не только от одного аргумента  $X$ , но и от других случайных факторов, которые от опыта к опыту колеблются по своим собственным случайным законам. Этим самым обуславливается случайная изменчивость исследуемой функции.

В результате аппроксимировать опытные данные с помощью интерполяционного многочлена, который проходил бы через все узловые точки таблицы, не всегда удается.

Более того, стремясь пройти через все узловые точки таблицы и увеличивая порядок многочлена, мы тем самым начинаем воспроизводить не только закономерные изменения изучаемой функции, но и ее случайные помехи.

**Второй способ.** На практике нашёл широкое применение другой способ аппроксимации опытных данных - **сглаживание опытных данных (аппроксимация)**. Сущность этого метода состоит в том, что табличные данные аппроксимируют кривой  $\varphi(x)$ , которая не обязательно должна пройти через все узловые точки, а должна как бы сгладить все случайные помехи табличной функции.

Таким образом, под аппроксимацией в общем смысле понимают задачу отыскания аналитической зависимости для описания табличной функции, и в частности – способ отыскания такой аналитической зависимости сглаживанием опытных данных.

## Сглаживание опытных данных методом наименьших квадратов

В этом методе при сглаживании опытных данных аппроксимирующую кривую  $\varphi(x)$  стремятся провести так, чтобы её отклонения  $\varepsilon_i$  от табличных данных по всем узловым точкам были минимальными ([рис. 1](#)), т.е.

$$\varepsilon_i = |\varphi(x_i) - y_i| \rightarrow \min \quad (1)$$

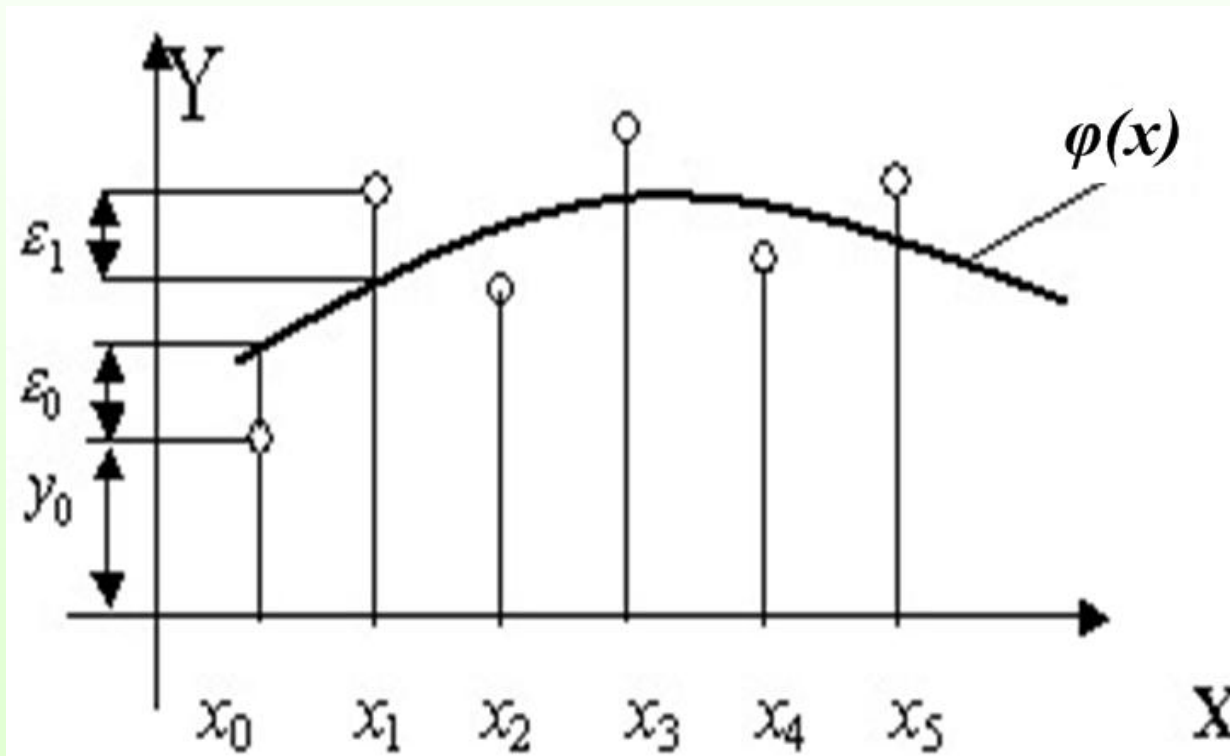


Рис.1. Графическая иллюстрация МНК

Избавимся от знака отклонения. Тогда условие (1) будет иметь вид:

$$Q = \varepsilon_i^2 = (\varphi(x_i) - y_i)^2 \rightarrow \min \quad (2)$$

Суть МНК заключается в следующем: для табличных данных, полученных в результате эксперимента, отыскать аналитическую зависимость  $\varphi(x)$ , сумма квадратов уклонений которой от табличных данных во всех узловых точках была бы минимальной, т.е.

$$Q = \sum_{i=0}^n p_i (\varphi(x_i) - y(x_i))^2 = \min, \quad (3)$$

где  $p_i$  – весовые коэффициенты,  $0 < p_i \leq 1$

Весу  $p$  для  $i$ -й точки придают смысл точности измерения данного значения: чем больше  $p$ , тем ближе аппроксимирующая кривая «притягивается» к данной точке. В дальнейшем будем по умолчанию полагать  $p = 1$  для всех точек.

Рассмотрим случай **линейной аппроксимации**:

$$\varphi(x) = a_0 \varphi_0(x) + a_1 \varphi_1(x) + \dots + a_m \varphi_m(x), \quad (4)$$

где  $\varphi_0 \dots \varphi_m$  – произвольные базисные функции,

$a_0 \dots a_m$  – неизвестные коэффициенты,  $m < n$ .

Если число коэффициентов аппроксимации взять равным числу узлов, то среднеквадратичная аппроксимация совпадет с интерполяцией Лагранжа, при этом, если не учитывать вычислительную погрешность,  $Q = 0$ .

Если известна экспериментальная (исходная) погрешность данных  $\xi$ , то выбор числа коэффициентов, то есть величины  $m$ , определяется условием:

$$\sqrt{Q} \approx \xi \quad (5)$$

Иными словами, если  $\sqrt{Q} \gg \xi$ , число коэффициентов аппроксимации недостаточно для правильного воспроизведения графика экспериментальной зависимости. Если  $\sqrt{Q} \ll \xi$ , многие коэффициенты в ур.(4) не будут иметь физического смысла.

Для решения задачи линейной аппроксимации в общем случае следует найти условия минимума суммы квадратов отклонений для ур.(4). Задачу на поиск минимума можно свести к задаче поиска корней системы уравнений

$$\frac{\partial Q}{\partial a_k} = 0, \quad k = 0, \dots, m. \quad (6)$$

Подставив ур.(4) в ур.(3), взяв производные и приравняв их к нулю, получим следующую систему *линейных алгебраических уравнений*:

$$\left\{ \begin{array}{l} \sum_{i=0}^n (a_0 \varphi_0(x_i) + a_1 \varphi_1(x_i) + \dots + a_m \varphi_m(x_i) - y_i) \varphi_0(x_i) = 0 \\ \sum_{i=0}^n (a_0 \varphi_0(x_i) + a_1 \varphi_1(x_i) + \dots + a_m \varphi_m(x_i) - y_i) \varphi_1(x_i) = 0 \\ \dots \\ \sum_{i=0}^n (a_0 \varphi_0(x_i) + a_1 \varphi_1(x_i) + \dots + a_m \varphi_m(x_i) - y_i) \varphi_m(x_i) = 0 \end{array} \right.$$

Далее следует решить полученную СЛАУ относительно коэффициентов  $a_0 \dots a_m$ , которые до их определения являются не постоянными, а варьируемыми переменными. Для её решения обычно составляется расширенная матрица коэффициентов, которую называют **матрицей Грама**, элементами которой являются скалярные произведения базисных функций и столбец свободных коэффициентов:

$$\begin{bmatrix} (\varphi_0, \varphi_0) & (\varphi_0, \varphi_1) & \dots & (\varphi_0, \varphi_m) & (\varphi_0, y) \\ (\varphi_1, \varphi_0) & (\varphi_1, \varphi_1) & \dots & (\varphi_1, \varphi_m) & (\varphi_1, y) \\ \dots & \dots & \dots & \dots & \dots \\ (\varphi_m, \varphi_0) & (\varphi_m, \varphi_1) & \dots & (\varphi_m, \varphi_m) & (\varphi_m, y) \end{bmatrix}$$



где , 
$$(\varphi_j, \varphi_k) = \sum_{i=0}^n \varphi_j(x_i) \varphi_k(x_i) \quad (\varphi_j, y) = \sum_{i=0}^n \varphi_j(x_i) y(x_i) \quad j = 0..m, k = 0..m$$
 .

После того как с помощью, например, метода Гаусса найдены коэффициенты  $a_0...a_m$ , можно построить аппроксимирующую кривую или вычислить координаты заданной точки. Таким образом, задача аппроксимации решена.

### Аппроксимация каноническим полиномом

Выберем базисные функции в виде последовательности степеней аргумента  $x$ :

$$\varphi_0(x) = x^0 = 1; \varphi_1(x) = x^1 = x; \varphi_m(x) = x^m, m < n.$$

Т.е. искомая функция  $\varphi(x)$  будет из класса алгебраических многочленов степени  $m$ :

$$\varphi(x) = P_m(x) = a_0x^0 + a_1x^1 + a_2x^2 + \dots + a_mx^m. \quad (7)$$

Аппроксимирующий многочлен не проходит через все узловые точки таблицы. Поэтому его степень  $m$  не зависит от их числа. При этом всегда  $m < n$  и может меняться в пределах

$$1 \leq m \leq N - 2.$$

При  $m=0$  многочлен примет вид:  $\varphi(x) = P_0(x) = a_0$  .

Для нахождения неизвестного коэффициента  $a_0$  имеем уравнение:  $(n+1)a_0 = \sum_{i=0}^n y_i$  . Получается, что коэффициент  $a_0$  есть

среднее арифметическое значений функции в заданных точках.

Если  $m=1$ , то мы аппроксимируем табличную функцию прямой линией. Такая задача называется **линейной регрессией**.

Аппроксимирующий полином имеет вид:  $\varphi(x) = P_1(x) = a + bx$ . Можно сразу получить значения коэффициентов  $a$  и  $b$  по следующим формулам:

$$a = \bar{y} - b\bar{x} \quad b = \frac{\sum_{i=0}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=0}^n (x_i - \bar{x})^2}$$

$$\bar{x} = \sum_{i=0}^n \frac{x_i}{n} \quad \bar{y} = \sum_{i=0}^n \frac{y_i}{n}$$

Если же используется многочлен второй степени

$P_2(x) = a_0 + a_1x + a_2x^2$ , то нормальная система уравнений примет вид:

$$\begin{aligned}\frac{\partial R}{\partial a_0} &= 2 \sum_{i=0}^n (a_0 + a_1x_i + a_2x_i^2 - y_i) = 0; \\ \frac{\partial R}{\partial a_1} &= 2 \sum_{i=0}^n (a_0 + a_1x_i + a_2x_i^2 - y_i)x_i = 0; \\ \frac{\partial R}{\partial a_2} &= 2 \sum_{i=0}^n (a_0 + a_1x_i + a_2x_i^2 - y_i)x_i^2 = 0\end{aligned}\tag{8}$$

Или

$$\begin{aligned}(n+1)a_0 + \left(\sum_{i=0}^n x_i\right)a_1 + \left(\sum_{i=0}^n x_i^2\right)a_2 &= \sum_{i=0}^n y_i; \\ \left(\sum_{i=0}^n x_i\right)a_0 + \left(\sum_{i=0}^n x_i^2\right)a_1 + \left(\sum_{i=0}^n x_i^3\right)a_2 &= \sum_{i=0}^n y_i x_i; \\ \left(\sum_{i=0}^n x_i^2\right)a_0 + \left(\sum_{i=0}^n x_i^3\right)a_1 + \left(\sum_{i=0}^n x_i^4\right)a_2 &= \sum_{i=0}^n y_i x_i^2.\end{aligned}\tag{9}$$

Такая задача называется **параболической аппроксимацией**.

Линейная относительно искомым параметров  $a_0, a_1, a_2$  система уравнений (8) может быть записана в матричной форме:

$$A a = b \quad (10)$$

Где

$$A = \begin{bmatrix} n+1 & \sum_{i=0}^n x_i & \sum_{i=0}^n x_i^2 \\ \sum_{i=0}^n x_i & \sum_{i=0}^n x_i^2 & \sum_{i=0}^n x_i^3 \\ \sum_{i=0}^n x_i^2 & \sum_{i=0}^n x_i^3 & \sum_{i=0}^n x_i^4 \end{bmatrix}; \quad a = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}, \quad b = \begin{bmatrix} \sum_{i=0}^n y_i \\ \sum_{i=0}^n y_i x_i \\ \sum_{i=0}^n y_i x_i^2 \end{bmatrix}. \quad (11)$$

Для удобства выполнения матричных операций введем матрицу  $\Phi$  размера  $(n+1) \times 3$ , элементы которой определяются через значения независимых переменных  $x_i$ ,  $i = 0, 1, \dots, n$ , следующим образом:

$$\Phi = \begin{bmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ \dots & \dots & \dots \\ 1 & x_n & x_n^2 \end{bmatrix}. \quad (12)$$

Тогда нетрудно проверить справедливость равенств (11)

$$A = \Phi^T \Phi = \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_0 & x_1 & \dots & x_n \\ x_0^2 & x_1^2 & \dots & x_n^2 \end{bmatrix} \begin{bmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ \dots & \dots & \dots \\ 1 & x_n & x_n^2 \end{bmatrix}; \quad (13)$$

$$b = \Phi^T y = \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_0 & x_1 & \dots & x_n \\ x_0^2 & x_1^2 & \dots & x_n^2 \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}. \quad (14)$$

Знание элементов матрицы  $\Phi$  (12) позволяет легко вычислить матрицу коэффициентов  $A$  и столбец свободных членов  $b$  в системе линейных уравнений, в результате чего неоднородное СЛАУ в развернутом виде записывается так:

$$\Phi^T \Phi a = \Phi^T y \quad (15)$$

Для решения системы уравнений (15) с учетом равенств (13) и (14) можно использовать рассмотренные ранее методы решения СЛАУ например метод обратной матрицы:

$$a = A^{-1} b \quad \text{или} \quad a = (\Phi^T \Phi)^{-1} \Phi^T y. \quad (16)$$

**Пример.** Построить аппроксимирующий полином второй степени по методу наименьших квадратов. Значения аргументов и функций приведены в табл.

*Таблица*      Исходные данные для решения задачи аппроксимации полиномом второй степени

$i$	0	1	2	3
$x_i$	1	2	3	4
$y_i$	2	2	4	5

▼ Для аппроксимирующей зависимости вида  $y = a_0 + a_1x + a_2x^2$  и по данным таблицы строим матрицу  $\Phi$  размера  $4 \times 3$ :

$$\Phi = \begin{bmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 4 & 16 \end{bmatrix}.$$

Вычисляем элементы  $\Phi$  и вектора  $b$ :

$$A = \Phi^T \Phi = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 4 & 9 & 16 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \\ 1 & 4 & 16 \end{bmatrix} = \begin{bmatrix} 4 & 10 & 30 \\ 10 & 30 & 100 \\ 30 & 100 & 354 \end{bmatrix};$$

$$b = \Phi^T y = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 4 & 9 & 16 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \\ 4 \\ 5 \end{bmatrix} = \begin{bmatrix} 13 \\ 38 \\ 126 \end{bmatrix}.$$

Решением системы уравнений

$$\begin{cases} 4a_0 + 10a_1 + 30a_2 = 13; \\ 10a_0 + 30a_1 + 100a_2 = 38; \\ 30a_0 + 100a_1 + 354a_2 = 126 \end{cases}$$

будут значения параметров

$$a = \begin{bmatrix} 1.75 \\ -0.15 \\ 0.25 \end{bmatrix}.$$

Тогда аппроксимирующее уравнение, описывающее данную совокупность точек наилучшим образом, будет иметь вид:

$$y = 1.75 - 0.15x + 0.25x^2$$



Матрица  $\Phi$  – это матрица Вандермонда. Используя оператор MATLAB для вычисления матрицы Вандермонда, решение этой задачи можно получить так:

```
x=[1;2;3;4]
```

```
y=[2;2;4;5]
```

```
W= vander(x) ---- вычисленная матрица Вандермонда:
```

```
%Удаление 1-ого столбца из матрицы W:
```

```
W=W(1:4,2:4) %вырезали из исходной матрицы W  
строки с 1 по 4 и столбцы со 2-ого по 4
```

```
A = W'*W
```

```
b = W'*y
```

```
a = inv(A)*b
```

```
a =  
  
    0.2500  
   -0.1500  
    1.7500
```

```
A =  
  
    354    100    30  
    100     30    10  
     30     10     4  
  
b =  
  
    126  
     38  
     13
```

```
y=0.25x2-0.15x+1.75
```

```
aa=polyfit(x,y,2)
```

```
aa =  
  
    0.2500   -0.1500    1.7500
```

```
W =  
  
     1     1     1     1  
     8     4     2     1  
    27     9     3     1  
    64    16     4     1
```

```
W =  
  
     1     1     1  
     4     2     1  
     9     3     1  
    16     4     1
```

Если  $m=3$ , то мы аппроксимируем табличную функцию кубическим полиномом - **кубическая аппроксимация**.

Рассмотрим общий случай полиномиальной аппроксимации.

$$P_m(x) = a_0x^0 + a_1x^1 + a_2x^2 + \dots + a_mx^m. \quad (17)$$

или  $P_m(x) = \sum_{j=0}^m a_j x^j$ . Тогда условие (3) будет иметь вид:

$$S = \sum_{i=0}^n (a_0x_i^0 + a_1x_i^1 + a_2x_i^2 + \dots + a_mx_i^m - y_i)^2 \rightarrow \min,$$

где  $x_i$  и  $y_i$  - координаты узловых точек таблицы,

$a_j, j=1, \dots, m$  - неизвестные коэффициенты многочлена (17).

Необходимым условием существования минимума функции  $S$  является равенство нулю её первых частных производных по каждой  $a_j$ :

$$\left\{ \begin{array}{l} \frac{\delta s}{\delta a_0} \rightarrow 2 \sum_{i=0}^n ((a_0 x_i^0 + a_1 x_i^1 + \dots + a_m x_i^m - y_i) x_i^0) = 0 \\ \frac{\delta s}{\delta a_1} \rightarrow 2 \sum_{i=0}^n ((a_0 x_i^0 + a_1 x_i^1 + \dots + a_m x_i^m - y_i) x_i^1) = 0 \\ \frac{\delta s}{\delta a_2} \rightarrow 2 \sum_{i=0}^n ((a_0 x_i^0 + a_1 x_i^1 + \dots + a_m x_i^m - y_i) x_i^2) = 0 \\ \dots\dots\dots \\ \frac{\delta s}{\delta a_m} \rightarrow 2 \sum_{i=0}^n ((a_0 x_i^0 + a_1 x_i^1 + \dots + a_m x_i^m - y_i) x_i^m) = 0 \end{array} \right.$$

В результате получили систему линейных алгебраических уравнений. Раскрывая скобки и перенося свободные члены в правые части уравнений, получим в знакомой форме СЛАУ:

$$\left\{ \begin{array}{l} c_0 a_0 + c_1 a_1 + c_2 a_2 + \dots + c_m a_m = d_0, \\ c_1 a_0 + c_2 a_1 + c_3 a_2 + \dots + c_{m+1} a_m = d_1, \\ c_2 a_0 + c_3 a_1 + c_4 a_2 + \dots + c_{m+2} a_m = d_2, \\ \dots\dots\dots \\ c_m a_0 + c_{m+1} a_1 + c_{m+2} a_2 + \dots + c_{2m} a_m = d_m, \end{array} \right. \quad (18)$$

где  $a_i$  - неизвестные СЛАУ(18) ,

$$c_k = \sum_{i=0}^n x_i^k, k = \overline{0, 2m} \quad - \text{коэффициенты,}$$
$$d_j = \sum_{i=0}^n y_i x_i^k, j = \overline{0, m} \text{ - свободные члены.}$$

Порядок системы равен  $m+1$ .

Изменим индексацию в системе (18), так чтобы она начиналась с 1, а не с нуля. В результате получим:

[illegible]

где  $a_j, j = \overline{1, (m+1)}$  - неизвестные переменные СЛАУ(19),

$$c_{k,j} = \sum_{i=1}^N x_i^{k+j-2}, k = \overline{1, (m+1)}, j = \overline{1, (m+1)} - \text{коэффициенты СЛАУ(19),}$$

$$d_k = \sum_{i=1}^N y_i x_i^{k-1}, k = \overline{1, (m+1)} - \text{свободные члены,}$$

$(x_i, y_i)$  - координаты узловых точек табличной функции,

$N$  - количество узловых точек,  $i = \overline{1, N}$

$m$  - степень аппроксимирующего многочлена вида:

$$P_m(x) = a_1 x^0 + a_2 x^1 + a_3 x^2 + \dots + a_{m+1} x^m. \quad (20)$$

Для удобства вычислений можно ввести матрицу  $\Phi$ , как при параболической аппроксимации, остальные рассуждения будут те же. В этом случае её размер будет равен  $(n+1) \times (m+1)$ , где  $n+1 = N$  – число опытов:

$$\Phi = \begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^m \\ 1 & x_1 & x_1^2 & \dots & x_1^m \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_n & x_n^2 & \dots & x_n^m \end{bmatrix}.$$

С ростом порядка матрицы Грама её определитель быстро стремится к нулю, матрица становится плохо обусловленной. Решение системы уравнений с такой матрицей приведёт к значительной потере точности. Проверим это.

Пусть в качестве системы функций  $\varphi_i$ ,  $i=1, \dots, m$ , выбираются степени, т.е.  $\varphi_i(x) = x^{i-1}$ ,  $i=1, \dots, m$ , тогда, полагая в качестве отрезка аппроксимации отрезок  $[0, 1]$ , находим матрицу Грама

$$(\varphi_i \varphi_j) = 1/(i+j-1), \quad i, j=1, 1, \dots, m.$$

Матрицу Грама такого вида называют **матрицей Гильберта**. Это классический пример так называемой плохо обусловленной матрицы. Расчёт определителей матрицы Гильберта 1-6 порядков осуществляет программа Data\_sheet5.m.

Поэтому полиномиальную аппроксимацию применяют для нахождения многочленов, степень которых не выше 5.

Для программирования составим алгоритм решения задачи.

# Алгоритм решения задачи аппроксимации

1. Строим систему линейных уравнений (19). Определяем коэффициенты  $c_{k,j}$  и свободные члены  $d_k$ . Т.к. система (19) симметрична относительно главной диагонали, то достаточно определить только наддиагональные элементы системы.
2. Решаем СЛАУ(19) методом Гаусса. Находим коэффициенты  $a_j$  аппроксимирующего многочлена (20).
3. Строим аппроксимирующий многочлен (20) и определяем его значение в каждой узловой точке  $P_i = P_m(x_i)$ .
4. Находим отклонение каждой узловой точки  $\epsilon_i = P_i - y_i$ .
5. Находим сумму квадратов отклонений по всем узловым точкам  $S = \sum_{i=1}^N \epsilon_i^2$ .
6. Находим остаточную дисперсию  $D = \frac{S}{N-(m+1)}$ .

Итак, расширенная матрица - матрица Грама для степенного базиса выглядит следующим образом:

$$\begin{bmatrix} n+1 & \sum_{i=0}^n x_i & \dots & \sum_{i=0}^n x_i^m & \sum_{i=0}^n y_i \\ \sum_{i=0}^n x_i & \sum_{i=0}^n x_i^2 & \dots & \sum_{i=0}^n x_i^{m+1} & \sum_{i=0}^n x_i y_i \\ \dots & \dots & \dots & \dots & \dots \\ \sum_{i=0}^n x_i^m & \sum_{i=0}^n x_i^{m+1} & \dots & \sum_{i=0}^n x_i^{2m} & \sum_{i=0}^n x_i^m y_i \end{bmatrix}$$

Выбор базисных функций в виде степеней  $x$  *не является оптимальным* с точки зрения достижения наименьшей погрешности. Это является следствием **неортогональности** выбранных базисных функций. Свойство *ортогональности* заключается в том, что для каждого типа полинома существует отрезок  $[x_0, x_n]$ , на котором обращаются в нуль скалярные произведения полиномов разного порядка:

$$\int_{x_0}^{x_n} \rho(x) \varphi_j(x) \varphi_k(x) dx = 0, \quad j \neq k, \quad \rho - \text{некоторая весовая функция.}$$



# Ортогональность полиномов

Для таких многочленов  $Q_i(x)$  справедливы равенства

$$\int_a^b Q_i(x) Q_j(x) dx = 0 \quad \text{для } \underline{i} \neq j$$

Если, кроме того,

$$\int_a^b Q_i^2(x) dx = 1 \quad (\underline{i} = 0, 1, \dots, n)$$

то говорят, что многочлены  $Q_i(x)$  образуют *ортонормированную систему*.

Если бы базисные функции были ортогональны, то все недиагональные элементы матрицы Грама были бы близки к нулю, что увеличило бы точность вычислений, в противном случае при  $n \rightarrow \infty$  определитель матрицы Грама очень быстро стремится к нулю, т.е. система становится плохо обусловленной.

## **Аппроксимация ортогональными классическими полиномами**

Представленные ниже полиномы, относящиеся к **многочленам Якоби**, обладают свойством ортогональности в изложенном выше смысле. То есть, для достижения высокой точности вычислений рекомендуется выбирать базисные функции для аппроксимации в виде этих полиномов.

**1. Полиномы Чебышёва** (*Чебышёв Пафнутий Львович, 1821-94, рус.математик, создатель петерб. научной школы, академик Петерб. АН; его труды положили начало развитию многих новых разделов математики*)

Определены и ортогональны на отрезке  $[-1, 1]$ . В интервал ортогональности всегда можно вписать область определения исходной функции с помощью линейных преобразований.

Условие ортогональности полиномов:

$$\sum_{i=0}^n P_k(x_i)P_j(x_i) = 0 \quad (k, j = 0, 1, \dots, m) \quad (21)$$

Аппроксимирующая зависимость, использующая многочлены Чебышёва, ищется в виде:

$$f(x, a_0, a_1, \dots, a_m) = \sum_{j=0}^m a_j P_j(x) \quad (22)$$

где

$$a_j = \sum_{i=0}^n P_j x_i y_i / \sum_{i=0}^n [P_j(x_i)]^2 \quad (23)$$

Общий вид многочлена Чебышёва  $j$ -го порядка представляется в виде:

$$P_j(x) = x^j + k_1^{(j)} x^{(j-1)} + k_2^{(j)} x^{(j-2)} + \dots k_j^{(j)} \quad (24)$$

Тогда

$$P_0(x) = 1;$$

$$P_1(x) = x + k_1^{(1)};$$

$$P_2(x) = x^2 + k_1^{(2)} x + k_2^{(2)}$$

причём эти полиномы удовлетворяют рекуррентному соотношению, которое позволяет по двум предыдущим полиномам  $P_m(x)$  и  $P_{m-1}(x)$  строить новый многочлен  $P_{m+1}(x)$ :

$$P_{m+1}(x) = (x + \beta_{m+1})P_m(x) + \gamma_{m+1}P_{m-1}(x). \quad (25)$$

Коэффициенты  $\beta$  и  $\gamma$  выбираются таким образом, чтобы для всех трёх многочленов  $P_m(x)$ ,  $P_{m-1}(x)$  и  $P_{m+1}(x)$  выполнялись условия ортогональности (21)

$$\sum_{i=0}^n P_{m+1}(x_i)P_{m-1}(x_i) = \sum_{i=0}^n (x_i + \beta_{m+1})P_m(x_i)P_{m-1}(x_i) + \sum_{i=0}^n \gamma_{m+1}P_m(x_i)P_{m-1}(x_i) = 0 \quad (26)$$

Учитывая, что  $\beta_{m+1} \sum_{i=0}^n P_m(x_i)P_{m-1}(x_i) = 0$  (21) получаем выражение для  $\gamma_{m+1}$ :

$$\gamma_{m+1} = -\sum_{i=0}^n x_i P_m P_{m-1}(x_i) / \sum_{i=0}^n [P_{m-1}(x_i)]^2 \quad (27)$$

Из соотношения

$$\sum_{i=0}^n P_{m+1}(x_i)P_m(x_i) = \sum_{i=0}^n (x_i + \beta_{m+1})[P_m(x_i)]^2 + \sum_{i=0}^n \gamma_{m+1}P_{m-1}(x_i)P_m(x_i) = 0$$

с учётом  $\gamma_{m+1} \sum_{i=0}^n P_m(x_i) P_{m+1}(x_i) = 0$  (21) определяем величину:

$$\beta_{m+1} = - \sum_{i=0}^n x_i [P_m(x_i)]^2 / \sum_{i=0}^n [P_m(x_i)]^2 \quad (28)$$

Значение коэффициента  $k_1^{(1)}$  (24) определяется из условия ортогональности многочленов  $P_0(x)$  и  $P_1(x)$ :

$$\sum_{i=0}^n P_0(x_i) P_1(x_i) = 0 \quad \text{т.е.} \quad \sum_{i=0}^n 1 \cdot P_1(x_i) = \sum_{i=0}^n x_i + \sum_{i=0}^n k_1^{(1)} = 0,$$

откуда

$$k_1^{(1)} = - \frac{1}{n+1} \sum_{i=0}^n x_i \quad (29)$$

С помощью формул (24) – (28) можно получить выражение для многочлена второго порядка  $P_2(x)$ , которое, в свою очередь используется для вычисления многочлена третьего порядка  $P_3(x)$  и т.д.

**Пример.** Построить аппроксимирующую зависимость первого и второго порядков с использованием многочленов Чебышёва для зависимости степени диссоциации  $\alpha$  йодистого водорода от температуры  $T$ , если известны следующие экспериментальные данные:

$T, K$	553	573	633	673	733
$\alpha$	0,178	0,182	0,196	0,207	0,228

Для удобства вычисления сделаем следующую замену переменных для  $T$  и для  $\alpha$ :

$$x = (T - 553)/20 \quad y = 1000(\alpha - 0,178).$$

Полученные значения  $x$  и  $y$ , а также промежуточные расчёты будем записывать в таблице.

Согласно формуле (22), аппроксимирующую зависимость многочленов первой степени найдём в виде:

$$f(x, \alpha_0, \alpha_1) = \alpha_0 P_0(x) + \alpha_1 P_1(x)$$

где  $P_0(x) = 1$ ,  $P_1(x) = x - k_1^{(1)}$ . Вычислим значение коэффициента  $k_1^{(1)}$  по формуле (29)

и найдём  $P_1(x)$ :

$$k_1^{(1)} = -\frac{1}{5} \sum_{i=0}^n x_i \quad k_1^{(1)} = -\frac{25}{5} = -5 \quad P_1(x) = x - 5.$$

Таблица. Данные для решения задачи аппроксимации с использованием многочленов Чебышёва

$i$	$x_i$	$y_i$	$P_1(x_i)$	$[P_1(x_i)]^2$	$y_i P_0$	$y_i P_1(x_i)$
0	1	0	-4.0	16.0	0	0
1	2	4	-3.0	9.0	4	-12.0
2	5	18	0	0	18	0
3	7	29	2	4	29	58
4	10	50	5.0	25	50	250
$\Sigma$	25			54	101	296

□

$i$	$x_i P_1(x_i) P_0$	$x_i [P_1(x_i)]^2$	$P_2(x_i)$	$[P_2(x_i)]^2$	$y_i P_2(x_i)$
0	-4.0	16	8.31	69.06	0
1	-6.0	18	12.52	157	50.1
2	0	0	-10.6	114	-191
3	14	28	-8.4	70.5	-243.5
4	50	250	10.2	102	510
$\Sigma$	54	312		512.6	125.6

□

Коэффициенты  $a_0$  и  $a_1$  определим по формуле (23):

$$a_0 = \frac{101}{5} = 20.2; \quad a_1 = \frac{296.0}{54} = 5.49$$

Найдём аппроксимирующую зависимость многочленом первой степени:

$$y = a_0 P_0(x) + a_1 P_1(x) = 20.2 + 5.49(x - 5.0) = 5.49x - 7.2.$$

Аппроксимирующую зависимость второй степени будем искать согласно соотношению (22) в виде:

$$y = a_0 P_0(x) + a_1 P_1(x) + a_2 P_2(x),$$

где  $P_2(x) = (x + \beta_2)P_1(x) + \gamma_2 P_0(x)$ . Коэффициенты  $\gamma_2$  и  $\beta_2$  рассчитываем по формулам (27) и (28), принимая  $P_m = P_1$  и  $P_{m-1} = P_0$ :

$$\gamma_2 = -54/5 = -10.8; \quad \beta_2 = -312/54 = -5.79.$$



Тогда  $P_2 = (x - 5.79) * (x - 5.0) - (-10.8) * 1 = x^2 - 10.79x + 18.1$  .

Коэффициент  $a_2$  рассчитываем по формуле (15), полагая  $P_j = P_2$ :

$$a_2 = 125.6 / 512.6 = 0.244 \text{ .}$$

В результате получаем следующую аппроксимирующую зависимость (многочлен второй степени):

$$y = 5.49x - 7.2 + 0.244(x^2 - 10.79x + 18.1) = 0.244x^2 + 2.87x + 2.79 \text{ .}$$

## 2. Полиномы Лежандра.

Определены и ортогональны на интервале  $[-1, 1]$  с весом  $\rho = 1$ .  
Строятся следующим образом:

$$P_n(x) = \frac{1}{2^n(n!)} \frac{d^n}{dx^n} (x^2 - 1)^n, \quad P_0(x) \equiv 1$$

*Ортогональность:*

$$\int_{-1}^1 P_n(x) P_m(x) dx = 0 \quad \text{для } m \neq n,$$

$$\int_{-1}^1 (P_n(x))^2 dx = \frac{2}{2n+1}$$

*Рекуррентная формула:*  $(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x)$ .

$P_n(x)$  имеет на  $[-1, 1]$  ровно  $n$  различных действительных корней.

Если при интерполяции обычно используют в качестве базисных функций степени, то при аппроксимации в среднем в качестве базисных функций выбирают ортогональные многочлены с заданным весом. Наиболее употребительными из них являются многочлены Якоби, частным случаем которых являются многочлены Чебышёва и Лежандра.

Используют также полиномы Лагерра и Эрмита.

Итак, если имеется набор дискретных значений функции в  $(n+1)$  точке, то порядок аппроксимирующего полинома всегда должен быть меньше их числа. Чем меньше  $m$ , тем проще эмпирическая зависимость. Однако, при малых  $m$  может быть существенна ошибка аппроксимации. Чрезмерное завышение  $m$  приводит к неоправданному увеличению вычислительных операций.

При выборе порядка аппроксимации можно исследовать таблицы конечных или разделённых разностей. Их порядок, соответствующий наиболее близким по величине и малым при этом разностям, и следует выбрать в качестве порядка аппроксимирующего полинома. Также можно исследовать поведение ошибки аппроксимации в узлах полиномами различных степеней и выбрать степень полинома, наиболее точно аппроксимирующего данную табличную функцию.

Методы аппроксимации для описания зависимостей переменных, как правило, являются более предпочтительными, чем методы интерполяции.

# Реализация аппроксимации МНК в MATLAB

- решением СЛАУ
- с помощью функции **lsqcurvefit(fun,x0,xdata,ydata)**

где: xdata,ydata– табличные значения аппроксимируемой функции;

$x_0$ –стартовое значение параметров функции;

fun – функция аппроксимации, задаваемая пользователем.

coef = lsqcurvefit(fun,x0,xdata,ydata)

x = lsqcurvefit(@myfun,x0,xdata,ydata)

function F = myfun(x,xdata)

F = ...      % Compute function values at x

## Пример аппроксимации функции полиномом 2-ого порядка

```
>> x=[30, 40, 50, 60, 70, 80, 90, 100, 110, 120];  
>> Y=[6.36, 6.85, 7.34, 7.84, 8.08, 8.32, 8.57, 8.70, 8.82, 8.94];  
>> a0=[1, 1, 1];  
>> f = inline('a(3)*x.^2 + a(2)*x + a(1)' 'a','x');  
>> coef = lsqcurvefit(f, a0, x, y)
```

```
coef =  
4.4747      0.0719      -0.0003
```

или

```
>> [coef,resnorm,residual,exitflag] = lsqcurvefit(f,a0,x,y)
```

Функция найдена и имеет вид:

$$G = 4.4747 + 0.0719S - 0.0003S^2.$$

Определим абсолютную  $E$  и максимальную относительную  $d$  погрешности

$$\varepsilon = \sqrt{\frac{\sum_{i=1}^n \Delta_i^2}{n}}, \quad \delta = \frac{\varepsilon}{y_{\min}} \cdot 100\%,$$

где:

- ◆  $\Delta_i = y(x_i) - \varphi(x_i)$ ;
- ◆  $y(x_i)$  — функция, заданная в виде таблицы;
- ◆  $\varphi(x_i)$  — функция
- ◆  $n$  — число узлов исходной функции, заданной таблично;
- ◆  $y_{\min}$  — минимальное значение функции.

Задача считается решенной, если погрешность модели не превосходит заданной.

```
>> x=[30,40,50,60,70,80,90,100,110,120];  
>> y=subs('4.4747+0.0719*t+0.0003*t^2','t',x);  
>> f1=y;  
>> f2=b;  
>> e=f1-f2;  
>> s=sum(e.^2);  
>> E=sqrt(s)/length(f2);  
>> d=E/min(f2)*100
```

```
E =  
    0.0231    d=0.3633%
```

Низкая погрешность аппроксимации является доказательством адекватности математической модели.

Функция **lsqcurvefit()** достаточно универсальна. Вид аппроксимирующей функции может быть не только полиномом  $n$ -ой степени, но и другими функциями, приведёнными выше.

Функция **polyfit(x,y,n)** находит коэффициенты полинома, аппроксимирующего функцию, заданную векторами **x** и **y**; **n** – степень полинома:  $P = \text{polyfit}(x,y,n)$

где  $P(x) = p_1x^n + p_2x^{n-1} + \dots + p_nx + p_{n+1}$

Функция **polyval(p,t)** вычисляет значение полинома в точке **x=t**.

**Пример 4.** Нахождение оптимальной степени многочлена, аппроксимирующего табличную функцию.

% Функция задана таблицей значений

% Введём функцию (x, f(x))

$x = [0, 1.13, 1.5, 2.25, 3];$

$y = [4.57, 0.68, 0.39, -1.9, -4.4];$

% Вычислим приближения с различной степенью

$p0 = \text{polyfit}(x, y, 0);$

$p1 = \text{polyfit}(x, y, 1);$

$p2 = \text{polyfit}(x, y, 2);$

$p3 = \text{polyfit}(x, y, 3);$



% Вычислим ошибки (СКО) в квадрате:

$y_0 = \text{polyval}(p_0, x);$

$y_1 = \text{polyval}(p_1, x);$

$y_2 = \text{polyval}(p_2, x);$

$y_3 = \text{polyval}(p_3, x);$

$\text{err}_0 = (1/5 * \text{sum}((y - y_0).^2))^{0.5}$

$\text{err}_1 = (1/5 * \text{sum}((y - y_1).^2))^{0.5}$

$\text{err}_2 = (1/5 * \text{sum}((y - y_2).^2))^{0.5}$

$\text{err}_3 = (1/5 * \text{sum}((y - y_3).^2))^{0.5}$

Результаты расчётов:

$\text{err}_0 = 2.9793$

$\text{err}_1 = 0.2802$

$\text{err}_2 = 0.2802$

$\text{err}_3 = 0.1899$

Сравнивая, видим что лучшую точность аппроксимации даёт  $n = 3$ .

- сплайн-аппроксимация методом наименьших квадратов с помощью функции **spap2**.

`sp = spap2(knots,k,x,y)`

`sp = spap2(1,k,x,y)` – сплайн-аппроксимация полиномом степени  $k$ .

`sp = spap2(knots,k,x,y,w)`

где `knots` – вид сплайн-функции, `knots=length(x)-k`;  $k$  – порядок сплайн-функции; `x,y` – табличные значения исследуемой функции (узлы аппроксимации);

`w` – весовые коэффициенты узловых точек.

**fnp1t(sp)** – график полученного сплайна.

`sp=spap2(augknt([a b xk],n),n,x,y,ro)` - сплайн-аппроксимация полиномом Эрмита степени  $n$ .

$$Q = \sum_{i=0}^n p_i (\varphi(x_i) - y(x_i))^2 = \min$$

- определение минимума функции

с помощью функции **fminsearch**. Она возвращает массив, в котором хранятся координаты минимума и само значение минимума.

`x = fminsearch(fun,x0)`

`x = fminsearch(fun,x0,options)`    `[x,fval] = fminsearch(...)`

где `fun` – минимизируемая функция;

`x0` – координаты начала поиска

MATLAB имеет мощные средства интерполяции и аппроксимации с визуализацией результата. Продемонстрируем это на примере.

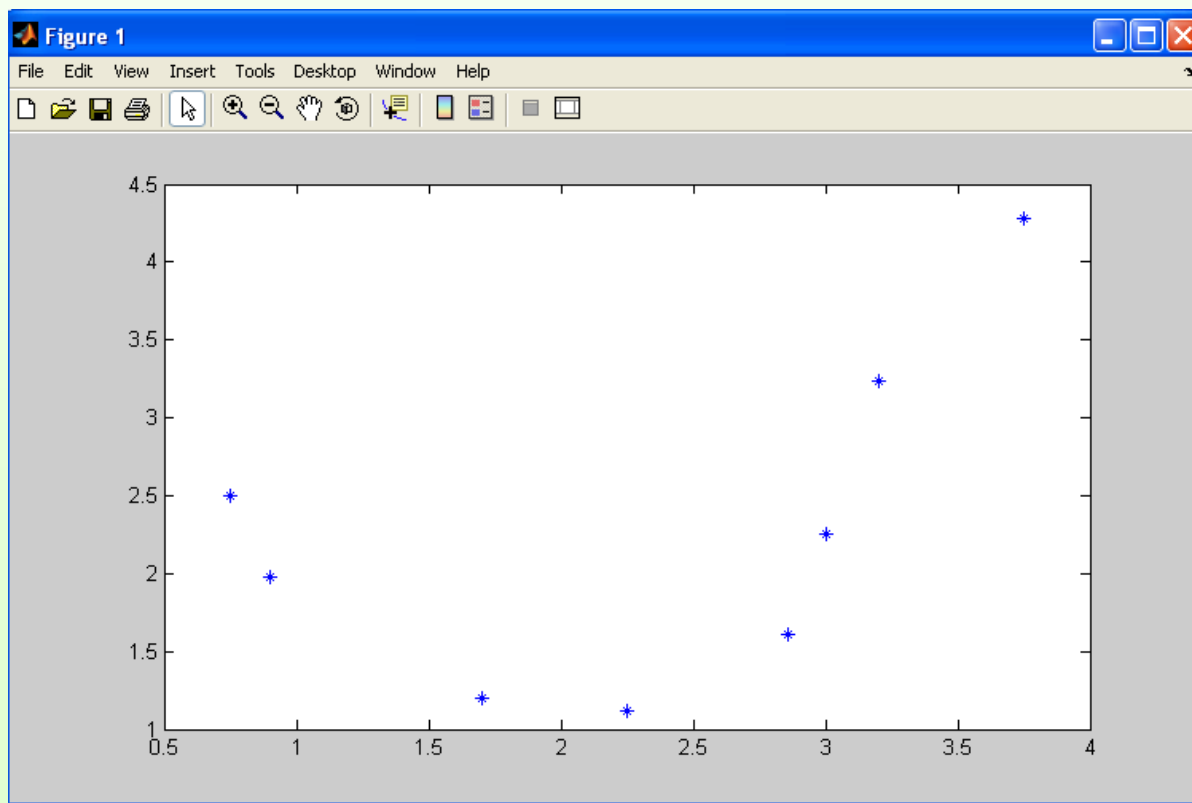
**Пример.** Аппроксимировать дискретную функцию  $y=g(x)$ .

Строим график узловых точек.

```
x=[0.75,0.90,1.70,2.25,2.86,3.00,3.20,3.75];
```

```
y=[2.50,1.98,1.20,1.12,1.61,2.25,3.24,4.28];
```

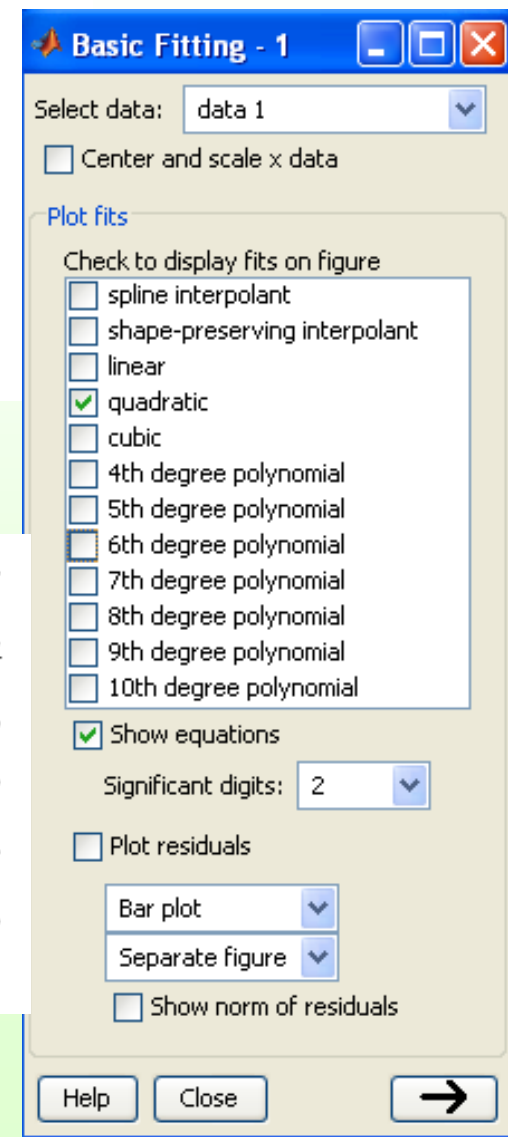
```
plot(x,y,'*')
```

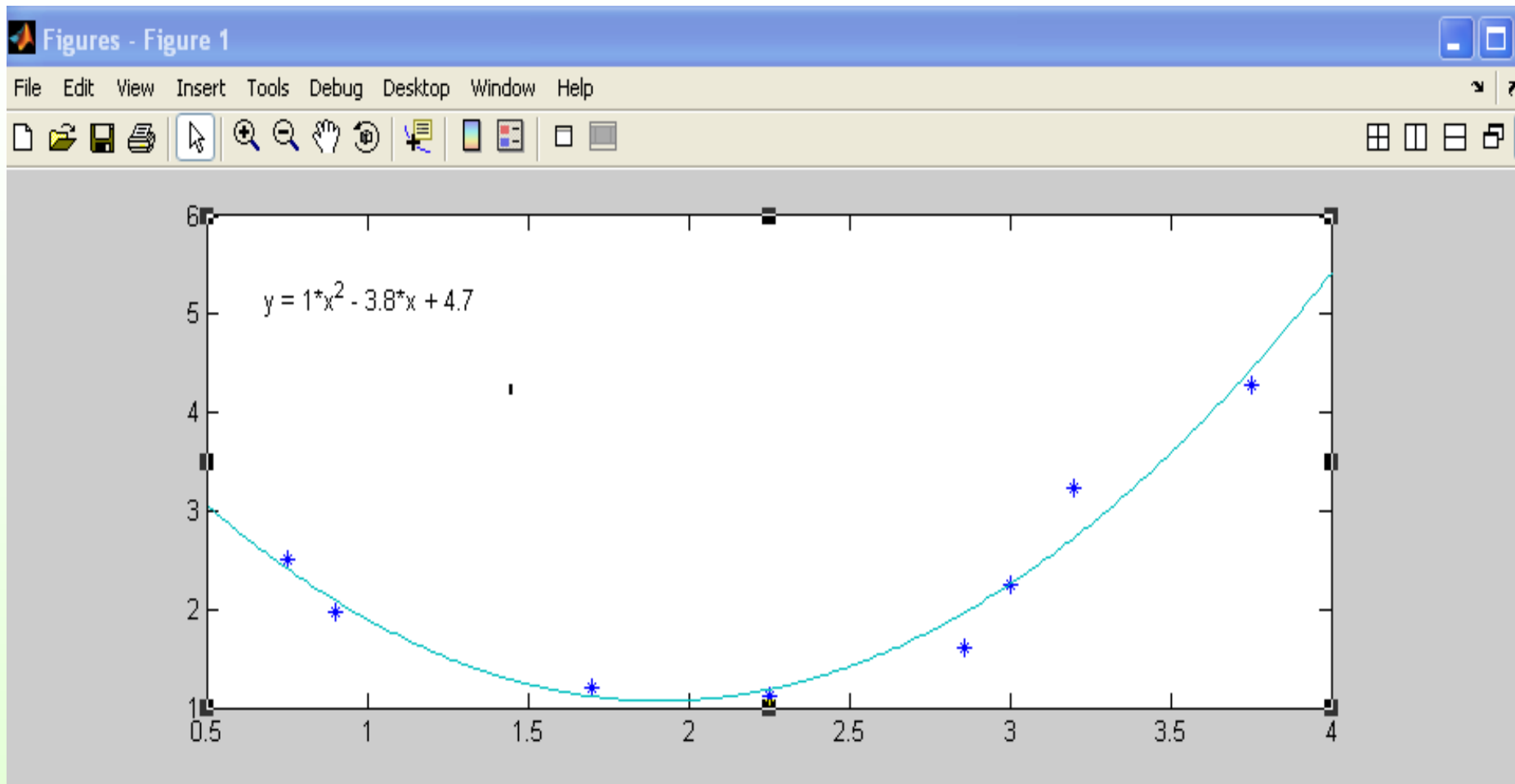


На панели инструментов окна графика узловых точек в меню **Tools** исполняем команду **Basic Fitting**. Появляется окно **Основной Монтаж**. В этом окне птичкой отмечаются необходимые данные аппроксимации. В частности, можно задать следующие операции:

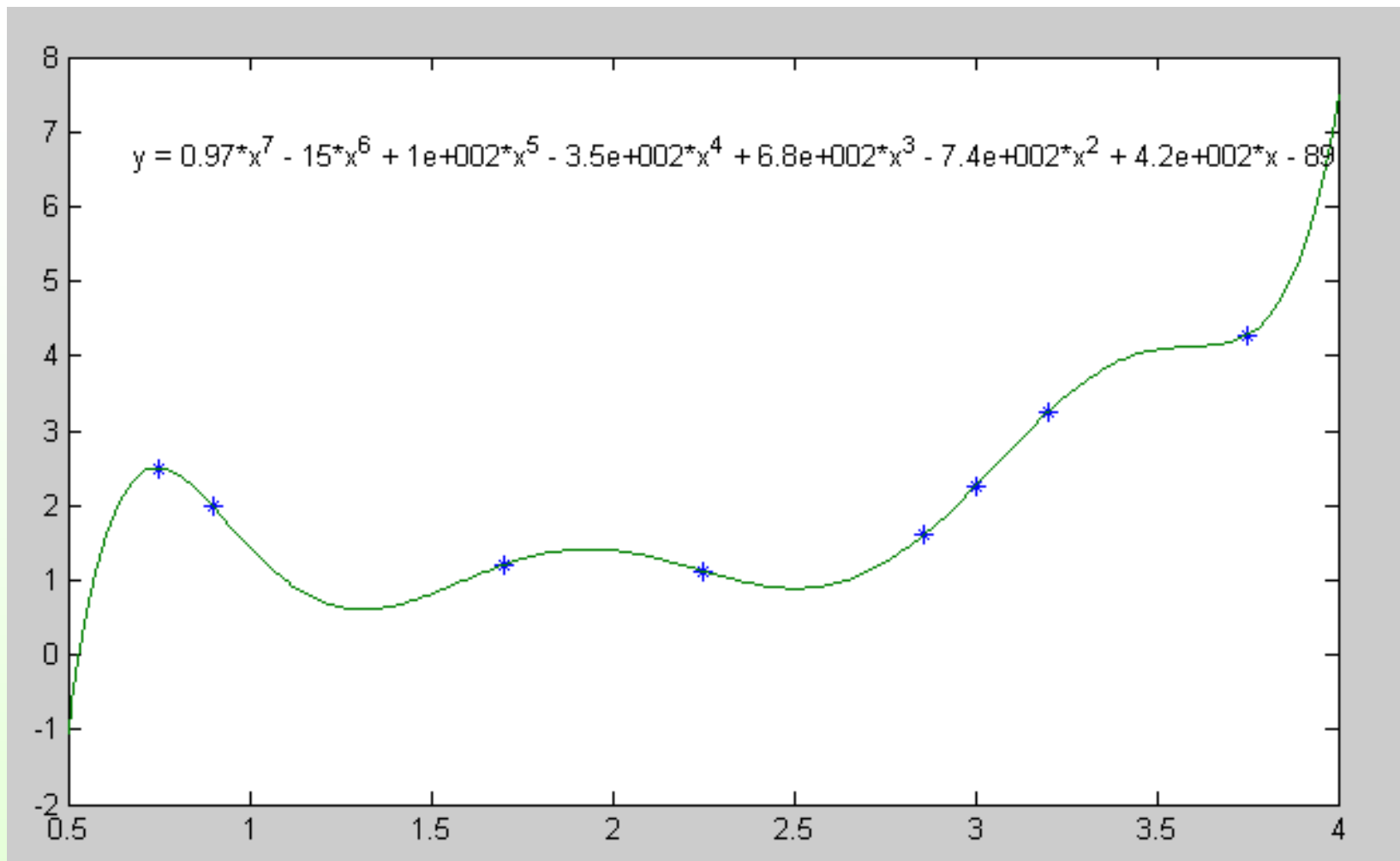
- *показать уравнение* аппроксимирующей функции  $y = g(x)$ ;
- *выбрать метод подбора*: сплайн интерполяции  
эрмитовская интерполяция  
линейный  
квадратные  
кубический  
.....

Чтобы узнать погрешность аппроксимации, надо *отметить птичкой* параметр **График остатка** в окне *Основной Монтаж*, и Показать норму остатков. График погрешностей с нормами можно вынести в *отдельное окно*, или вместе с графиком и аппроксимирующих функций – суб-график. Норма погрешностей указывает на статистическую оценку среднеквадратической погрешности. Чем она меньше, тем точнее полученная аппроксимирующая функция  $y = g(x)$ . В нашем примере:





Аппроксимация табличных данных полиномом второй степени. Аппроксимирующий полином выведен на поле графика.



Аппроксимация табличных данных полиномом седьмой степени

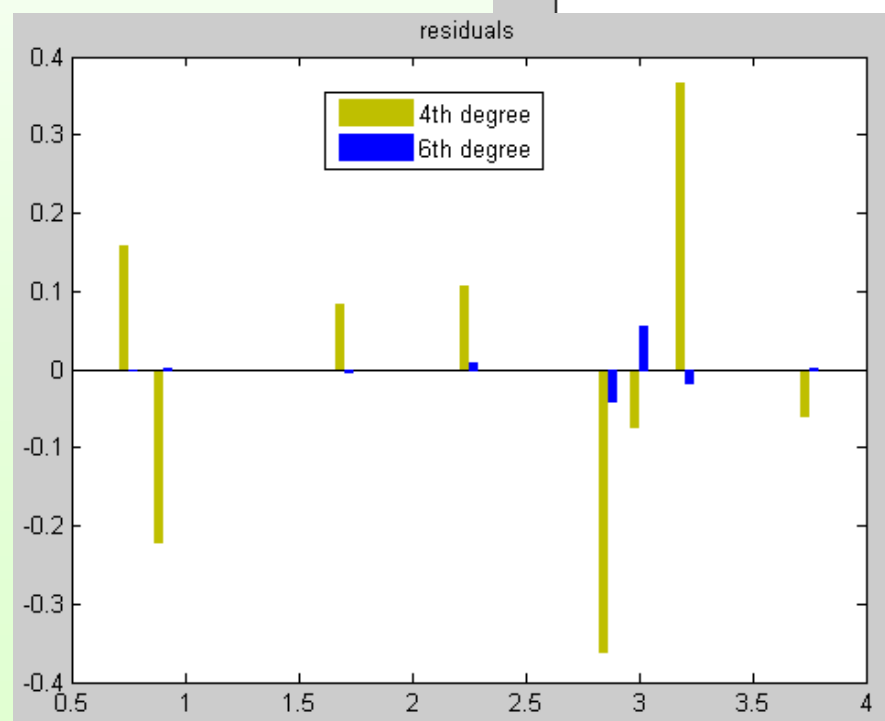
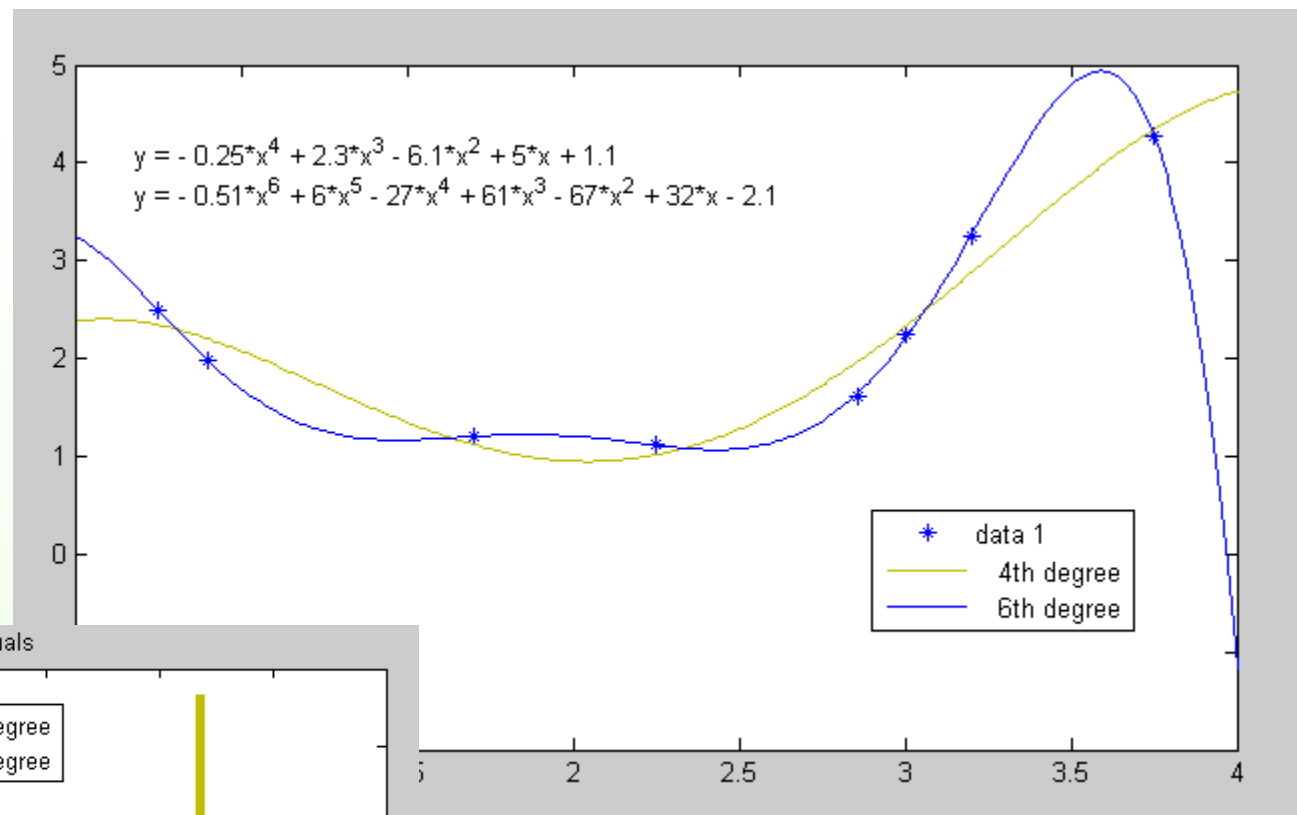


График погрешностей можно выводить в виде диаграмм (зоны), линий (линии) или отдельных точек (фрагменты). Сам график погрешностей представляет собой зависимость разности  $g(x) - f(x)$  в условных точках, соединенных прямыми линиями.

Кроме линейной и полиномиальной аппроксимации можно выбрать *сплайн-аппроксимацию* – когда на каждом интервале приближения используется *кубический полином* с новыми коэффициентами. В этом случае нельзя получить выражение для аппроксимирующей функции, т. е. такая аппроксимация является неполной. Аналогичными свойствами обладает и *Эрмитовая аппроксимация*. Она имеет только графическую интерпретацию.



## Задача.

Найти  $y(0.25)$  путём построения аппроксимирующего полинома методом наименьших квадратов согласно данным:

x	0	0.1	0.2	0.3	0.5
y	3	4.5	1.7	0.7	-1
p	0.5	0.8	1.6	0.8	0.1

Определить степень полинома, наиболее точно аппроксимирующего экспериментальные данные. Построить этот полином без учёта весовых коэффициентов с использованием определителя Вандермонда и стандартных операторов. Построить аппроксимирующий полином с учётом весовых коэффициентов с использованием функций `spar2` и `fminsearch`. Оценить точность аппроксимации.