

Метод наименьших квадратов

Волченко Ю.М.

Содержание лекции

Суть метода наименьших квадратов. Минимизация векторной функции векторного аргумента. Дифференцирование по вектору. Метод наименьших квадратов для степенной модели. Линейная, параболическая и кубическая модели. Решение учебно-практического примера.

Анимация особенностей аппроксимации функций методом наименьших квадратов.

Анимация работает только в программе Acrobat Reader!

Метод наименьших квадратов в системе *Mathematica*.

5 июля 2013 г.

1 Постановка задачи

В науке и технике используется множество формул, полученных человеком в результате познания фундаментальных законов природы: законы Ньютона, Ома, уравнения Максвелла и многие-многие другие. Кажется, что формулами описано уже все и остается только одно — постараться их запомнить.

Но это лишь на первый взгляд. На самом деле в практической инженерной деятельности потребность в формулах возникает постоянно, именно потому, что необходимых формул нет, а, может быть, и никогда не будет!

Какой формулой описать зависимость времени работы двигателя до полного его износа от количества ремонтов двигателя? Как связать количество сбоев аппаратуры с климатическими условиями ее эксплуатации? Сколько проработает устройство данного типа до того, как откажет (это время называется наработкой на отказ), при том или ином уровне вибрации в помещении?

На последнем примере мы и остановимся. Предположим, у нас есть данные по наработкам устройств одного типа на отказ (в месяцах) в зависимости от

уровня вибрации в помещении (в Дб):

Уровень вибрации	50	60	60	70	90	100	110	110	130	140
Наработка на отказ	39	24	28	22	17	12	12	10	10	6

Так как подходящая формула, связывающая наработку на отказ с уровнем вибрации, отсутствует, попробуем создать ее сами. Но сначала рассмотрим более общую ситуацию.

Пусть имеется **векторная функция векторного аргумента**:

$$\mathbf{y} = \mathbf{f}(\mathbf{x}), \quad (1)$$

где $\mathbf{x} = (x_1, \dots, x_p)^T$, $\mathbf{y} = (y_1, \dots, y_m)^T$, $\mathbf{f} = (f_1, \dots, f_m)^T$ – векторы. Такие функции нам еще не встречались, поэтому договоримся о том, что, записывая вектор-столбец (в данном случае \mathbf{x}) в качестве аргумента функции, будем считать его вектор-строкой, чтобы не загромождать запись значком транспонирования. Кроме того, компоненты вектор-функции имеют вид $f_i(\mathbf{x})$, хотя раньше мы считали функцию нескольких переменных функцией точки, а не вектора. Но из курса линейной алгебры известно, что точка отождествляется со своим радиус-вектором, поэтому такая запись правомерна.

Предположим, что, хотя о функции (1) нам ничего не известно, мы можем собрать о ней информацию такого же вида, что и в примере с наработками на отказ:

Значения аргументов	\mathbf{x}_1	\mathbf{x}_2	\mathbf{x}_3	\dots	\mathbf{x}_n
Значения функции	\mathbf{y}_1	\mathbf{y}_2	\mathbf{y}_3	\dots	\mathbf{y}_n

Эта таблица содержит пары наблюдений $(\mathbf{x}_i, \mathbf{y}_i)$, $\mathbf{y}_i = \mathbf{f}(\mathbf{x}_i)$, за поведением функции, полученные в ходе n экспериментов.

Так как точную зависимость (1) нам, скорее всего, найти не удастся, попробуем заменить ее приближенной $\mathbf{w}(\mathbf{x})$, взяв за основу систему каких-нибудь известных нам вектор-функций $\varphi_1(\mathbf{x}), \dots, \varphi_k(\mathbf{x})$:

$$\mathbf{w}(\mathbf{x}) = \sum_{j=1}^k \beta_j \varphi_j(\mathbf{x}), \quad (2)$$

где β_1, \dots, β_k — неизвестные параметры. Определив эти параметры из каких-либо разумных соображений, мы получим зависимость (2), которой можно будет пользоваться в приближенных расчетах взамен неизвестной функции (1).

Чтобы реализовать этот план, прежде всего введем компактные обозначения:

$$\Phi = \begin{pmatrix} \varphi_1(\mathbf{x}_1) & \dots & \varphi_k(\mathbf{x}_1) \\ \dots & \dots & \dots \\ \varphi_1(\mathbf{x}_n) & \dots & \varphi_k(\mathbf{x}_n) \end{pmatrix}, \quad \beta = \begin{pmatrix} \beta_1 \\ \dots \\ \beta_k \end{pmatrix}, \quad \mathbf{Y} = \begin{pmatrix} \mathbf{y}_1 \\ \dots \\ \mathbf{y}_n \end{pmatrix}.$$

Теперь значения функции (1) будут представлены матрицей \mathbf{Y} , а ее приближенные значения по формуле (2) — матрицей $\Phi\beta \approx \mathbf{Y}$.

Чтобы формула (2) обеспечивала хорошую точность приближения, необходимо, чтобы матрицы \mathbf{Y} и $\Phi\beta$ в некотором смысле были близки между собой. В качестве критерия близости в *методе наименьших квадратов (МНК)* принимают минимум функции

$$S = (\mathbf{Y} - \Phi\beta)^T \mathbf{R} (\mathbf{Y} - \Phi\beta), \quad (3)$$

где $\mathbf{R} = \{R_{ij}\}_n$ — симметричная положительно определенная матрица, элементы которой характеризуют, например, относительную точность измерения аргументов и значений функции (чем меньше точность, тем меньше соответствующий элемент матрицы \mathbf{R}).

В простейшем случае диагональной матрицы \mathbf{R} критерий (3) имеет вид

$$I = \sum_{i=1}^n R_{ii} [y_i - \mathbf{w}(\mathbf{x}_i)]^2.$$

Эта формула объясняет название метода, который заключается в минимизации суммы квадратов отклонений (взятых с «весом» R_{ii}) наблюдаемых значений функции $\mathbf{f}(\mathbf{x})$ от предполагаемых приближением (2).

2 Дифференцирование по вектору

Чтобы найти минимум функции (3), к ней нельзя вот так сразу применить теорию экстремума функции нескольких переменных, изученную вами ранее[†]. Ведь функция (3) — не скалярная функция вектора, а векторная! Поэтому упомянутую теорию следует немного модифицировать. Например, научиться брать производную от векторной функции по вектору.

Итак, пусть в общем случае имеется векторная функция векторного аргумента вида

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} f_1(\mathbf{x}) \\ \dots \\ f_m(\mathbf{x}) \end{pmatrix},$$

где $\mathbf{x} = (x_1, \dots, x_n)^T$. Она называется *дифференцируемой* в точке $\mathbf{x}_0 = (x_1^0, \dots, x_n^0)^T$, если существует матрица $\mathbf{A} = \{a_{ij}\}_n^m$ такая, что

$$\lim_{\mathbf{x} \rightarrow \mathbf{x}_0} \frac{\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{x}_0) - \mathbf{A}^T (\mathbf{x} - \mathbf{x}_0)}{\rho(\mathbf{x}, \mathbf{x}_0)} = \mathbf{0}. \quad (4)$$

[†]Лекция «Экстремумы функции нескольких переменных».

Теорема 1. Функция $\mathbf{f}(\mathbf{x})$ дифференцируема в точке \mathbf{x}_0 , тогда и только тогда, когда в этой точке дифференцируема каждая ее компонента.

Доказательство. Выпишем i -ю строку векторного равенства (4):

$$\lim_{\mathbf{x} \rightarrow \mathbf{x}_0} \frac{f_i(\mathbf{x}) - f_i(\mathbf{x}_0) - \sum_{j=1}^n a_{ji} (x_j - x_j^0)}{\rho(\mathbf{x}, \mathbf{x}_0)} = 0.$$

Тогда полное приращение скалярной функции $f_i(\mathbf{x})$ будет иметь вид

$$\Delta f_i(\mathbf{x}) = f_i(\mathbf{x}) - f_i(\mathbf{x}_0) = \sum_{j=1}^n a_{ji} \Delta x_j + o(\rho(\mathbf{x}, \mathbf{x}_0)),$$

где $\Delta x_j = x_j - x_j^0$. Последнее равенство означает, что компонента $f_i(\mathbf{x})$ дифференцируема в точке \mathbf{x}_0 . \square

Пусть все компоненты векторной функции $\mathbf{f}(\mathbf{x})$ имеют непрерывные первые частные производные по всем аргументам. Тогда каждая компонента дифференцируема, и ее полное приращение выражается с помощью ее частных производных:

$$\Delta f_i(\mathbf{x}) = \sum_{j=1}^n \frac{\partial f_i}{\partial x_j} \Delta x_j + o(\rho(\mathbf{x}, \mathbf{x}_0)).$$

Следовательно, $a_{ji} = \frac{\partial f_i}{\partial x_j}$ и формулу (4) можно переписать в виде

$$\lim_{\mathbf{x} \rightarrow \mathbf{x}_0} \frac{\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{x}_0) - \frac{d\mathbf{f}}{d\mathbf{x}}^T (\mathbf{x} - \mathbf{x}_0)}{\rho(\mathbf{x}, \mathbf{x}_0)} = \mathbf{0},$$

где

$$\frac{d\mathbf{f}}{d\mathbf{x}} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_1} \\ \cdots & \cdots & \cdots \\ \frac{\partial f_1}{\partial x_n} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix} = \left\{ \frac{\partial f_j}{\partial x_i} \right\}_n^m, \quad (5)$$

так что $\mathbf{A} = \frac{d\mathbf{f}}{d\mathbf{x}}$. Для скалярной функции f имеем

$$\frac{df}{d\mathbf{x}} = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \cdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix} = \nabla f.$$

Это позволяет записать необходимые условия экстремума скалярной функции f в виде

$$\frac{df}{d\mathbf{x}} = 0.$$

СВОЙСТВА ПРОИЗВОДНОЙ ПО ВЕКТОРУ

1° $\frac{d\mathbf{x}}{d\mathbf{x}} = \mathbf{E}$, где \mathbf{E} – единичная матрица порядка n .

2° Постоянную матрицу \mathbf{C} можно выносить за знак производной:

$$\frac{d(\mathbf{C}\mathbf{f})}{d\mathbf{x}} = \frac{d\mathbf{f}}{d\mathbf{x}} \mathbf{C}^T,$$

3° Производные суммы, разности и произведения двух функций:

$$\frac{d(\mathbf{f} \pm \mathbf{g})}{d\mathbf{x}} = \frac{d\mathbf{f}}{d\mathbf{x}} \pm \frac{d\mathbf{g}}{d\mathbf{x}}, \quad \frac{d(\mathbf{f}^T \mathbf{g})}{d\mathbf{x}} = \frac{d\mathbf{f}}{d\mathbf{x}} \mathbf{g} + \frac{d\mathbf{g}}{d\mathbf{x}} \mathbf{f}.$$

4° Производная сложной функции:

$$\frac{d\mathbf{f}[\mathbf{g}(\mathbf{x})]}{d\mathbf{x}} = \frac{d\mathbf{g}}{d\mathbf{x}} \cdot \frac{d\mathbf{f}}{d\mathbf{g}}.$$

Доказательства вынесены в Приложение¹).

Докажите самостоятельно равенства, считая вектор \mathbf{c} и матрицу \mathbf{A} постоянными²):

$$\frac{d(\mathbf{c}^T \mathbf{x})}{d\mathbf{x}} = \frac{d(\mathbf{x}^T \mathbf{c})}{d\mathbf{x}} = \mathbf{c}, \quad (6)$$

$$\frac{d(\mathbf{A}\mathbf{x})}{d\mathbf{x}} = \mathbf{A}^T, \quad (7)$$

$$\frac{d(\mathbf{x}^T \mathbf{A}\mathbf{x})}{d\mathbf{x}} = (\mathbf{A}^T + \mathbf{A}) \mathbf{x}, \quad (8)$$

$$\frac{d(\mathbf{x}^T \mathbf{A}\mathbf{x})}{d\mathbf{x}} = 2\mathbf{A}\mathbf{x}, \text{ если матрица } \mathbf{A} \text{ симметрична.} \quad (9)$$

Определим вторую производную скалярной функции f :

$$\frac{d^2 f}{d\mathbf{x}^2} \triangleq \frac{d}{d\mathbf{x}} \left(\frac{d\mathbf{f}}{d\mathbf{x}} \right) = \frac{d}{d\mathbf{x}} \begin{pmatrix} \frac{df}{dx_1} \\ \dots \\ \frac{df}{dx_n} \end{pmatrix} = \begin{pmatrix} \frac{d^2 f}{dx_1^2} & \dots & \frac{d^2 f}{dx_n dx_1} \\ \dots & \dots & \dots \\ \frac{d^2 f}{dx_1 dx_n} & \dots & \frac{d^2 f}{dx_n^2} \end{pmatrix} = \mathbf{H}^T.$$

Получилась транспонированная матрица Гессе. Известно, что положительная (отрицательная) определенность матрицы Гессе определяет наличие экстремума и его тип для функции f в ее стационарной точке. Но определенность матриц \mathbf{H} и \mathbf{H}^T одинакова, так как:

$$\mathbf{x}^T \mathbf{H}^T \mathbf{x} = \sum_{i=1}^n \sum_{j=1}^n x_i h_{ji} x_j = \sum_{j=1}^n \sum_{i=1}^n x_j h_{ji} x_i = \mathbf{x}^T \mathbf{H} \mathbf{x}.$$

Таким образом, матрица $d^2 f / d\mathbf{x}^2$ определяет наличие экстремума и его тип для функции f в ее стационарной точке.

3 Минимизация суммы квадратов отклонений

Вернемся к минимизации функции (3). Используя формулу производной сложной функции и формулы (7), (9), найдем первую производную функции S и приравняем ее к нулю:

$$\begin{aligned} \frac{dS}{d\beta} &= 2 \frac{d(\mathbf{Y} - \Phi\beta)}{d\beta} \mathbf{R}(\mathbf{Y} - \Phi\beta) = -2\Phi^T \mathbf{R}(\mathbf{Y} - \Phi\beta) = \\ &= -2(\Phi^T \mathbf{R} \mathbf{Y} - \Phi^T \mathbf{R} \Phi \beta) = 0. \end{aligned} \quad (10)$$

Решим полученное уравнение относительно вектора неизвестных параметров β , предполагая, что матрица $\Phi^T \mathbf{R} \Phi$ невырождена:

$$\beta = (\Phi^T \mathbf{R} \Phi)^{-1} \Phi^T \mathbf{R} \mathbf{Y}. \quad (11)$$

Найдем вторую производную функции S :

$$\frac{d^2 S}{d\beta^2} = \frac{d}{d\beta} \left(\frac{dS}{d\beta} \right) = \frac{d}{d\beta} [-2(\Phi^T \mathbf{R} \mathbf{Y} - \Phi^T \mathbf{R} \Phi \beta)] = 2\Phi^T \mathbf{R} \Phi.$$

Если матрица \mathbf{R} положительно определена, то и матрица $\Phi^T \mathbf{R} \Phi$ положительно определена, так как для любого вектора \mathbf{x} получаем

$$\mathbf{x}^T \Phi^T \mathbf{R} \Phi \mathbf{x} = \mathbf{y}^T \mathbf{R} \mathbf{y} \geq 0, \quad (12)$$

где $\mathbf{y} = \Phi \mathbf{x}$. Следовательно, в стационарной точке (11) функция S имеет минимум.

Впрочем, минимум функции S можно найти и без дифференцирования³⁾.

4 МНК для степенной модели

Давайте детально рассмотрим применение МНК для аппроксимации скалярной функции скалярного аргумента

$$y = f(x), \quad (13)$$

взяв для простоты матрицу \mathbf{R} единичной: $\mathbf{R} = \mathbf{E}$. В этом случае приближение (2) запишется как

$$w(x) = \sum_{j=1}^k \beta_j \varphi_j(x), \quad (14)$$

а вектор (11), обеспечивающий минимум функции S , как

$$\beta = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{Y}. \quad (15)$$

Кроме того, и функции $\varphi_j(x)$ выберем достаточно простыми — пусть они представляют собой одночлены $\varphi_j(x) = x^{j-1}$, $j = \overline{1, k}$, причем, $k - 1 < n$; такое приближение часто встречается в практических приложениях. Тогда формула (2) примет вид

$$w(x) = \sum_{j=1}^k \beta_j x^{j-1}. \quad (16)$$

Фактически, мы рассматриваем приближение функции (13) многочленом. Но приближение многочленами мы уже изучали на лекции, посвященной интерполяции; зачем же нам еще один вид аппроксимации? Здесь надо вспомнить, что интерполяционный многочлен обязан проходить через заданные точки на плоскости. В методе МНК такое требование невозможно. Почему? А вы взгляните на таблицу данных для наработок на отказ. Там есть пара точек (60; 24) и (60; 28), через которые график многочлена провести ну никак не возможно, потому что эти точки расположены на вертикальной прямой. Ничего странного в этом нет, поскольку в экспериментах вполне может наблюдаться такое явление, что при одном и том же уровне вибрации у разных (но идентичных) приборов наработки на отказ не одинаковы.

Как же проходит график многочлена МНК в области, в которой располагаются пары точек (x_i, y_i) (назовем ее облаком точек)? Обсудим это несколько позже.

Вернемся теперь к формуле (16). При $k = 2$ ее называют **линейной**, при $k = 3$ — **параболической**, при $k = 4$ — **кубической** моделью функции (13). Матрица Φ преобразуется к виду

$$\Phi = \begin{pmatrix} 1 & x_1 & \dots & x_1^{k-1} \\ \dots & \dots & \dots & \dots \\ 1 & x_n & \dots & x_n^{k-1} \end{pmatrix},$$

так что нетрудно найти матрицы, используемые в выражении (15):

$$\begin{aligned} \Phi^T \Phi &= \begin{pmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \\ \dots & \dots & \dots & \dots \\ x_1^{k-1} & x_2^{k-1} & \dots & x_n^{k-1} \end{pmatrix} \begin{pmatrix} 1 & x_1 & \dots & x_1^{k-1} \\ 1 & x_2 & \dots & x_2^{k-1} \\ \dots & \dots & \dots & \dots \\ 1 & x_n & \dots & x_n^{k-1} \end{pmatrix} = \\ &= \begin{pmatrix} n & M_1 & M_2 & \dots & M_{k-1} \\ M_1 & M_2 & M_3 & \dots & M_k \\ \dots & \dots & \dots & \dots & \dots \\ M_{k-1} & M_k & M_{k+1} & \dots & M_{2k-2} \end{pmatrix}, \end{aligned}$$

где $M_l = \sum_{i=1}^n x_i^l, l = \overline{1, 2k-2}$,

$$\Phi^T \mathbf{Y} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \\ \dots & \dots & \dots & \dots \\ x_1^{k-1} & x_2^{k-1} & \dots & x_n^{k-1} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{pmatrix} = \begin{pmatrix} K_0 \\ K_1 \\ \dots \\ K_{k-1} \end{pmatrix},$$

где $K_l = \sum_{i=1}^n x_i^l y_i, l = \overline{0, k-1}$.

Таким образом, решение (15) принимает вид

$$\begin{pmatrix} \beta_1 \\ \dots \\ \beta_n \end{pmatrix} = \begin{pmatrix} n & M_1 & \dots & M_{k-1} \\ M_1 & M_2 & \dots & M_k \\ \dots & \dots & \dots & \dots \\ M_{k-1} & M_k & \dots & M_{2k-2} \end{pmatrix}^{-1} \begin{pmatrix} K_0 \\ K_1 \\ \dots \\ K_{k-1} \end{pmatrix}. \quad (17)$$

Обратная матрица всегда существует, так как $\Phi^T \Phi$ невырождена⁴). Наиболее простой вид выражение (17) приобретает для линейной модели:

$$\begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} n & M_1 \\ M_1 & M_2 \end{pmatrix}^{-1} \begin{pmatrix} K_0 \\ K_1 \end{pmatrix}.$$

Обратную матрицу тоже несложно найти:

$$\begin{pmatrix} n & M_1 \\ M_1 & M_2 \end{pmatrix}^{-1} = \frac{1}{nM_2 - M_1^2} \begin{pmatrix} M_2 & -M_1 \\ -M_1 & n \end{pmatrix},$$

поэтому можно в явном виде получить коэффициенты функции (16):

$$\beta_1 = \frac{K_0 M_2 - K_1 M_1}{nM_2 - M_1^2}, \quad \beta_2 = \frac{-K_0 M_1 + K_1 n}{nM_2 - M_1^2}. \quad (18)$$

5 Решение примера

Построим линейную, параболическую и кубическую модели для примера, рассмотренного в начале лекции. Чтобы определить, какая модель лучше, для каждой из них вычислим **остаток**

$$e = \frac{1}{n} \sum_{i=1}^n [y_i - w(x_i)]^2.$$

Вычисления сведем в таблицу:

x_i	y_i	$x_i^2 10^{-2}$	$x_i^3 10^{-3}$	$x_i^4 10^{-4}$	$x_i^5 10^{-5}$	$x_i^6 10^{-6}$	$x_i y_i$	$x_i^2 y_i 10^{-2}$	$x_i^3 y_i 10^{-3}$	
50	30	25	125	625	3125	15625	1500	750	3750	
60	24	36	216	1296	7776	46656	1440	864	5184	
60	28	36	216	1296	7776	46656	1680	1008	6048	
70	22	49	343	2401	16807	117649	1540	1078	7546	
90	17	81	729	6561	59049	531441	1530	1377	12393	
100	12	100	1000	10000	100000	1000000	1200	1200	12000	
110	12	121	1331	14641	161051	1771561	1320	1452	15972	
110	10	121	1331	14641	161051	1771561	1100	1210	13310	
130	10	169	2197	28561	371293	4826809	1300	1690	21970	
140	6	196	2744	38416	537824	7529536	840	1176	16464	
Σ	920	171	934	10232	118438	1425752	17657494	13450	11805	114637
M_1	K_0	M_2	M_3	M_4	M_5	M_6	K_1	K_2	K_3	

Линейная модель. По формулам (18) получаем

$$\beta_1 = \frac{171 \cdot 93400 - 13450 \cdot 920}{10 \cdot 93400 - 920^2} = 41,0662;$$

$$\beta_2 = \frac{-171 \cdot 920 + 13450 \cdot 10}{10 \cdot 93400 - 920^2} = -0,260502;$$

следовательно,

$$w(x) = -0,260502x + 41,0662;$$

остаток:

$$e = 3,843.$$

Параболическая модель. По формуле (17) при $k = 3$ находим

$$\begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix} = \begin{pmatrix} 10 & 920 & 93400 \\ 920 & 93400 & 10232000 \\ 93400 & 10232000 & 1184380000 \end{pmatrix}^{-1} \begin{pmatrix} 171 \\ 13450 \\ 1180500 \end{pmatrix} =$$

$$= \begin{pmatrix} 56,5078 \\ -0,627405 \\ 0,0019608 \end{pmatrix}$$

и, значит,

$$w(x) = 0,0019608x^2 - 0,627405x + 56,5078;$$

остаток:

$$e = 1,809.$$

Кубическая модель. Аналогично находим

$$w(x) = -7,63808 \cdot 10^{-6}x^3 + 0,00417015x^2 - 0,828925x + 62,2272;$$

остаток:

$$e = 1,794.$$

Графики полученных функций вместе с экспериментальными данными представлены на рис. 1.

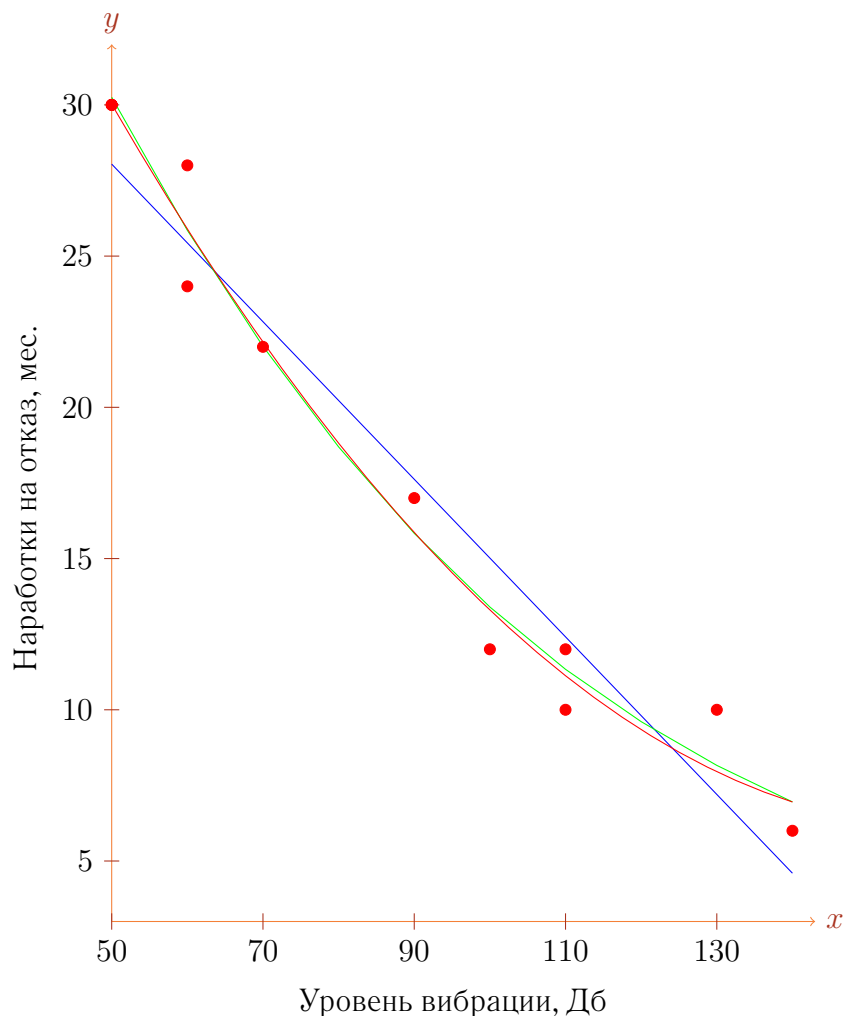


Рис. 1. Аппроксимация степенными моделями.

Вот теперь самое время обсудить вопрос о том, как пролегает график аппроксимирующего многочлена в облаке точек. Из рис. видно, что он проходит «где-то в середине» этого облака! Если облако напоминает эллипс, график расположится вдоль его оси, если утолщенную параболу — вдоль графика этой параболы и т. д. Щелкая мышкой на анимационном рис. 2, можно увидеть облака точек, наблюдавшихся в некоторых экспериментах (не имеющих отношения к нашему примеру), и аппроксимирующие их кривые, построенные методом наименьших квадратов.

А какой модели мы должны отдать предпочтение в нашем примере? Конечно, остаток минимален для кубической модели. Но и для параболической он имеет почти такое же значение. Поэтому есть смысл остановиться на более простой, параболической, модели, обеспечивающей к тому же достаточно ма-

Рис. 2. Приближение методом наименьших квадратов.

лый остаток.

Из этого следует также практическое правило, которое заключается в том, что выбор модели должен основываться на трех вещах: 1) на анализе остатков; 2) на простоте модели; 3) на согласованности математической модели с теми представлениями, которые считаются приемлемыми в области знаний, для которой предназначена модель.

Последний пункт можно пояснить в рамках разобранного примера. То, что повышение уровня вибрации плохо сказывается на работе аппаратуры, мало у кого вызывает сомнения; поэтому убывающие функции, вроде тех, которые мы получили, специалистами, скорее всего, будут признаны подходящими. А вот какую-нибудь уж очень зигзагообразную кривую они вряд ли посчитают приемлемой! Впрочем, возможно, исследование ее зигзагообразности ведет к новому открытию...

Как аппроксимировать функции методом наименьших квадратов в системе *Mathematica*, рассказывается в Приложении⁵⁾.

Приложение

1)

СВОЙСТВА ПРОИЗВОДНОЙ ПО ВЕКТОРУ

1° $\frac{d\mathbf{x}}{d\mathbf{x}} = \mathbf{E}$, где \mathbf{E} – единичная матрица порядка n .

Следует из определения производной (5):

$$\frac{d\mathbf{x}}{d\mathbf{x}} = \begin{pmatrix} \frac{\partial x_1}{\partial x_1} & \dots & \frac{\partial x_n}{\partial x_1} \\ \dots & \dots & \dots \\ \frac{\partial x_1}{\partial x_n} & \dots & \frac{\partial x_n}{\partial x_n} \end{pmatrix} = \begin{pmatrix} 1 & \dots & 0 \\ \dots & \dots & \dots \\ 0 & \dots & 1 \end{pmatrix}.$$

2° Постоянную матрицу \mathbf{C} можно выносить за знак производной:

$$\frac{d(\mathbf{C}\mathbf{f})}{d\mathbf{x}} = \frac{d\mathbf{f}}{d\mathbf{x}} \mathbf{C}^T,$$

Действительно, пусть $\mathbf{C} = \{c_{ij}\}_r^m$, тогда

$$\frac{d(\mathbf{C}\mathbf{f})}{d\mathbf{x}} = \left\{ \frac{\partial (\sum_{k=1}^m c_{jk} f_k)}{\partial x_i} \right\}_n^r = \left\{ \sum_{k=1}^m \frac{\partial f_k}{\partial x_i} c_{jk} \right\}_n^r = \frac{d\mathbf{f}}{d\mathbf{x}} \mathbf{C}^T.$$

3° Производные суммы, разности и произведения двух функций:

$$\frac{d(\mathbf{f} \pm \mathbf{g})}{d\mathbf{x}} = \frac{d\mathbf{f}}{d\mathbf{x}} \pm \frac{d\mathbf{g}}{d\mathbf{x}}, \quad \frac{d(\mathbf{f}^T \mathbf{g})}{d\mathbf{x}} = \frac{d\mathbf{f}}{d\mathbf{x}} \mathbf{g} + \frac{d\mathbf{g}}{d\mathbf{x}} \mathbf{f}.$$

Первое равенство сразу следует из определения производной, а второе докажем:

$$\frac{d(\mathbf{f}^T \mathbf{g})}{d\mathbf{x}} = \left\{ \sum_{k=1}^n \frac{\partial (f_k g_k)}{\partial x_i} \right\}_n^1 = \left\{ \sum_{k=1}^n \frac{\partial f_k}{\partial x_i} g_k + \sum_{j=1}^n \frac{\partial g_k}{\partial x_i} f_k \right\}_n^1 = \frac{d\mathbf{f}}{d\mathbf{x}} \mathbf{g} + \frac{d\mathbf{g}}{d\mathbf{x}} \mathbf{f}.$$

4° Производная сложной функции:

$$\frac{d\mathbf{f}[\mathbf{g}(\mathbf{x})]}{d\mathbf{x}} = \frac{d\mathbf{g}}{d\mathbf{x}} \cdot \frac{d\mathbf{f}}{d\mathbf{g}}.$$

В самом деле, пусть \mathbf{f} имеет p аргументов, тогда матрица производной сложной функции получается как сумма произведений i -й строки матрицы $d\mathbf{g}/d\mathbf{x}$ на j -й столбец матрицы $d\mathbf{f}/d\mathbf{g}$:

$$\frac{d\mathbf{f}[\mathbf{g}(\mathbf{x})]}{d\mathbf{x}} = \left\{ \frac{\partial \{f_j[\mathbf{g}(\mathbf{x})]\}}{\partial x_i} \right\}_n^m = \left\{ \sum_{k=1}^p \frac{\partial \{f_j[\mathbf{g}(\mathbf{x})]\}}{\partial g_k} \cdot \frac{\partial g_k(\mathbf{x})}{\partial x_i} \right\}_n^m = \frac{d\mathbf{g}}{d\mathbf{x}} \cdot \frac{d\mathbf{f}}{d\mathbf{g}}.$$

2) Формула (6). Применяя первое и второе свойства производной, получаем

$$\frac{d(\mathbf{c}^T \mathbf{x})}{d\mathbf{x}} = \frac{d\mathbf{x}}{d\mathbf{x}} \mathbf{c} = \mathbf{E} \mathbf{c} = \mathbf{c}.$$

Так как $\mathbf{c}^T \mathbf{x}$ – скаляр, то $\mathbf{x}^T \mathbf{c} = (\mathbf{x}^T \mathbf{c})^T = \mathbf{c}^T \mathbf{x}$. Используем это и уже доказанное равенство:

$$\frac{d(\mathbf{x}^T \mathbf{c})}{d\mathbf{x}} = \frac{d(\mathbf{c}^T \mathbf{x})}{d\mathbf{x}} = \mathbf{c}.$$

Формула (7). Доказывается аналогично:

$$\frac{d(\mathbf{Ax})}{d\mathbf{x}} = \frac{d\mathbf{x}}{d\mathbf{x}} \mathbf{A}^T = \mathbf{EA}^T = \mathbf{A}^T.$$

Формула (8). Применим производную произведения, представив квадратичную форму $\mathbf{x}^T \mathbf{Ax}$ в виде произведения строки \mathbf{x}^T на столбец \mathbf{Ax} :

$$\frac{d(\mathbf{x}^T \mathbf{Ax})}{d\mathbf{x}} = \frac{d\mathbf{x}}{d\mathbf{x}} \mathbf{Ax} + \frac{d(\mathbf{Ax})}{d\mathbf{x}} \mathbf{x} = \mathbf{EAx} + \mathbf{A}^T \mathbf{x} = (\mathbf{A}^T + \mathbf{A}) \mathbf{x}.$$

Формула (9). Следует из предыдущего равенства, потому что для симметричной матрицы $\mathbf{A}^T = \mathbf{A}$.

3) Так как величина $\mathbf{Y}^T \mathbf{R} \Phi \beta$ скалярна, то

$$\beta^T \Phi^T \mathbf{R} \mathbf{Y} = (\beta^T \Phi^T \mathbf{R} \mathbf{Y})^T = \mathbf{Y}^T \mathbf{R} \Phi \beta. \quad (\text{П1})$$

Раскроем скобки в правой части выражения (3) и используем свойство (П1):

$$\begin{aligned} S &= \mathbf{Y}^T \mathbf{R} \mathbf{Y} - \beta^T \Phi^T \mathbf{R} \mathbf{Y} - \mathbf{Y}^T \mathbf{R} \Phi \beta + \beta^T \Phi^T \mathbf{R} \Phi \beta = \\ &= \mathbf{Y}^T \mathbf{R} \mathbf{Y} - 2\beta^T \Phi^T \mathbf{R} \mathbf{Y} + \beta^T \Phi^T \mathbf{R} \Phi \beta. \end{aligned} \quad (\text{П2})$$

Обозначим $\mathbf{A} = \Phi^T \mathbf{R} \Phi$, $\mathbf{B} = \Phi^T \mathbf{R} \mathbf{Y}$ и предположим, что матрица \mathbf{A} невырождена. Тогда с учетом симметричности \mathbf{A} и формулы (П1) равенство (П2) перепишется так:

$$\begin{aligned} S &= \mathbf{Y}^T \mathbf{R} \mathbf{Y} + \beta^T \mathbf{A} \beta - 2\beta^T \mathbf{B} = \\ &= \mathbf{Y}^T \mathbf{R} \mathbf{Y} + (\beta - \mathbf{A}^{-1} \mathbf{B})^T \mathbf{A} (\beta - \mathbf{A}^{-1} \mathbf{B}) - \mathbf{B}^T \mathbf{A}^{-1} \mathbf{B}. \end{aligned} \quad (\text{П3})$$

Поскольку матрица \mathbf{A} положительно определена (см. (12)), то

$$(\beta - \mathbf{A}^{-1} \mathbf{B})^T \mathbf{A} (\beta - \mathbf{A}^{-1} \mathbf{B}) \geq 0$$

для любого вектора β , причем, нуль получается при $\beta = \mathbf{A}^{-1} \mathbf{B}$. Это значит, что величина (П3) не может быть меньше, чем $\mathbf{Y}^T \mathbf{R} \mathbf{Y} - \mathbf{B}^T \mathbf{A}^{-1} \mathbf{B}$, и этот минимум достигается при $\beta = \mathbf{A}^{-1} \mathbf{B}$, или

$$\beta = (\Phi^T \mathbf{R} \Phi)^{-1} \Phi^T \mathbf{R} \mathbf{Y}.$$

4) Доказательство проведем от противного. Предположим, что матрица $\Phi^T \Phi$ вырождена, и перепишем уравнение (10) с учетом того, что $\mathbf{R} = \mathbf{E}$, в виде

$$\Phi^T \Phi \beta = \Phi^T \mathbf{Y}.$$

Так как матрица $\Phi^T \Phi$ вырождена, то соответствующее этому уравнению однородное уравнение

$$\Phi^T \Phi \beta = \mathbf{0}$$

должно иметь ненулевое решение β . Далее будем считать, что β и есть это ненулевое решение.

Детализируем однородное уравнение:

$$\Phi^T \Phi \beta = \begin{pmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \\ \dots & \dots & \dots & \dots \\ x_1^{k-1} & x_2^{k-1} & \dots & x_n^{k-1} \end{pmatrix} \begin{pmatrix} 1 & x_1 & \dots & x_1^{k-1} \\ 1 & x_2 & \dots & x_2^{k-1} \\ \dots & \dots & \dots & \dots \\ 1 & x_n & \dots & x_n^{k-1} \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \dots \\ \beta_k \end{pmatrix} =$$

$$\begin{aligned}
&= \begin{pmatrix} n & \sum_{s=1}^n x_s & \dots & \sum_{s=1}^n x_s^{k-1} \\ \sum_{s=1}^n x_s & \sum_{s=1}^n x_s^2 & \dots & \sum_{s=1}^n x_s^k \\ \dots & \dots & \dots & \dots \\ \sum_{s=1}^n x_s^{k-1} & \sum_{s=1}^n x_s^k & \dots & \sum_{s=1}^n x_s^{2k-2} \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \dots \\ \beta_k \end{pmatrix} = \\
&= \begin{pmatrix} \beta_1 n + \beta_2 \sum_{s=1}^n x_s + \dots + \beta_k \sum_{s=1}^n x_s^{k-1} \\ \beta_1 \sum_{s=1}^n x_s + \beta_2 \sum_{s=1}^n x_s^2 + \dots + \beta_k \sum_{s=1}^n x_s^k \\ \dots \\ \beta_1 \sum_{s=1}^n x_s^{k-1} + \beta_2 \sum_{s=1}^n x_s^k + \dots + \beta_k \sum_{s=1}^n x_s^{2k-2} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \dots \\ 0 \end{pmatrix}.
\end{aligned}$$

Или в более компактной форме:

$$\sum_{j=1}^k \beta_j \sum_{s=1}^n x_s^{i+j-2} = 0, \quad i = \overline{1, k}.$$

Умножим первое уравнение этой системы на β_1 , второе — на β_2 , и т. д., последнее — на β_k , и все полученные таким образом уравнения сложим:

$$\sum_{i=1}^k \beta_i \sum_{j=1}^k \beta_j \sum_{s=1}^n x_s^{i+j-2} = 0,$$

или

$$\begin{aligned}
\sum_{i=1}^k \beta_i \sum_{j=1}^k \beta_j \sum_{s=1}^n x_s^{i-1} x_s^{j-1} &= \sum_{s=1}^n \left(\sum_{i=1}^k \beta_i x_s^{i-1} \right) \left(\sum_{j=1}^k \beta_j x_s^{j-1} \right) = \\
&= \sum_{s=1}^n \left(\sum_{i=1}^k \beta_i x_s^{i-1} \right)^2 = \sum_{s=1}^n w^2(x_s) = 0,
\end{aligned}$$

где $w(x)$ — многочлен (16).

Последнее равенство означает, что $w(x_1) = w(x_2) = \dots = w(x_n) = 0$, так что многочлен $w(x)$ степени $k-1 < n$, не все коэффициенты которого нули, имеет n корней. Поскольку такого быть не может, то полученное противоречие и доказывает невырожденность матрицы $\Phi^T \Phi$.

5) Реализация метода наименьших квадратов в системе *Mathematica* представлена рядом операторов. Мы рассмотрим лишь один из них, оператор `Fit[data, funcs, vars]`, аргументы которого имеют следующий смысл: `data` — пары наблюдений за поведением аппроксимируемой функции, `funcs` — список аппроксимирующих функций, `vars` — аргументы этих функций.

Решим, например, с помощью этого оператора задачу о наработках на отказ, рассмотренную на лекции. Для этого сначала введем данные наблюдений:

```
data={{50,30},{60,24},{60,28},{70,22},{90,17},{100,12},{110,12},{110,10},
      {130,10},{140,6}};
```

а затем применим `Fit`, построив сразу четыре степенные модели и вычислив их остатки. Заодно познакомимся с оператором `Sum`, который можно выщелкнуть из палитры инструментов в виде символа суммирования, и оператором `Length[list]`, подсчитывающим количество элементов списка `list`. Линейная модель:

```
n=Length[data];
line=Fit[data,{1,x},x]
```

$$le = \frac{1}{n} \sum_{i=1}^n (\text{data}_{[[i,2]]} - \text{line}/.x \rightarrow \text{data}_{[[i,2]])}^2$$

41.0662 - 0.260502 x
3.84338

Параболическая модель:

$$\text{quadr} = \text{Fit}[\text{data}, \{1, x, x^2\}, x]$$

$$qe = \frac{1}{n} \sum_{i=1}^n (\text{data}_{[[i,2]]} - \text{quadr}/.x \rightarrow \text{data}_{[[i,2]])}^2$$

56.5078 - 0.627405 x + 0.00196075 x²
1.80903

Кубическая модель:

$$\text{cubic} = \text{Fit}[\text{data}, \{1, x, x^2, x^3\}, x]$$

$$ce = \frac{1}{n} \sum_{i=1}^n (\text{data}_{[[i,2]]} - \text{cubic}/.x \rightarrow \text{data}_{[[i,2]])}^2$$

62.2272 - 0.828925 x + 0.00417015 x² - 7.63808*10⁻⁶ x³
1.7938

Модель восьмой степени:

$$\text{octa} = \text{Fit}[\text{data}, \{1, x, x^2, x^3, x^4, x^5, x^6, x^7, x^8\}, x]$$

$$oe = \frac{1}{n} \sum_{i=1}^n (\text{data}_{[[i,2]]} - \text{octa}/.x \rightarrow \text{data}_{[[i,2]])}^2$$

-6686.97 + 522.482 x - 16.1376 x² + 0.242424 x³ - 0.0015346 x⁴
- 3.50833*10⁻⁶ x⁵ + 1.13269*10⁻⁷ x⁶ - 6.34144*10⁻¹⁰ x⁷ + 1.20928*10⁻¹² x⁸
1.

Построим графики полученных приближений к неизвестной функции (рис. 3):

```
Show[ListPlot[data, PlotStyle -> {PointSize[Medium], Red}],
      Plot[{line, quadr, cubic, octa}, {x, 50, 140},
          PlotLegends -> Placed[{"line", "quadr", "cubic", "octa"}, {0.75, 0.82}]],
      PlotRange -> {0, 31}, AxesOrigin -> {50, 3}, AspectRatio -> 4/5]
```

Как видим, минимальный остаток получен для модели 8-й степени, причем, меньший остаток получить невозможно, так как график найденного многочлена 8-й степени проходит точно через заданные точки, за исключением двух пар точек с одинаковыми абсциссами и разными ординатами. Но и здесь график пересекает середины отрезков, соединяющих пары точек! Так что получить остаток, меньший, чем 1, невозможно!

Значит, многочлен 8-й степени и является наилучшим многочленом? Как сказать... С точки зрения МНК — безусловно. А вот с точки зрения смысла решаемой задачи — вряд ли. Ведь полученную кривую надо как-то объяснить в терминах эксплуатации оборудования определенного вида. А как обосновать все «американские горки» графика? Поэтому в свете правила, предложенного в конце лекции, все же следует считать наиболее приемлемой квадратичную аппроксимацию.

Получается, математика бессильна найти наилучшую модель в рассмотренной задаче? Так ведь задача так и не формулировалась! Математика прекрасно решает проблему подбора коэффициентов аппроксимирующего многочлена *заданной степени* — и не более того. Других

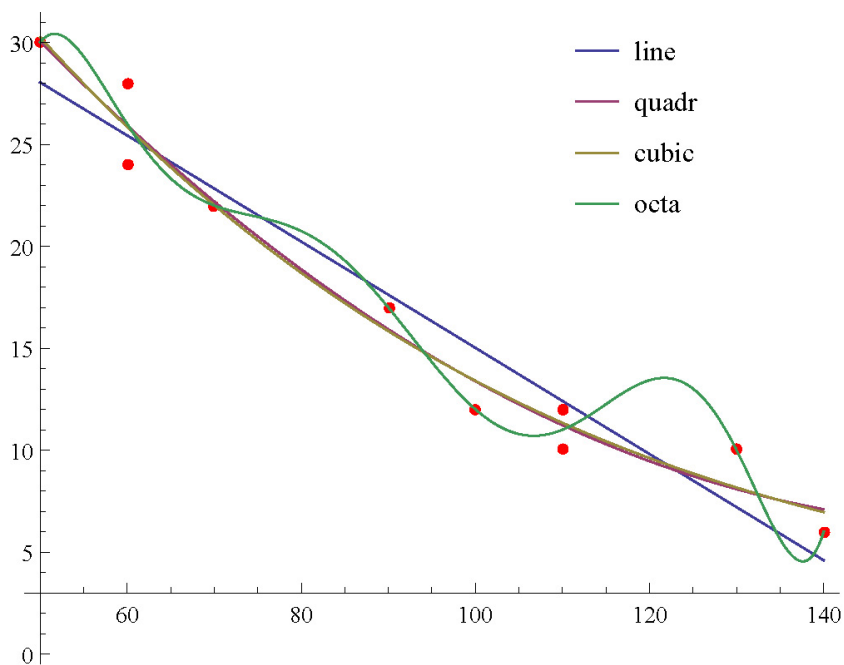
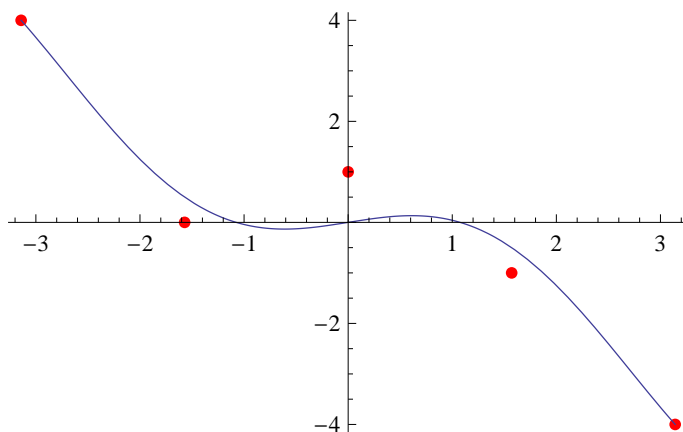


Рис. 3.

задач мы перед ней не ставили. Это уже мы сами взялись рассуждать, какой же многочлен считать наилучшим, а это — область не математики, а инженерных приложений! А там свои понятия и правила игры, которые я и попытался сформулировать.

Вернемся, однако, к оператору `Fit`. Он, конечно, может строить не только степенные модели. Например, можно взять систему функций из синусов и аппроксимировать данные их линейной комбинацией, поручив системе *Mathematica* подобрать подходящие коэффициенты этой комбинации:

```
data={{-π, 4}, {-π/2, 0}, {0, 1}, {π/2, -1}, {π, -4}};
Fit[data, {Sin[x/2], Sin[x], Sin[2 x]}, x]
0. - 4. Sin[x/2] + 2.32843 Sin[x]
Show[ListPlot[data, PlotStyle -> {PointSize[Medium], Red}], Plot[%, {x, -π, π}]]
```

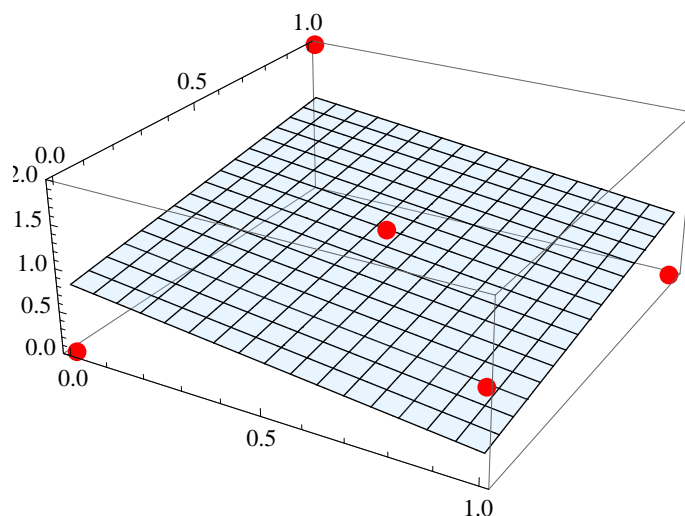


Mathematica вполне успешно выполнила задание и в связи с этим снова возникает вопрос, какую же систему функций следует предпочесть? Может быть, лучший результат дадут какие-нибудь логарифмические или показательные функции? И опять придется повторить,

что к математике такой вопрос не имеет никакого отношения! Только опыт исследователя, его интуиция, а, может быть, и случай могут подсказать ему, какую систему функций надо выбрать для использования МНК.

Оператор `Fit` в состоянии приблизить и трехмерные данные, например, плоскостью:

```
data={{0,0,0},{1,0,1},{0,1,2},{1,1,0},{1/2,1/2,1}};  
plane=Fit[data,{1,x,y},{x,y}]  
0.8 - 0.5x + 0.5y  
Show[Plot3D[plane,{x,0,1},{y,0,1},PlotStyle -> Opacity[.5],  
PlotRange -> {0,2}],Graphics3D[{Red,PointSize[Large],Map[Point,data]}]]
```



Оператор `Map` применяет свой первый аргумент ко всем элементам списка, которым является второй аргумент. Таким образом в данном случае изображаются точки (`Point`) с координатами, помещенными в список `data`.

Литература

- [1] Пискунов Н.С. *Дифференциальное и интегральное исчисления для втузов*. – М.: Наука, 1985, т. 1. – с. 276-280.