Техническая статистика

Техническая статистика. Что это такое?! Что автор, а вслед за ним и читатели будут понимать под этим термином?! Когда говорят просто о статистике, то сразу вспоминают избитую сентенцию о том, что бывает "большая ложь, маленькая ложь и... статистика". Но, шутки в сторону. В данной главе будет рассказано, как можно применить встроенные функции Mathcad, относящиеся к категории "Статистика", для решения некоторых инженерных задач — восстановление формул по табличным и графическим данным и др. Этот класс задач мы и будем условно называть *технической статистикой* — применение статистических функций для решения некоторых технических задач. На интернет-форум Mathcad (**www.exponenta.ru/forum** и **http://collab.mathsoft.com**) пользователи очень часто спрашивают, как решить такую задачу — восстановление функции по графикам и таблицам. В данной главе читатель найдет несколько практических советов и ссылки на соответствующие ресурсы Интернета.

> ***Примечание***
> К категории "Статистика" можно отнести функции из групп **Аппроксимация и сглаживание**, **Интерполяция и прогнозирование** и **Статистика** диалогового окна **Вставка функции**.

1. От графика к формуле

Очень часто в технической литературе функциональные зависимости даются не *формулами*, а *графиками*. На рис. 1 в качестве примера представлен один из таких графиков, отражающий влияние скорости воды и ее температуры на удельное гидравлическое сопротивление в фильтре, через который воду прокачивают. Рисунок взят из технической документации одной известной фирмы, поставляющей водоочистное оборудование[1].

[1] Автор работает на кафедре технологии воды и топлива Московского энергетического института. Отсюда, как понимает читатель, такое внимание воде.

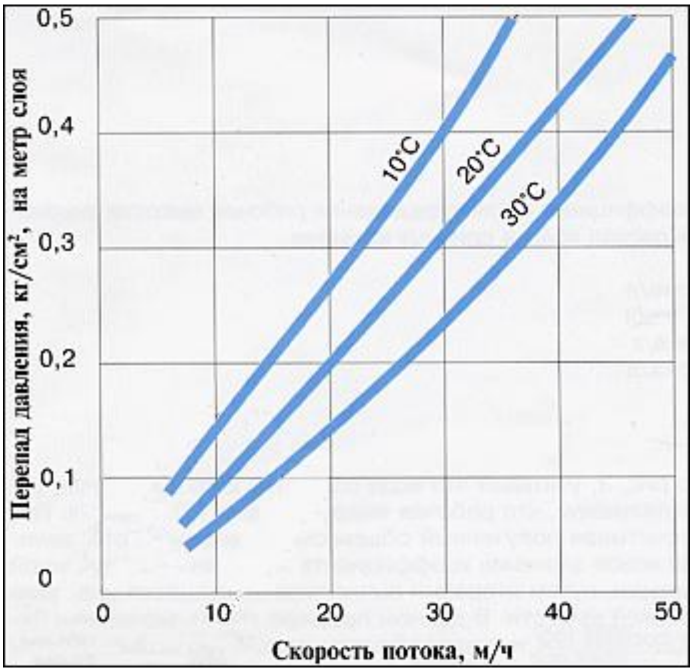


Рис. 1. Пример графика из технической литературы

Подобные графики приводятся не только для *качественного* описания тех или иных явлений (гидравлическое сопротивление растет при увеличении скорости и падает при повышении температуры, если говорить о рис. 1), но и для их *количественной* оценки — для расчетов. В упомянутой фирменной документации описан расчет этого гидравлического сопротивления по методике "вождения пальцем по графику": отложите по оси абсцисс значение скорости (первый аргумент), мысленно проведите недостающую кривую (изотерму — второй аргумент) и считайте ответ (значение функции двух аргументов) на оси ординат.

В технической литературе (особенно в справочной) встречаются также и разного рода *номограммы* с инструкциями такого рода: отложите значение первого аргумента на левой шкале, а второго — на правой; соедините точки линейкой и на средней шкале считайте ответ. По адресу **http://twt.mpei.ac.ru/MAS/Worksheets/Boiler/Th_C_B_Nom_2.mcd** хранится образец такой номограммы, моделирующей функцию уже трех аргументов (рис. 2 с "живой" данной диаграммой; "живой" в том смысле, что изменение в полях ввода скажется на номограмме после нажатия кнопки **Recalculate** (Пересчитать)).

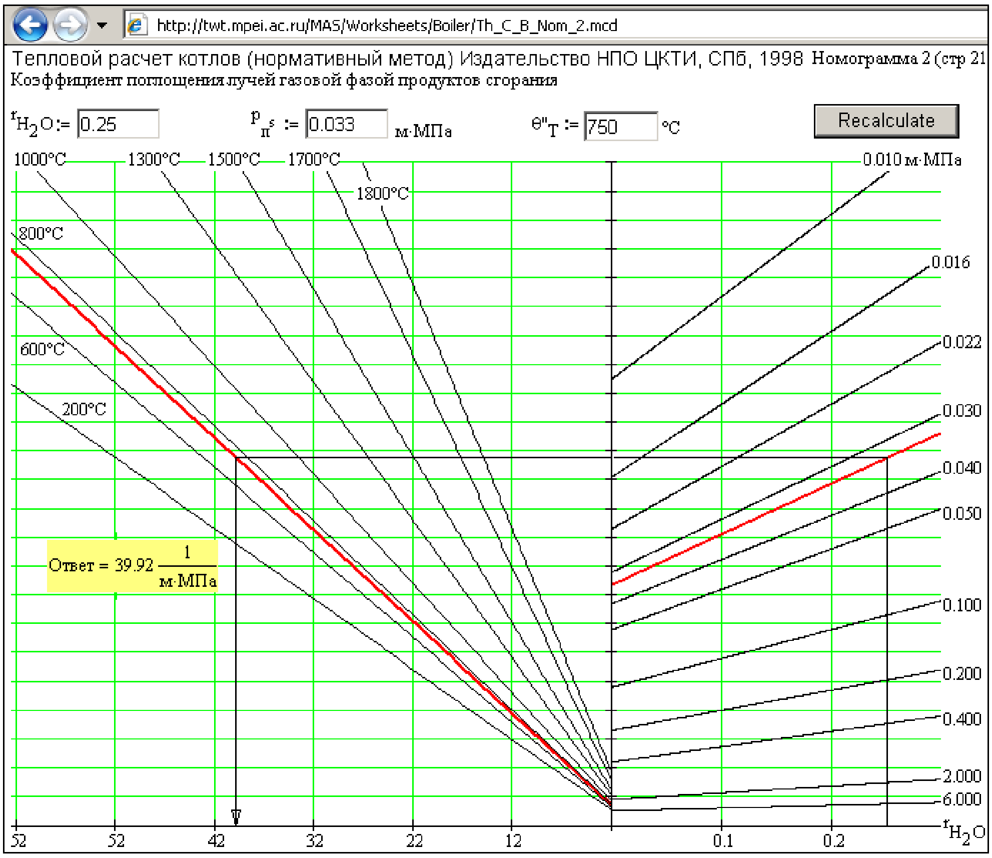


Рис. 2. Пример "живой" номограммы в Интернете

В номограммах (а на них выросло целое поколение инженеров) "тонет" физика задачи — ее качественная оценка, зато повышается точность таких "графических" расчетов. Выпускались даже нехитрые механические устройства типа логарифмической линейки с вшитыми в них алгоритмами расчетов. Такие устройства были особо популярны у штурманов, прокладывающих маршруты морских и воздушных судов до наступления эры бортовых компьютеров и глобальных систем позиционирования. Сейчас что-то подобное можно купить в газетных киосках — совместил на двух дисках свой вес и рост и узнал, пора ли переходить на диету...

Но для современных расчетов с использованием компьютеров или просто калькуляторов более подходят не графики и номограммы, а *формулы*, которые часто не приводятся в технической литературе по следующим причинам.

Во-первых, формулы не даются из благих намерений освободить читателя от расчетов. Тем более это зачастую и не расчет в привычном понимании этого слова, а некая *оценка*, *прикидка* того или иного параметра. В той же документации, откуда взят рис. 1, рекомендовано при выборе насоса для фильтра (а его напор — это произведение удельного гидравлического сопротивления на высоту фильтрующего материала) увеличить расчетное гидравлическое сопротивление на 10—20% (так называемый *инженерный запас*, нивелирующий помимо прочего и ошибки считывания "пальцем" чисел с графика).

Во-вторых, нередко никакой формулы не было, и нет, т. к. на графиках даны результаты некой *графической обработки* опытных точек. Кривые, показанные на рис. 1, получены после испытания фильтрующего материала на специальном стенде, где есть возможность менять скорость потока, а так же температуру воды и замерять перепады давления. В научной же (не технической) литературе считается хорошим тоном оставлять на графике экспериментальные точки и показывать различного рода доверительные интервалы. В последнее время получает распространение практика ссылок из научных статей на сайт, где хранятся первичные протоколы опытов, по которым читатель (оппонент) может не только проверить выводы автора, но и дать свою трактовку результатов. Можно идти дальше и делать ссылки на программу с расчетом по этому графику. На бумаге (в технической документации — см. рис. 1) видна качественная картина явления, а на сайте, поддерживающем эту документацию, прописан соответствующий расчет. Для этого можно:

- попытаться связаться с автором и попросить его дать формулу, если она, конечно, есть;
- вывести самому нужную формулу, опираясь на "физику" задачи;

> ***Примечание***
>
> В нашем случае сопротивление может зависеть от скорости в степени близкой к квадратной.

- провести *интерполяцию сплайнами* или аппроксимацию по заданной формуле, например, т. е. сделать то, о чем пойдет речь в данной главе.

В среде Mathcad есть встроенные функции `lspline`, `pspline` и `cspline` для сплайн-интерполяции табличных зависимостей функций одного или двух аргументов.

> ***Примечание***
>
> Перечисленные функции возвращают коэффициенты интерполяционного полинома. Сама же интерполяция ведется через универсальную функцию `interp` (см. рис. 3).

Но работа с этими функциями при двух аргументах (наша задача, зафиксированная на рис. 1) затруднена из-за того, что эти функции требуют "квадратных" исходных табличных данных, где число точек по первому аргументу равно числу точек по второму аргументу. Реальные же данные на графиках, как правило, ложатся в прямоугольную (далеко не квадратную) таблицу. Из-за этого приходится либо искусственно "оквадрачивать" исходные табличные данные, убирая из матрицы некоторые строки или столбцы, что влечет за собой потерю точности (прямоугольник сводится к квадрату по наименьшей стороне), либо менять одну двумерную интерполяцию на две одномерные.

На рис. 3 показан универсальный Mathcad-документ автоматизации работы с семейством кривых.

> ***Примечание***
>
> Чтобы данная функция была универсальной и по внешнему виду, ей стоит дать имя `f(x, y)`, не привязывая имена аргументов и функции к конкретной задаче со скоростью, температурой и удельным перепадом давления, как в нашем случае.

Его универсальность в том, что в матрице исходных данных можно произвольно менять значения элементов, а также число строк и столбцов. Левая "девятка"[2] матрицы (элемент с индексом 0, 0[3]) хранит названия боковика и шапки таблицы. Программно создается функция пользователя с именем ΔP_y (удельный перепад давления) и с двумя аргументами: `t` — температура и `v` — скорость потока.

---

[2] Футбольный термин, если кто не знает или забыл. "Девятка" — верхний угол ворот, а "шестерка" — нижний.

[3] "Верхний" элемент вектора и "левый верхний" угол матрицы могут иметь и другую нумерацию 1 и 1, 1, например, что определяется значением переменной `ORIGIN`. Но автор взял за правило никогда "не трогать" нулевого значения этой переменной, данного по умолчанию. Кстати, в среде Mathcad 12/13/14 значение переменной `ORIGIN` может влиять и на нумерацию символов в текстовых константах.

Единицы измерения

$ч := hr \quad м := m \quad кгс := kgf \quad см := cm \quad °C(t) := (t + 273.15)K$

Единицы измерения

Исходные данные

Скорость воды $v := 25\,\frac{м}{ч}$ Температура $t := 15\,°C$

Исходные данные

Расчет перепада давления

$$\Delta P_y(t,v) := \left| \begin{array}{l} \left(t \leftarrow \frac{t}{K} \quad v \leftarrow v \cdot \frac{hr}{m} \quad \text{"Лишаем аргументы размерности"} \right) \\ M \leftarrow \begin{pmatrix} \text{"T (K)\v (м/ч)"} & 10 & 20 & 30 & 40 & 50 \\ 10 + 273.15 & 0.140 & 0.273 & 0.407 & 0.580 & 0.850 \\ 20 + 273.15 & 0.092 & 0.204 & 0.309 & 0.428 & 0.549 \\ 30 + 273.15 & 0.059 & 0.141 & 0.230 & 0.342 & 0.470 \end{pmatrix} \\ T \leftarrow \text{submatrix}(M, 1, \text{rows}(M) - 1, 0, 0) \\ V \leftarrow (\text{submatrix}(M, 0, 0, 1, \text{cols}(M) - 1))^T \\ \Delta P \leftarrow \text{submatrix}(M, 1, \text{rows}(M) - 1, 1, \text{cols}(M) - 1)^T \\ \text{for } \; i \in 0 .. \text{cols}(\Delta P) - 1 \\ \quad \Delta P'_i \leftarrow \text{interp}\left(\text{cspline}\left(V, \Delta P^{\langle i \rangle}\right), V, \Delta P^{\langle i \rangle}, v\right) \\ \text{"Возвращаем функцию с размерностью"} \\ \text{interp}(\text{cspline}(T, \Delta P'), T, \Delta P', t) \cdot \frac{kgf}{cm^2 m} \end{array} \right.$$

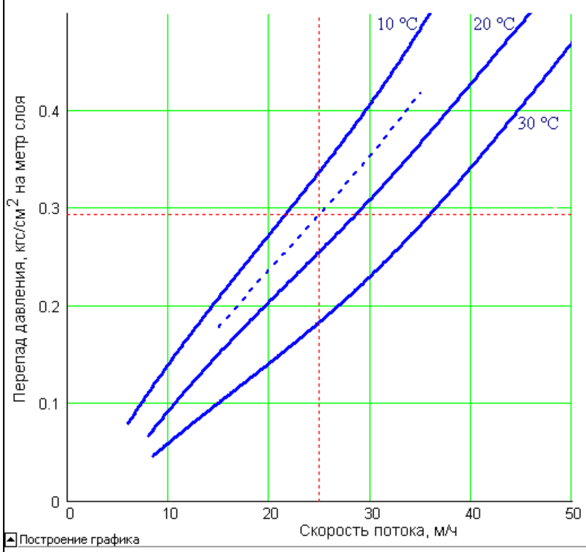
$\Delta P_y := \Delta P_y(t, v)$

Расчет перепада давления

Построение графика

$v_{10} := 6\,\frac{м}{ч}, 6.1\,\frac{м}{ч} .. 50\,\frac{м}{ч} \qquad v_{20} := 8\,\frac{м}{ч}, 8.1\,\frac{м}{ч} .. 50\,\frac{м}{ч} \qquad v_{30} := 8.5\,\frac{м}{ч}, 8.6\,\frac{м}{ч} .. 50\,\frac{м}{ч}$

$v_t := v - 10\,\frac{м}{ч}, v - 9.9\,\frac{м}{ч} .. v + 10\,\frac{м}{ч} \qquad := \frac{\Delta P_y(t,v)}{\frac{kgf}{cm^2 m}} \qquad := \frac{v}{\frac{m}{hr}}$



Построение графика

Вывод ответа

$\Delta P_y = 0.285\,\frac{atm}{м} \qquad \Delta P_y = 1.278\,\frac{psi}{ft} \qquad v = 10.2\,\frac{gpm}{ft^2}$

Вывод ответа

Рис. 3. Программа трехмерной сплайн-интерполяции

Функция ΔP_y содержит в матрице M опорные точки интерполяции (результаты обмера графика на бумаге обычной деревянной, а лучше — прозрачной пластиковой линейкой или обмера отсканированного или "выуженного" из Интернета графика на экране дисплея "штангенциркулем" какого-либо графического редактора[4]) с боковиком (первый аргумент функции — температура) и "шапкой" (второй аргумент — скорость, см. комментарий — текстовую константу в уже упомянутой левой "девятке"). Еще одна особенность функции — возможность ее работы с размерными величинами *(см. главу 2)*, что позволяет вводить данные с любыми единицами скорости и в любых температурных шкалах и иметь ответ также в разных единицах.

> ***Примечание***
> Исходный график, показанный на рис. 1, в английском варианте единиц измерения, кстати, имел ошибку — скорость потока за океаном измеряется не в gpm/ft^3, а в gpm/ft^2 (галлон в минуту через квадратный фут). Ввод в расчет единиц измерения позволяет выявлять подобные ошибки — бумага все стерпит, но Mathcad — нет.

Данная функция позволяет не только вести интерполяцию, но и восстанавливать исходное "семейство кривых" с графической интерпретацией самого процесса интерполяции: по значению первого аргумента уже автоматически, а не мысленно строится промежуточная кривая (у нас это изотерма, проведенная пунктиром), а по значению второго — считывается и выводится с разными единицами измерения искомое значение на оси *y*. Тут сам график может показаться лишним. Но его оставляют в расчете для отображения некой динамики процесса и для контроля исходных данных.

На рис. 3 показано, как двухмерная задача сводится к одномерной генерацией вспомогательного вектора $\Delta P'$, хранящего значения перепада давления в промежуточных точках, которые были получены одномерной сплайн-интерполяцией (оператор $\Delta P'_i \leftarrow ...$). Окончательный результат получен при новой сплайн-интерполяции уже по второму аргументу — по температуре (последний оператор программы).

Если при штатной двухмерной сплайн-интерполяции, как уже отмечалось, табличные значения будущей функции нужно держать в квадратной матрице, а соответствующие им значения двух аргументов — в матрице с двумя столбцами и с числом строк, равным порядку матрицы значений функции, то в нашей задаче на рис. 3 все эти значения хранятся в одной матрице M. Это существенно упрощает ее редактирование

[4] Эти данные можно "выуживать" и из exe-файлов программ с их диалоговыми окнами, где пользователь меняет значения аргументов и видит, как при этом меняется функция. В таких программах нередко запрятаны функции одного или нескольких аргументов, формулы которых авторы программ по ряду причин не приводят в описаниях. Так вот наша методика позволяет восстановить эти формулы (своего рода хакерский промышленный шпионаж). Есть специальные приемы защиты таких данных — искусственное повышение числа аргументов (числа вводимых данных для расчета), искажение шумом промежуточных данных и т. д. Тут остро стоит вопрос автоматизации такой ручной работы.

или просто обзор. Матрица `M` встроенной функцией `submatrix` раскладывается на векторы-аргументы `T` и `V` и на матрицу-функцию Δ`P`.

Наш подход к снижению порядка задачи, показанный на рис. 3, можно применить для решения уже четырехмерной задачи — генерации с помощью сплайн-интерполяции функции трех аргументов (рис. 4).

$$M := \begin{bmatrix} \begin{pmatrix} \text{"x1="} \\ 0.2 \end{pmatrix} & \begin{pmatrix} \text{"y\z"} & 0.1 & 0.2 & 0.3 & 0.4 & 0.5 & 0.6 & 0.8 & 1 \\ 5 & 0.63 & 0.56 & 0.49 & 0.42 & 0.35 & 0.28 & 0.21 & 0.14 \\ 10 & 0.7 & 0.56 & 0.49 & 0.42 & 0.42 & 0.49 & 0.56 & 0.63 \\ 15 & 0.7 & 0.63 & 0.56 & 0.49 & 0.49 & 0.49 & 0.42 & 0.42 \\ 20 & 0.56 & 0.56 & 0.49 & 0.49 & 0.42 & 0.42 & 0.35 & 0.35 \\ 30 & 0.49 & 0.42 & 0.35 & 0.42 & 0.49 & 0.42 & 0.35 & 0.35 \end{pmatrix} \\ \begin{pmatrix} \text{"x2="} \\ 0.5 \end{pmatrix} & \begin{pmatrix} & & & & & & & & \\ 5 & 0.81 & 0.72 & 0.63 & 0.54 & 0.45 & 0.36 & 0.27 & 0.18 \\ 10 & 0.9 & 0.72 & 0.63 & 0.54 & 0.54 & 0.63 & 0.72 & 0.81 \\ 15 & 0.9 & 0.81 & 0.72 & 0.63 & 0.63 & 0.63 & 0.54 & 0.54 \\ 20 & 0.72 & 0.72 & 0.63 & 0.63 & 0.54 & 0.54 & 0.45 & 0.45 \\ 30 & 0.63 & 0.54 & 0.45 & 0.54 & 0.63 & 0.54 & 0.45 & 0.45 \end{pmatrix} \\ \begin{pmatrix} \text{"x3="} \\ 1 \end{pmatrix} & \begin{pmatrix} & & & & & & & & \\ 5 & 0.72 & 0.64 & 0.56 & 0.48 & 0.4 & 0.32 & 0.24 & 0.16 \\ 10 & 0.8 & 0.64 & 0.56 & 0.48 & 0.48 & 0.56 & 0.64 & 0.72 \\ 15 & 0.8 & 0.72 & 0.64 & 0.56 & 0.56 & 0.56 & 0.48 & 0.48 \\ 20 & 0.64 & 0.64 & 0.56 & 0.56 & 0.48 & 0.48 & 0.4 & 0.4 \\ 30 & 0.56 & 0.48 & 0.4 & 0.48 & 0.56 & 0.48 & 0.4 & 0.4 \end{pmatrix} \end{bmatrix}^T$$

$$f(M,x,y,z) := \left| \begin{array}{l} \text{for } i \in 0..\,\mathrm{cols}(M)-1 \\ \quad X_i \leftarrow \left(M_{0,i}\right)_1 \\ \text{for } i \in 1..\,\mathrm{cols}\left(M_{1,0}\right)-1 \\ \quad \text{for } j \in 1..\,\mathrm{rows}\left(M_{1,0}\right)-1 \\ \quad\quad \left| \begin{array}{l} \text{for } k \in 0..\,\mathrm{cols}(M)-1 \\ \quad F_k \leftarrow \left(M_{1,k}\right)_{j,i} \\ M1_{j,i} \leftarrow \mathrm{interp}(\mathrm{cspline}(X,F),X,F,x) \end{array} \right. \\ Z \leftarrow \left(\mathrm{submatrix}\left(M_{1,0},0,0,1,\mathrm{cols}\left(M_{1,0}\right)-1\right)\right)^T \\ Y \leftarrow \mathrm{submatrix}\left(M_{1,0},1,\mathrm{rows}\left(M_{1,0}\right)-1,0,0\right) \\ F \leftarrow \mathrm{submatrix}(M1,1,\mathrm{rows}(M1)-1,1,\mathrm{cols}(M1)-1) \\ \text{for } i \in 0..\,\mathrm{cols}(F)-1 \\ \quad Fy_i \leftarrow \mathrm{interp}\left(\mathrm{cspline}\left(Y,F^{\langle i \rangle}\right),Y,F^{\langle i \rangle},y\right) \\ \mathrm{interp}(\mathrm{cspline}(Z,Fy),Z,Fy,z) \end{array} \right.$$

$x := 0.22 \quad y := 25 \quad z := 0.87 \qquad f(M,x,y,z) = 0.367$

Рис. 4. Программа четырехмерной сплайн-интерполяции

Для хранения исходных табличных данных четырехмерной задачи[5] нужна уже не плоская, а объемная матрица со строками, столбцами и слоями. В языках программирования такая конструкция есть (трехмерный массив: `A(i, j, k)` — BASIC, `A[i, j, k]`[6] — Pascal и C), но в Mathcad — нет. Один из выходов — использование составного массива, т. е. массива (в нашем случае матрицы), элементы которого — новые массивы (матрицы). На рис. 4 "трехмерная" таблица исходных значений для интерполяции занесена в составной массив `M`, где первый столбец хранит три вектора, элементы которых — значения аргумента `x` с комментариями, а второй — значения аргументов `y` и `z`, а также значения самой функции по структуре, уже описанной нами на рис. 3 ("девятка", шапка, боковых и пр.). Далее — дело техники: средствами программирования создается функция `f(M, x, y, z)`, где наша трехмерная задача сначала сводится к двухмерной, а затем к одномерной[7].

При решении задач интерполяции, т. е. нахождения нового значения *внутри* заданного диапазона, важно не перейти к новой задаче — к задаче *экстраполяции*, для решения которой в Mathcad есть специальные инструменты — см. сайт **http://twt.mpei.ac.ru/mas/worksheets/predict.mcd** с примером работы функции `predict` (предсказание).

Поэтому на рис. 5 показано, как в отображенную на рис. 4 программу внесены "контрольно-пропускные пункты" (КПП), отсекающие неверно заданные значения аргументов `x`, `y`, и `z` с выдачей пользовательского сообщения об ошибках, или текстового результата вместо числового (см. последнюю строку на рис. 5).

---

[5] Здесь нет единого мнения о *порядке* задачи. Иногда говорят, что это (рис. 5.4) трехмерная задача, имея в виду, что задача на рис. 5.3 — двухмерная: `y(x)` — одномерность, `z(x, y)` — двухмерность, `f(x, y, z)` — трехмерность и т. д.

[6] Отсюда, кстати, и взялась быстрая клавиша <[> ввода оператора, возвращающего элемент массива в Mathcad: `A[1` → A_1.

[7] Любимый прием математиков — сведение неизвестной задачи к известной, более сложной к простой или наоборот.... Если, например, математика попросить составить алгоритм кипячения воды в чайнике, то он скажет, что нужно налить в чайник воды, поставить его на плиту, зажечь газ и т. д. Но если дать новую задачу с вводной, что чайник уже наполнен, то математик скажет, что нужно вылить воду и тогда новая задача сведется к старой, уже решенной.

```
f(M, x, y, z) := | "4D Spline Interpolation with out of Range Checking"
                 | Or(a, b) ← a + b
                 | for  i ∈ 0.. cols(M) − 1
                 |   X_i ← (M_{0,i})_1
                 | return  error("x out of Range")  if          Or
                 |                                          /        \
                 |                                   x < min(X)   x > max(X)
                 | for  i ∈ 1.. cols(M_{1,0}) − 1
                 |   for  j ∈ 1.. rows(M_{1,0}) − 1
                 |     | for  k ∈ 0.. cols(M) − 1
                 |     |   F_k ← (M_{1,k})_{j,i}
                 |     | M1_{j,i} ← interp(cspline(X, F), X, F, x)
                 | Y ← submatrix(M_{1,0}, 1, rows(M_{1,0}) − 1, 0, 0)
                 | return  error("y out of Range")  if  (y < min(Y)) + (y > max(Y))
                 | Z ← [ submatrix(M_{1,0}, 0, 0, 1, cols(M_{1,0}) − 1) ]^T
                 | return  "z out of Range"   if  z < min(Z) ∨ z > max(Z)
                 | F ← submatrix(M1, 1, rows(M1) − 1, 1, cols(M1) − 1)
                 | for  i ∈ 0.. cols(F) − 1
                 |   Fy_i ← interp(cspline(Y, F^⟨i⟩), Y, F^⟨i⟩, y)
                 | interp(cspline(Z, Fy), Z, Fy, z)

x := 0.22    y := 25    z := 0.87     f(M, x, y, z) = 0.367
x := 1.1     y := 25    z := 0.87     f(M, x, y, z) = ▪▪
                                            x out of Range
x := 0.22    y := 31    z := 0.87     f(M, x, y, z) = ▪▪
                                            y out of Range
x := 0.22    y := 25    z := −0.1     f(M, x, y, z) = "z out of Range"
```

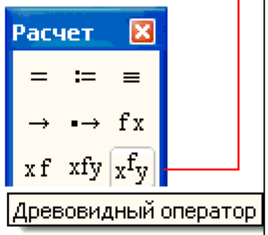


Рис. 5. Программа четырехмерной сплайн-интерполяции с КПП

Это сделано (ввод КПП для "защиты от дурака"), как уже отмечено, для того чтобы пользователь программы интерполяции не стал ненароком решать с ее помощью задачи экстраполяции. С другой стороны эти КПП могут мешать реализации обратной задачи, когда по известному значению функции необходимо найти значение аргумента. Такая задача решается с помощью встроенной функции `root` *(см. главу 2)* итерациями, некоторые из которых могут и должны выходить за установленные интервалы изменения значений аргументов, не искажая при этом конечный результат.

Интерполяция (проведение линии, поверхности и др.) строго по точкам имеет свои плюсы и минусы по сравнению и *аппроксимацией* — проведением линий, поверхностей и др. *вблизи* точек так, чтобы эти линии и поверхности отвечали определенному критерию. Обычно в качестве такого критерия выбирают *минимум суммы квадратов отклонений точек от кривой* по оси *y* — *метод наименьших квадратов*. В этом смысле задача аппроксимации является типичной оптимизационной задачей, особенности решения которой были рассмотрены в *главе 3*. В принципе для решения аппроксимационных задач достаточно иметь под рукой только функции `Minimize` и `MinErr`, приспособив их для расчета коэффициентов аппроксимирующей (сглаживающей) функции.

В Mathcad встроено достаточно много специализированных функций для решения аппроксимационных задач — задач *регрессионного анализа*, как их еще называют *(см. начало разд. 2)*. Главная такая функция так и называется — `regress`. Она возвращает коэффициенты полинома степени `n` (третий аргумент функции `regress`, первые два аргумента — это, как и в случае с функциями интерполяции, два массива исходных данных) в задачах двух- и трехмерной аппроксимации.

Работа функции `regress` в паре с универсальной функцией `interp` при решении задачи двухмерной аппроксимации показана на рис. 6.

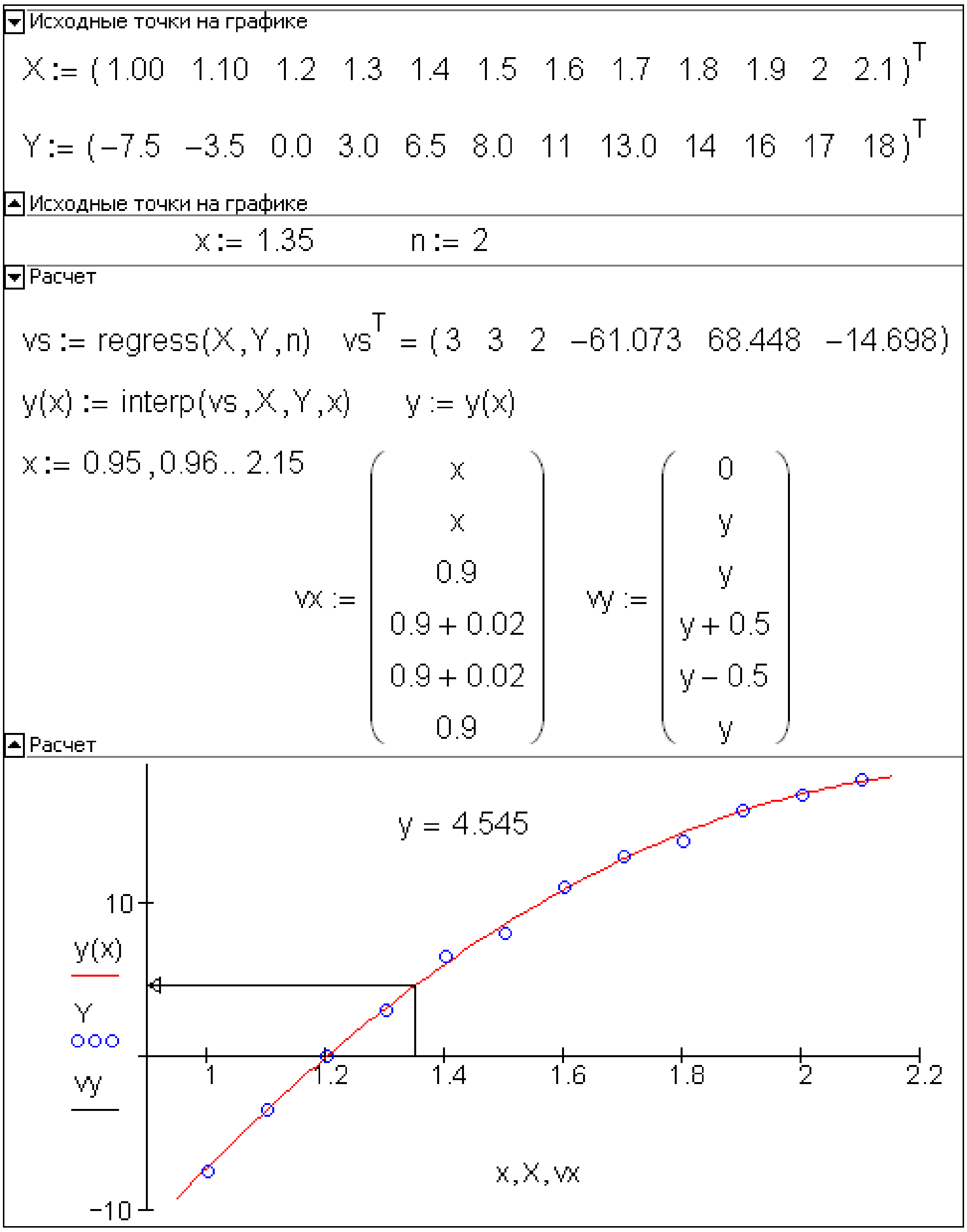


Рис. 6. 2D-сглаживание полиномом

Примечание

На рис. 6 искомая (рабочая) точка на графике фиксируется не *маркерами* (см. пунктирное перекрестие на рис. 3), а с помощью вспомогательных векторов `vx` и `vy`, "рисующих" изломанную стрелку на графике. Пар маркеров на графике всего может быть две, а стрелок — больше. Кроме того, стрелка четко показывает, где аргумент, а где функция.

Плюсы и минусы методов интерполяции и аппроксимации можно оценить при сравнении рис. 6—8, на которых через те же точки проведена не аппроксимирующая, а интерполирующая кривая: на рис. 7 со сплайн-интерполяцией, а на рис. 8 — интерполяцией полиномом предельной степени (аппрксимация перешла в интерполяцию).

Исходные точки на графике

$$X := (1.00\ \ 1.10\ \ 1.2\ \ 1.3\ \ 1.4\ \ 1.5\ \ 1.6\ \ 1.7\ \ 1.8\ \ 1.9\ \ 2\ \ 2.1)^T$$

$$Y := (-7.5\ \ -3.5\ \ 0.0\ \ 3.0\ \ 6.5\ \ 8.0\ \ 11\ \ 13.0\ \ 14\ \ 16\ \ 17\ \ 18)^T$$

Исходные точки на графике

$$x := 1.35$$

Расчет

$$vs := \text{lspline}(X, Y)$$

$$y(x) := \text{interp}(vs, X, Y, x) \qquad y := y(x)$$

$$x := 0.95, 0.96 .. 2.15$$

Расчет

$$y = 4.896$$

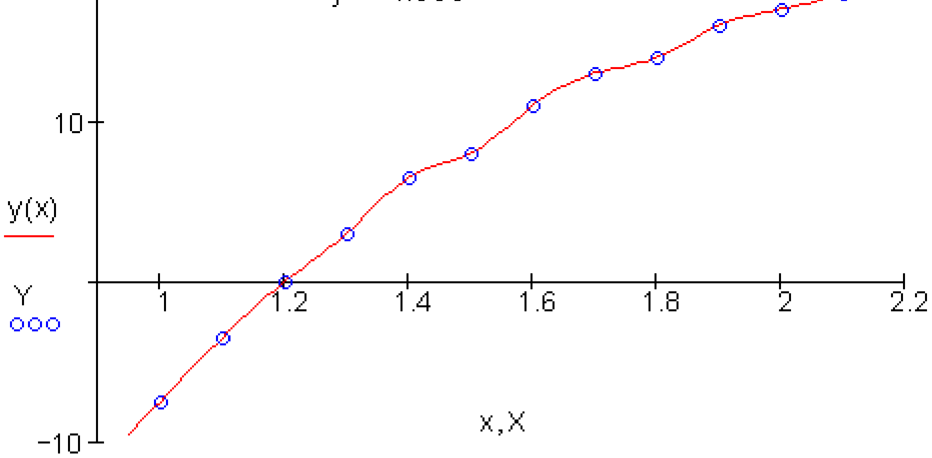


Рис. 7. 2D-интерполяция сплайном

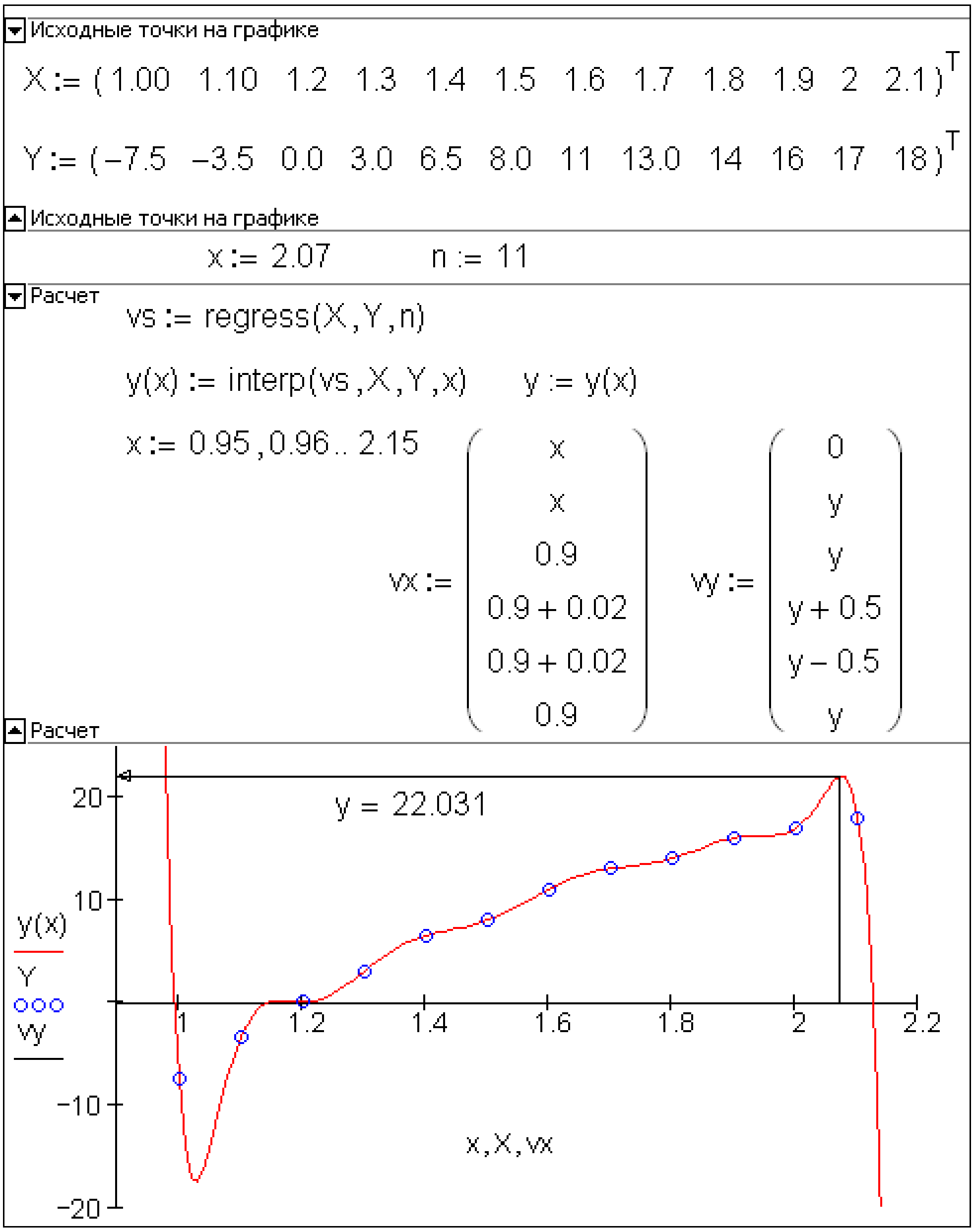


Рис. 8. 2D-интерполяция полиномом предельной степени

Вот какие можно отметить достоинства интерполяции (недостатки аппроксимации).

Достоинство интерполяции — не нужно задавать вид функции, которая будет описывать табличные значения[8]. Есть такие специальные программы для компьютеров, которым достаточно сообщить только массивы исходных данных, а они (эти программы) сами переберут множество функциональных зависимостей разного вида (степенная, логарифмическая и т. д., а также их комбинации) и сообщат пользователю, что введенные данные лучше всего описываются (сглаживаются) такой-то функцией. Критерием "лучше всего" тут может выступать минимум среднего квадратичного отклонения точек от кривой (поверхности и т. д.). Такую программу можно создать и в среде Mathcad. Но стоит ли!? Здесь главное не потерять чувство меры. Дело в том, что минимальное отклонение — это нулевое отклонение, т. е. случай, когда линия (поверхность и др.) проходит строго через точки, иными словами, когда интерполяция совпадает с аппроксимации (см. рис. 8).

Линейную интерполяцию (рис. 9) сглаживающей назвать никак нельзя, тем не менее, она широко используется для обработки табличных данных. Основной ее недостаток — это "рваная", ступенчатая производная от такой ломанной функции.

---

[8] Хотя эту функцию можно получить —
см. **http://twtmas.mpei.ac.ru/mas/Worksheets/Maple/spline.mcd**.

Исходные точки на графике

$X := (1.00\ \ 1.10\ \ 1.2\ \ 1.3\ \ 1.4\ \ 1.5\ \ 1.6\ \ 1.7\ \ 1.8\ \ 1.9\ \ 2\ \ 2.1)^T$

$Y := (-7.5\ \ -3.5\ \ 0.0\ \ 3.0\ \ 6.5\ \ 8.0\ \ 11\ \ 13.0\ \ 14\ \ 16\ \ 17\ \ 18)^T$

Исходные точки на графике

$x := 1.35 \qquad n := 2$

Расчет

$y(x) := \mathrm{linterp}(X, Y, x) \qquad y := y(x)$

$x := 0.95, 0.96 .. 2.15$

Расчет

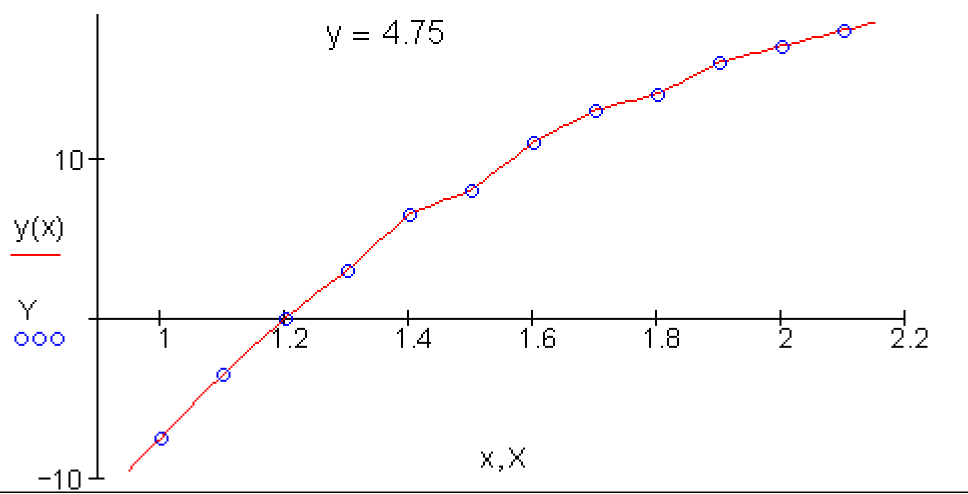


Рис. 9. 2D-линейная интерполяция

Другой крайний случай — проведение через `n` точек полинома `n-1` степени (см. рис. 8, где через 12 точек проводится полином 11-й степени).

Полином можно "гладко" дифференцировать `n` раз, но разброс его значений между точками может быть очень большим — сглаживание великолепное, но интерполяции, особенно на краях — никакой. Компромиссным вариантом решения этой проблемы и являются *сплайны* (см. рис. 3 и 7), комбинирующие линейную и полиномиальную интерполяции: внутри интервала берутся четыре точки, через которые проводится кубический полином. От него при необходимости можно "гладко" взять первую и вторую производные, которые, в свою очередь, необходимы для решения, например, оптимизационных задач градиентными методами *(см. главу 3)*. На концах интервала ("у обрыва") точек не четыре, а три или две, и здесь сплайн-интерполяция ведет себя по-разному — в зависимости от приставки корня слова "spline" данной встроенной функции: `lspline` — линейная, `pspline` — параболическая и `cspline` — кубическая (рис. 10).

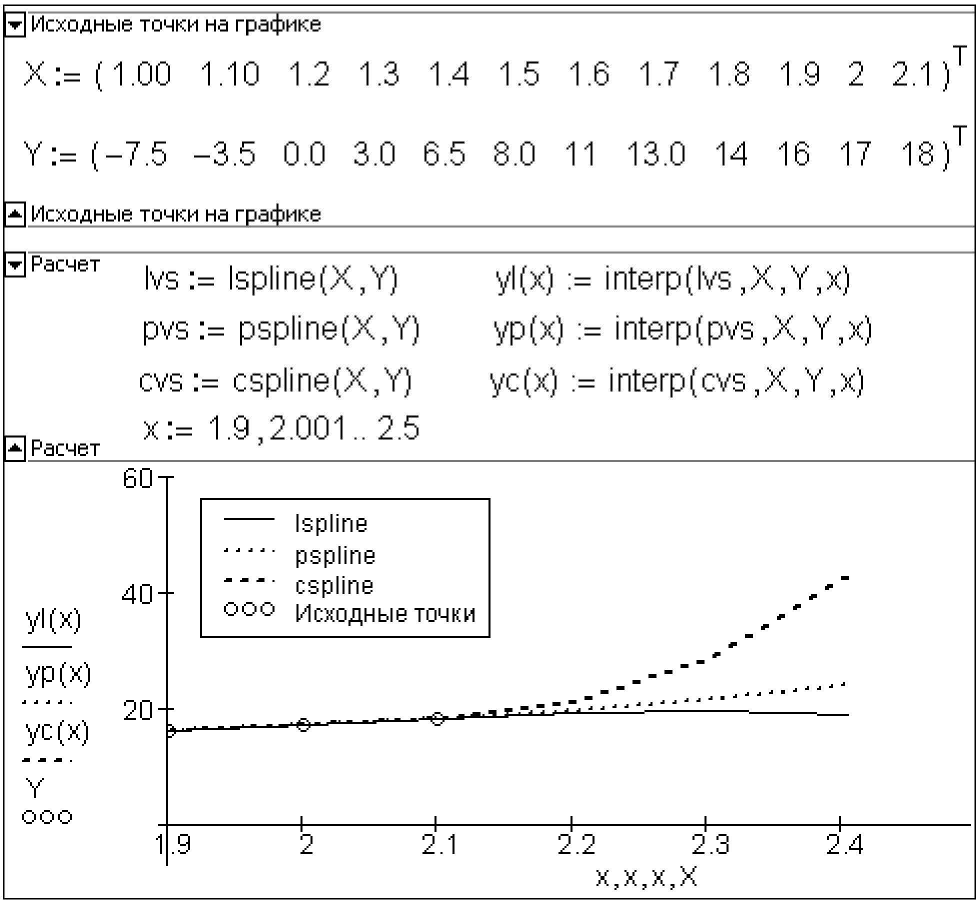


Рис. 10. 2D-экстраполяция сплайном за пределами последней точки

Возвращаясь к проблеме "как по точкам найти функцию", т. е. к вопросу, который очень часто задают на форумах Mathcad, следует выделить два случая.

❒ Человек, обрабатывающий на компьютере табличную зависимость, и переводящий дискретную функцию в разряд непрерывной, примерно представляет себе, какие физические законы в эту функцию заложены. Возвращаясь к нашему первому примеру с восстановлением функциональной зависимости перепада давления от температуры и скорости потока (см. рис. 1 и 3), можно утверждать, что перепад зависит от скорости по степенной зависимости $a \cdot x^b$, т. е. задача сглаживания сводится к нахождению этих коэффициентов, второй из которых (b) близок к двойке (значение для первого приближения, если оно потребуется для решения задачи).

- Если же человек не имеет ни малейшего понятия о виде искомой функции, то можно идти по пути сплайн-интерполяции (особенно в случае работы с точками не таблицы, а графика), сглаживанием полиномом "разумной" степени или "разумной" функцией иного вида.

На рис. 11 и 12 показаны страницы Интернета с адресами **http://twt.mpei.ac.ru/mas/worksheets/Fit_5_form.mcd** и **http://twt.mpei.ac.ru/mas/worksheets/Fit_f_x_a_b_c.mcd**.

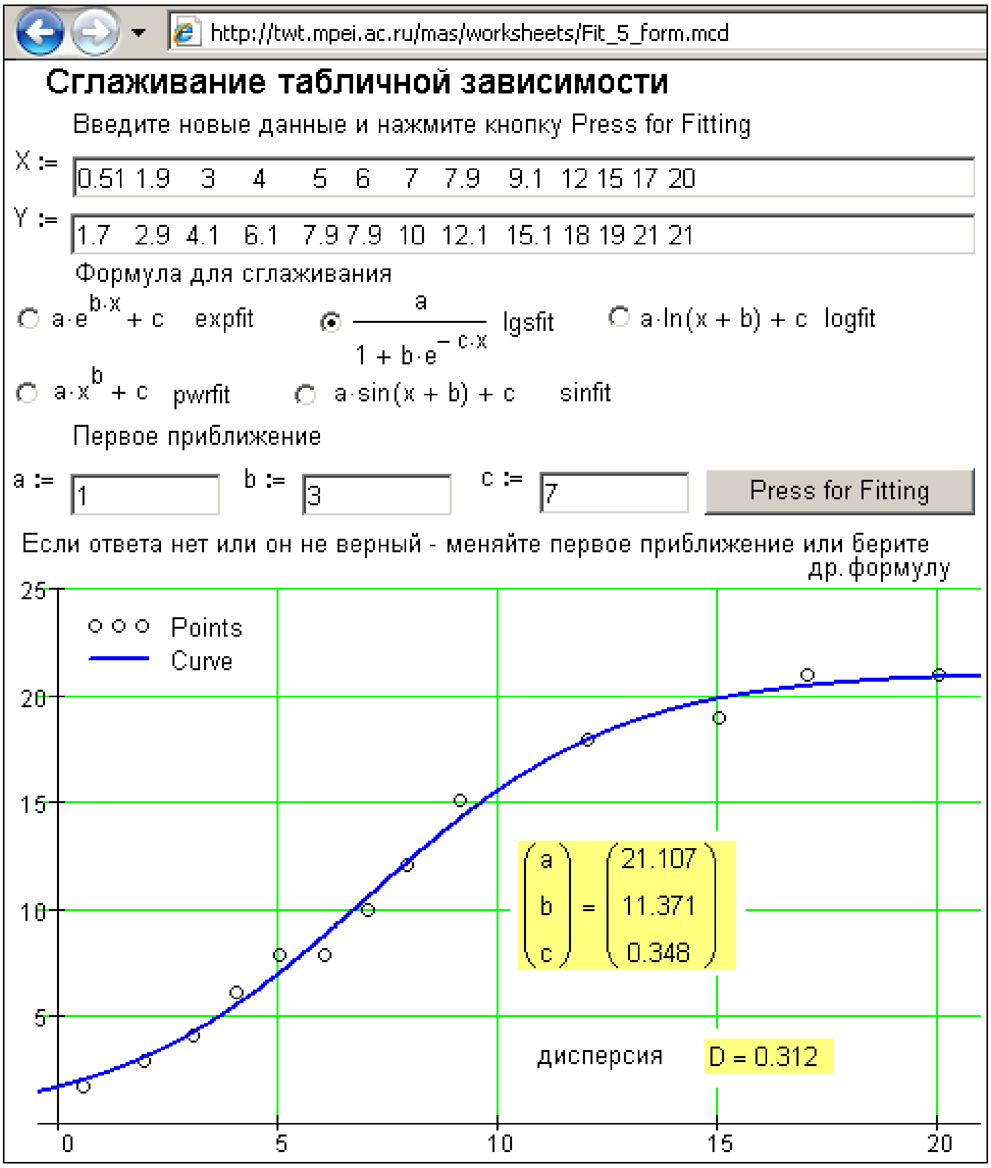


Рис. 11. Сглаживание табличной зависимости по пяти формулам в Интернете

На рис. 11 отображена интернет-страница в режиме on-line с аппроксимацией точек (они вводятся в два текстовых поля: **X** и **Y**) по пяти функциям разного вида, которые поддерживаются в среде Mathcad встроенными функциями с корнем `fit` (от англ. *fitting* — сглаживание: `expfit`, `lgsfit`, `logfit`, `pwrfit` и `sinfit`). Работа этих функций требует первого приближения (см. текстовые поля **a**, **b** и **c** на рис. 11). Требует первого приближения и универсальная функция `genfit` (general fitting), работающая с любой аппроксимирующей функцией. Кроме того, она требует ввода и частных производных сглаживающей функции по искомым коэффициентам. Другие подобные функции (`linfit`, `lnfit`, `medfit`, `line`, `loess`, `slope` и `intercept`) не требуют первого приближения.

На рис. 12 аппроксимирующая функция не выбирается из заданных, а вводится в отдельное текстовое поле.

Рис. 12. Сглаживание табличной зависимости по произвольной формуле в Интернете

Недостатки интерполяции (достоинства аппроксимации) таковы. Первый недостаток — промах по единственной точке (неверные исходные данные[9] или ошибка при их ручном вводе) сразу сильно искажает результат. Также сильно искажает результат преднамеренный или невольный выход за рамки интервала интерполяции (второй недостаток); при аппроксимации искажение будет не таким сильным, что дает основание считать аппроксимацию в какой-то мере и экстраполирующим решением.

Из интерполяции трудно вычленить вид искомой функции для ее использования, например, в другой программной среде, чего не скажешь про аппроксимацию. Покажем это на примере задачи трехмерной статистической обработки данных (см. на рис. 13—16). Решалась реальная инженерная задача по "оживлению" одной из характеристик работы паровой турбины.

На рис. 13 показан "бумажный" график некоего параметра паровой турбины, с которым приходилось работать по технологии "вождения пальцем по графику". Читатель может не вдаваться в саму технологию (пар, турбина, энтальпия), а просто вычленить для себя суть задачи — имеется застывшая на бумаге[10] функция двух аргументов (семейство кривых), которую необходимо "оживить" методами, описанными в данной главе так, чтобы можно было вводить значения двух аргументов, получать рассчитанное значение функции и заодно видеть этот "расчет" на графике.

На рис. 14 показано решение этой задачи.

---

[9] Автор когда-то сканированием таблицы из весьма солидного издания создавал базу данных по топливу для электростанций, открытую на MAS (**http://twt.mpei.ac.ru/MAS/Worksheets/Boiler/Th_C_B_Tab_XV.mcd**). Но не смотря на "солидность" издания, в книге оказалось очень много "числовых" опечаток — введено **3**531 вместо **5**531, например. Эти опечатки очень трудны для исправления корректором — как человеком, так и компьютером. Но интерполяция сразу выявила эти опечатки резким искривлением линий, соединяющей точки. Так что отмеченный недостаток интерполяции иногда может обернуться и достоинством.

[10] Графики несут много субъективной информации: через какие точки кто их провел!? Поэтому результаты обработки графиков будут включать, помимо методических, еще и субъективные ошибки.

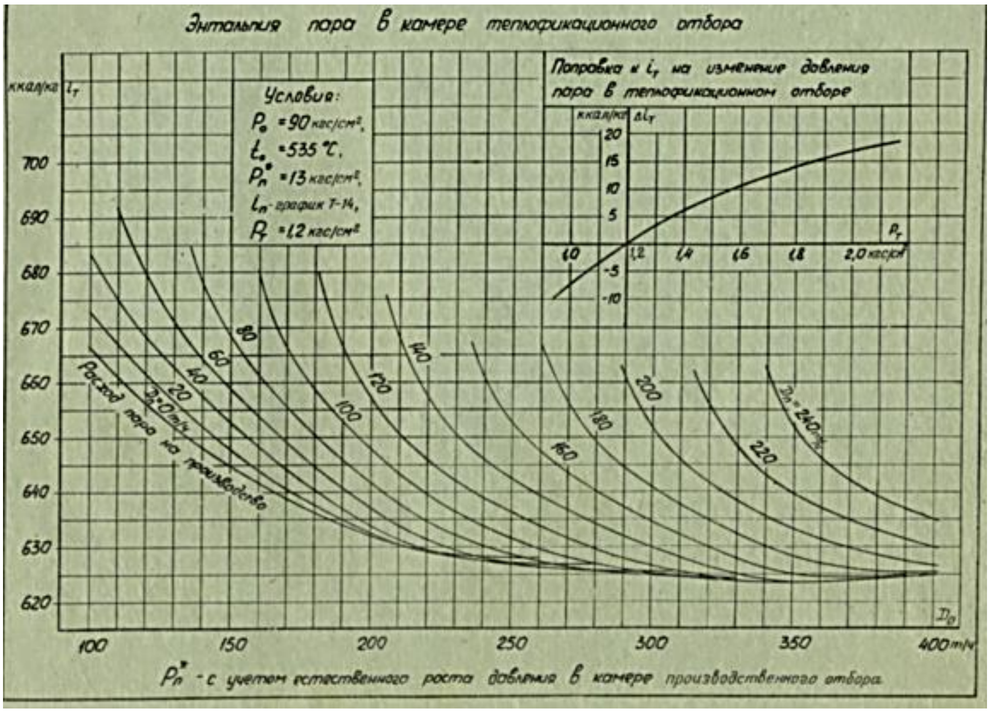


Рис. 13. "Сырой" график

Data :=

"X \ Y"	100	120	140	160	180	200	220	240	260	280	300	320	340	360	380	400
0	666	656	648	642	636	632	629									
20	672	661	652	644	637	632	629.4	627								
40	683	668	657	648	640	634	629	627								
60		680	664	654	645	637	631	627	627	625						
80			678	661	650	642	635	630	627	625	625					
100				680	659	648	639	634	630	626	625	624				
120						657	646	639	634	630	626	624	623	624		
140							657	646	639	634	630	626	624	624	624	625
160								660	647	640	635	630	626	625	625	625
180									667	649	641	635	630	627	625	625
200											652	641	630	631	628	626
220												650	643	637	633	630
240													662	646	639	635

Calculations X, Y and Z

$n := 2 \qquad Mxy := \text{augment}(X, Y) \qquad vs := \text{regress}(Mxy, Z, n) \qquad z_{interp}(x, y) := \text{interp}\left[vs, Mxy, Z, \begin{pmatrix} x \\ y \end{pmatrix}\right]$

Calculation a and I

$z(x, y) := \sum_{i=0}^{\text{last}(a)} \left(a_i \cdot x^{I_{i,0}} \cdot y^{I_{i,1}} \right) \qquad z(125, 240) = 642.452 \qquad z_{interp}(125, 240) = 642.452$

Аппроксимация и сглаживание кривой — regress

regress(vx, vy, n)

Возвращает вектор коэффициентов для многомерной аппроксимации многочленом n-й степени по методу наименьших квадратов данных в vx и vy. Этот вектор становится первым аргументом функции interp.

$I = \begin{pmatrix} 1 & 1 \\ 0 & 2 \\ 0 & 1 \\ 0 & 0 \\ 1 & 0 \\ 2 & 0 \end{pmatrix} \qquad a = \begin{pmatrix} -0.0040363 \\ 0.0022158 \\ -0.9810707 \\ 740.6516733 \\ 0.7296822 \\ 0.0025283 \end{pmatrix}$

$z_1(x, y) := a_0 \cdot x \cdot y + a_1 \cdot y^2 + a_2 \cdot y + a_3 + a_4 \cdot x + a_5 \cdot x^2$

$z_1(125, 240) = 642.452$

Рис. 14. Статистическая обработка "сырого" графика

Основная проблема в решении задачи сводится к заполнению матрицы `Data`, хранящей параметры точек на графике. Эти значения можно брать непосредственно из графика, используя обычную ("деревянную") или виртуальную линейку, если график отсканирован и его можно отобразить на дисплее компьютера. Тут лучше взять сами исходные данные, по которым строились точки, но они, как правило, очень часто бывают недоступными — график построили, а точки просто-напросто выкинули. На рис. 14 читатель видит, что некоторые элементы матрицы `Data` остались незаполненными, но это "обман зрения". В среде Mathcad такие (разреженные) матрицы недопустимы, и мы уже об этом говорили (см. рис. 1.53) — на этих пустых местах записаны нули, цвет шрифта которых был изменен с черного на белый *(см. разд. 1.2.3)*. Такая "фигурная" матрица (а у нас получилась "фигура" — стрелка, направленная вверх и влево) хорошо отображает точки на графике: их разное количество на разных линиях, а сами линии существуют в различных диапазонах аргумента по оси *x*.

На рис. 14 две области расчета скрыты. Первая область с именем "`Calculations X, Y and Z`" содержит операторы, генерирующие три вектора, прописанные в названии области — `x`, `y` и `z`. Мы как бы "распускаем свитер" связанной нами матрицы `Data` на три нитки одной длины[11], игнорируя "дырки на свитере" — нулевые (невидимые) элементы матрицы. Эта операция проводится несложным двойным циклом с включенным оператором `if` (см. рис. 1.53).

По полученным векторам `x`, `y` и `z` генерируется функция `z` двух аргументов, которая на рис. 15 и 16 отображается графически.

[11] А наши данные, показанные на рис. 5.14, и похожи на свитер: читатель, поверни голову чуть-чуть влево.

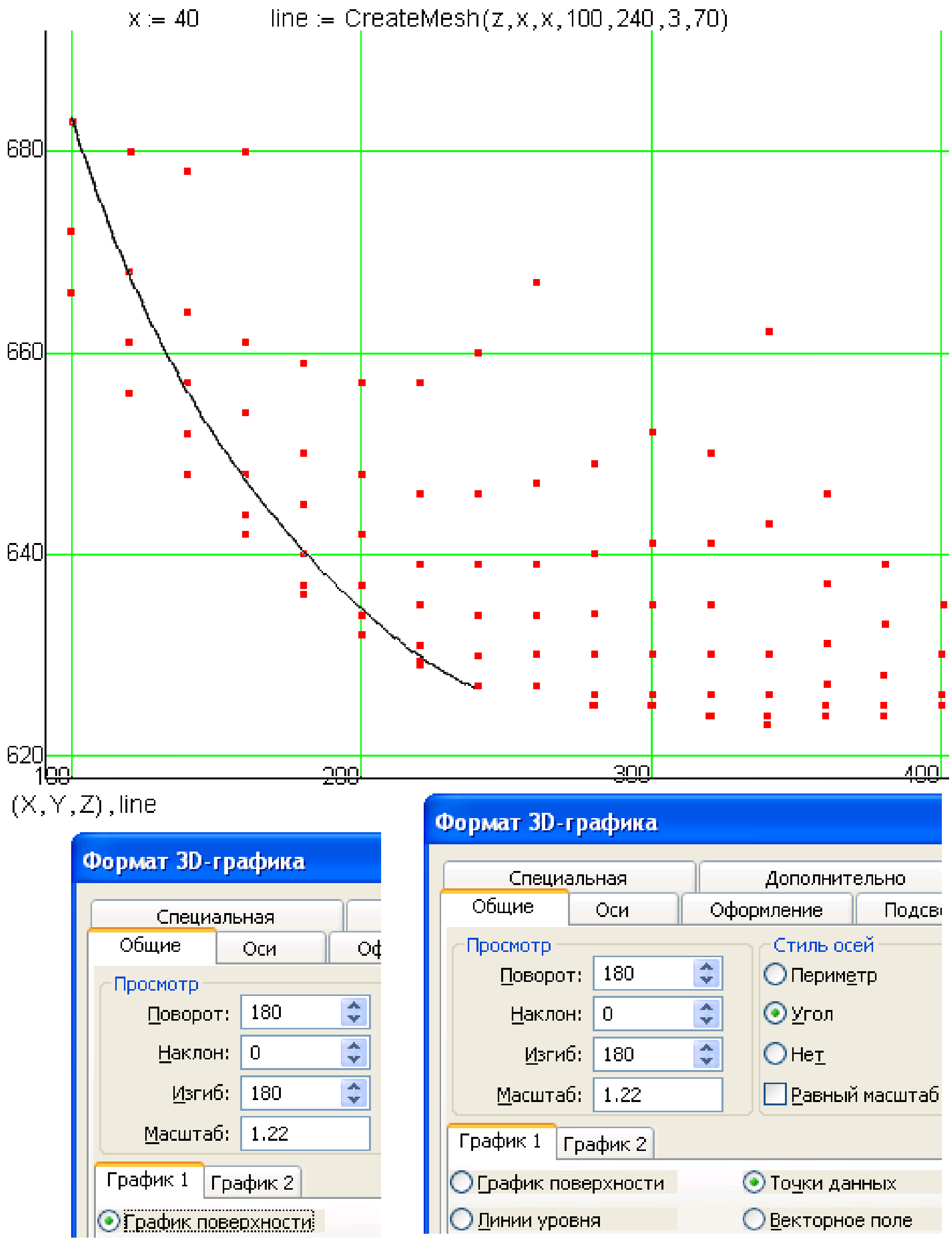


Рис. 15. Линия проходит через точки

На рис. 15 показаны все исходные "видимые" точки матрицы `Data`, показанной на рис. 14, и сглаживающая кривая при втором аргументе, равном 40. Показаны и окна форматирования 3D-графика. Так можно и нужно поверить все ряды точек по двум координатам на предмет возможной ошибки ввода.

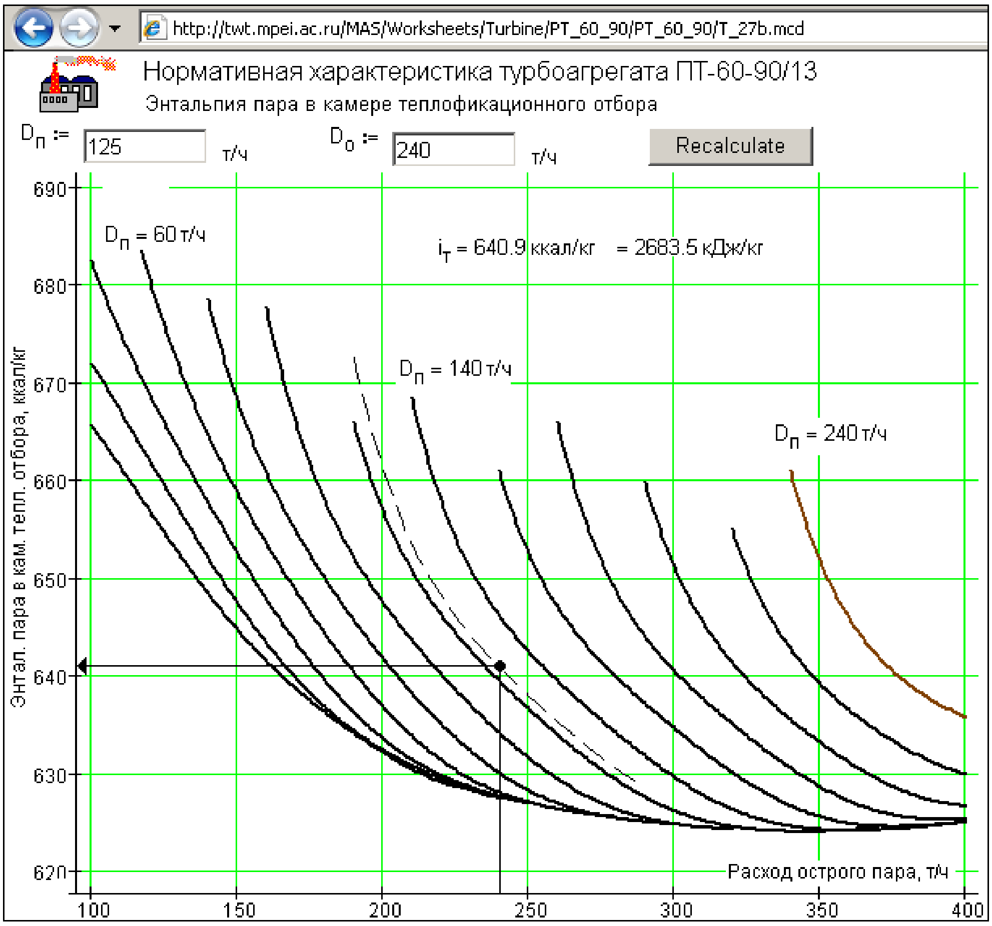


Рис. 16. Живой и "несырой" график

Рисунок 16 — это восстановление на MA/CS исходного бумажного графика, показанного на рис. 13. Теперь данный "живой" график можно открыть в Сети (**http://twt.mpei.ac.ru/MAS/Worksheets/Turbine/PT_60_90/PT_60_90/T_27b.mcd**), изменить в текстовых полях значения аргументов и получить новый числовой и "графический" ответ. Здесь, повторяю, достаточно дать только числовой ответ, но график, во-первых, показывает динамику процесса, во-вторых, позволяет контролировать введенные аргументы искомой функции без КПП (см. рис. 5) и, в-третьих, к графику уже привыкли, и без него "цифра" кажется уж очень сухой.

Но вернемся к рис. 14. Вторая скрытая область с именем "`Calculation a and I`" хранит довольно сложные операторы, рассчитывающие значение коэффициентов аппроксимирующего полинома (у нас он второй степени `n:=2`), которые выведены в нижней части рис. 14. Информация по этим операторам хранится в справочной системе Mathcad и отыскивается по ключевому слову *regress*.

На рис. 17 показан открытый в Интернете Mathcad-документ, по которому можно провести сплайн-интерполяцию по двум переменным, введя опорные точки в текстовые поля.

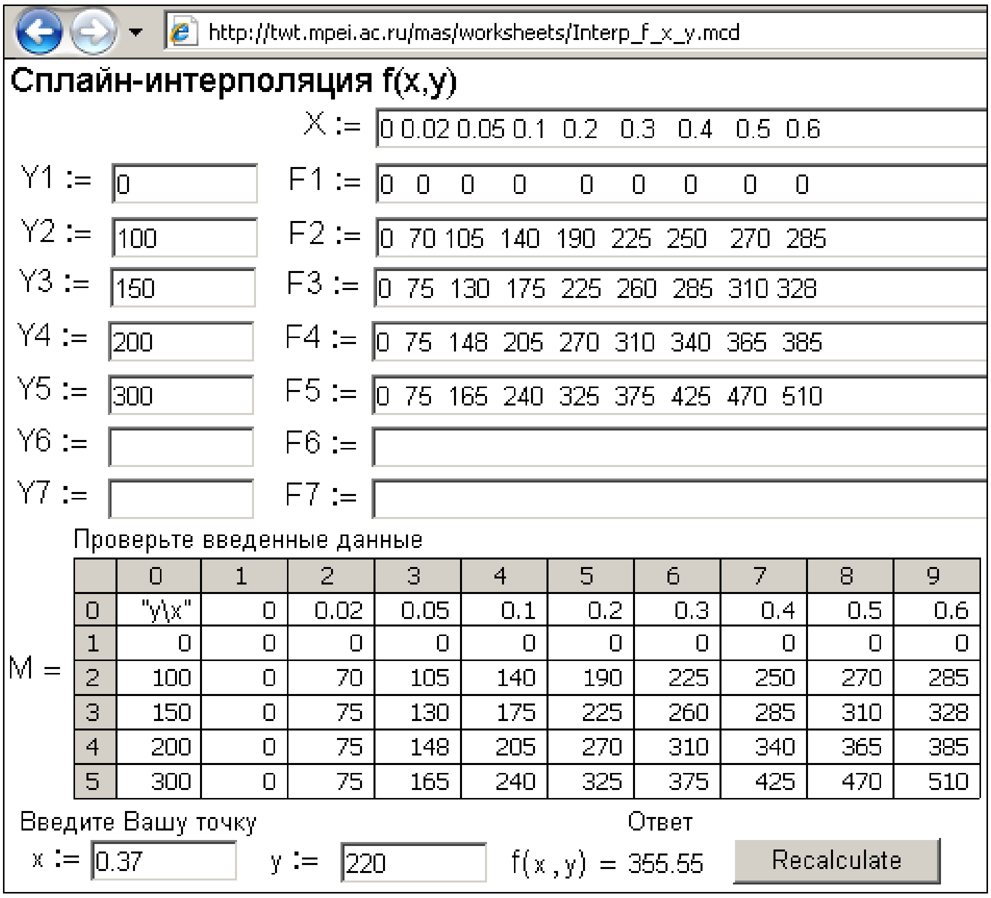


Рис. 17. Интернет-интерполяция по двум переменным

Но полином (см. рис 14) предпочтительнее сплайна (см. рис. 3) тем, что через полином при необходимости без труда можно перенести в другую программную среду.