Глава 8
Интерполяция и аппроксимация
или
Загадка рисунка ветки дерева

Сайт с анимацией и расчетом главы

https://community.ptc.com/t5/PTC-Mathcad/Fine-Spline/td-p/698455

Есть произведения изобразительного искусства, примечательные не только и не столько занимательностью сюжета или мастерством художника, а загадкой, которая в них спрятана. Эта тайна зачастую не бросается в глаза – о ней, как правило, рассказывают экскурсоводы, её растолковывают искусствоведы в описании картины в альбомах и каталогах выставок. Маленькая деталь на ином живописном полотне сведущим зрителям может показать скрытую первоначальную суть картины и существенно расширить панораму изображенного на картине сюжета…

Ещё большую интригу можно отметить в картинах, в какие заложены загадки с некоторым *математическим смыслом*. Тут стыкуются наука и искусство. Давайте рассмотрим одну такую «картину».

Рисунок 8.1 незамысловат и никак не претендует на какую-либо художественную ценность, но в него как раз и заложена одна загадка, о которой будет рассказано ниже.

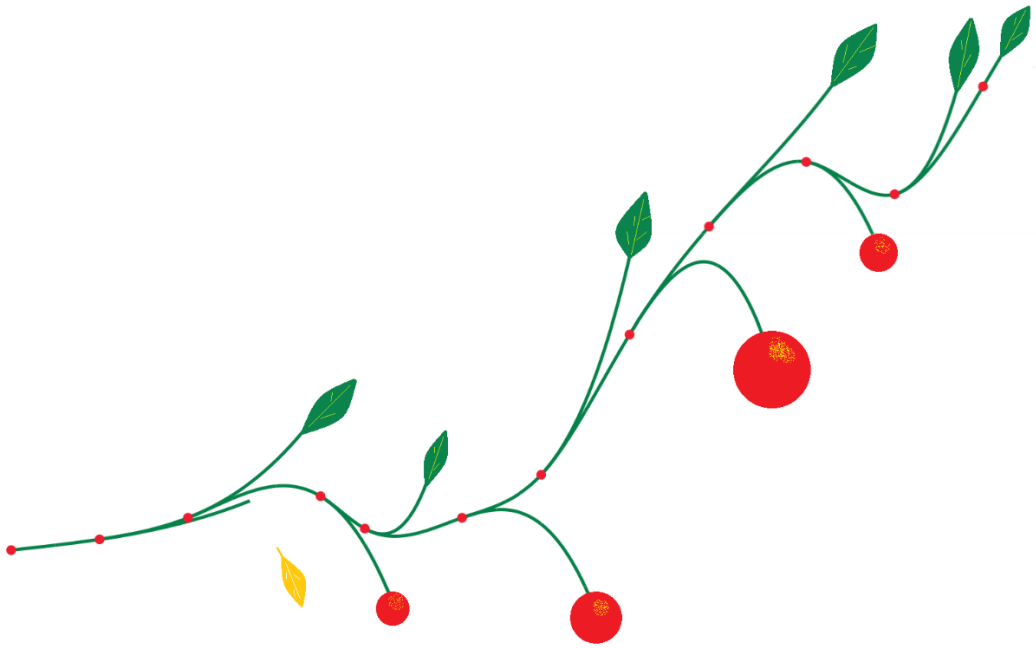


Рис. 8.1 Ветка дерева с листьями и плодами

Есть такая математическая задача. Даны дискретные значения некой функциональной зависимости. Необходимо восстановить эту функцию так, чтобы можно было рассчитать её значения в промежуточных точках внутри интервала аргументов (*интерполяция*) или даже вне его (*экстраполяция*) [1]. С такой задачей мы сталкиваемся тогда, когда, например, видим в справочнике табличные данные по плотности некоего материала в зависимости от температуры. В таблице есть данные для 10 и 20 градусов по шкале Цельсия, а нам нужно знать плотность при 15 градусах…

На рисунке 8.2 показано, как подобная задача может решаться с помощью пакета Mathcad. Дана матрица *Data* с двумя строками и 12 столбцами, хранящая дискретные значения некой функциональной зависимости. Из этой матрицы извлекаются вектор *X* (значения аргумента) и вектор *Y* (значения функции при данном аргументе). Встроенная в Mathcad функция с именем *linterp* (l – linear, interp – interpolation) генерирует функцию пользователя с именем *y*, по которой можно рассчитать промежуточные данные методом *кусочно-линейной интерполяции*. Полученная зависимость и отображена на графике – смежные точки соединены отрезками прямой линии.

$$Data := \begin{bmatrix} 1 & 2 & 3 & 4.5 & 5 & 6.1 & 7 & 8 & 8.9 & 10 & 11 & 12 \\ 4 & 5 & 7 & 9 & 6 & 7 & 11 & 24 & 34 & 40 & 37 & 47 \end{bmatrix}$$

$$X := \left(Data^{T}\right)^{\langle 1 \rangle} \qquad Y := \left(Data^{T}\right)^{\langle 2 \rangle}$$

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

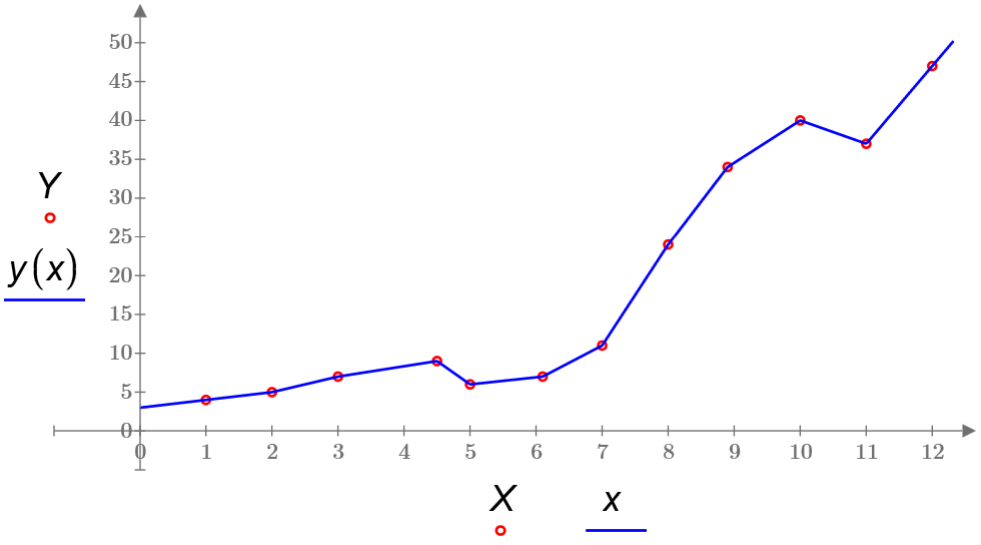


Рис. 8.2 Кусочно-линейная интерполяция

Но природа не терпит не только острых, но даже и тупых углов! У графика на рис. 8.1 три изъяна: природный, эстетический и математический. Во-первых, трудно представить себе какое-то свойство или какой-то процесс в живой природе с такими изломами в характеристиках. Во-вторых, линия на рис. 8.1 просто некрасива. А в-третьих, если мы захотим численными методами математики определить локальные минимумы и максимумы

данной функциональной зависимости, то нам понадобятся первая и вторая производные. А они представляют собой ступеньки (первая производная) или вообще равны нулю (вторая производная). Это затруднит работу с такой зависимостью.

В докомпьютерную эру углы на рис. 8.1 убирали с помощью лекал, прикладывая их разными краями к смежным точкам (любительское лекало – рис. 8.2). Были и более сложные профессиональные лекала с винтами, крутя которые можно менять кривизну стальной полоски (рис. 8.3).

Рис. 8.3 Пластмассовое и деревянные лекала

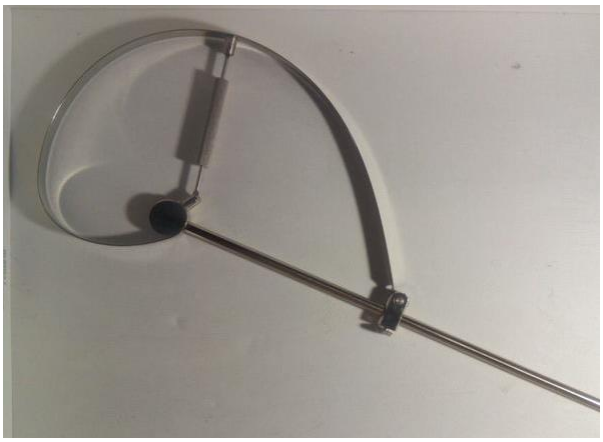


Рис. 8.4. Пружинное лекало

В арсенале Mathcad есть подобное «лекало» – пара встроенных функций **spline* и *interp* (рис. 8.5), позволяющих вести интерполяцию *сплайнами*[1].

Как ни странно, сплайны пришли в прикладную математику из кораблестроения, которое англичане в буквальном переводе называют морской архитектурой. Для изготовления деревянных деталей парусных судов размером до нескольких метров необходимо было сначала намечать их контуры, для чего использовалась стальная линейка, которая укладывалась между гвоздями, забиваемых в заготовку детали. После обводки карандашом деталь вырезалась. Можно показать, что кривая, образуемая линейкой между гвоздями как раз описывается кусочно-линейными полиномами третьей степени.

Вторично интерес к сплайнам возник только в начале 60х годов двадцатого века, когда из-за появления ЭВМ и машинной графики стало необходимым вычерчивать гладки кривые, а решения систем линейных алгебраических уравнений стало простой задачей.

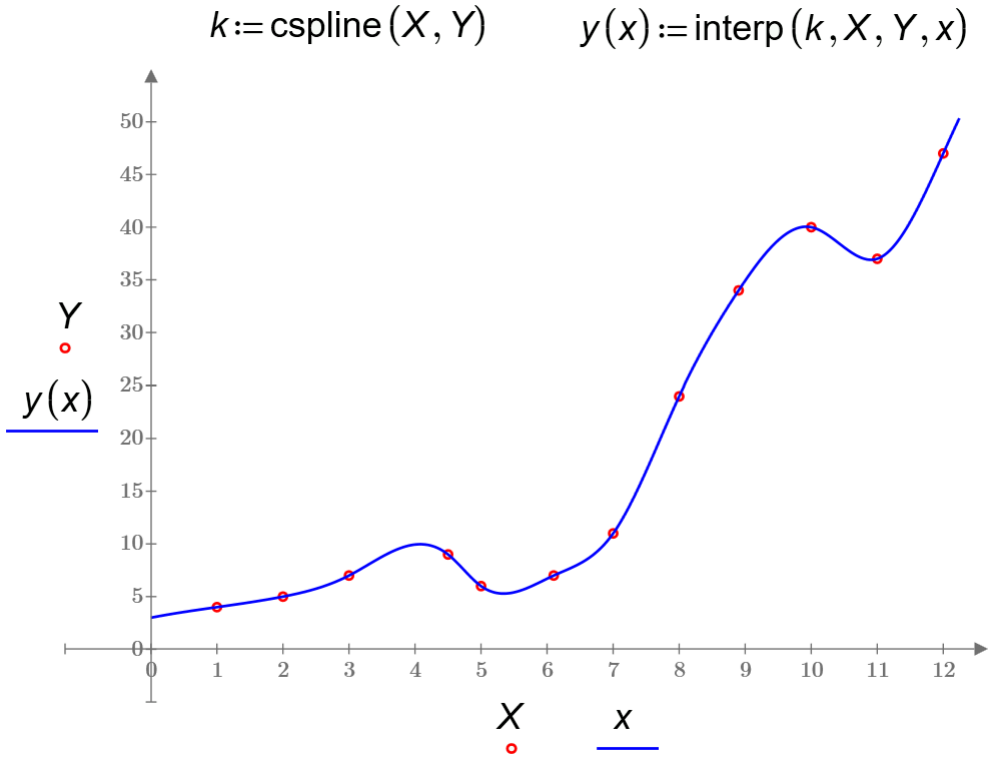


Рис. 8.5 Интерполяция кубическими сплайнами

В качестве префикса * функции **spline* можно использовать буквы *l* (linrar), *p* (parabolic) или *c* (cubic). На суть сплайна это не влияет, и он остается кубическим – смежные точки сединяются отрезками кривых полинома третьего порядка. Эти префиксы определяют порядок интерполяции и экстраполяции на краях заданного интервала точек. Как правило, пользователи Mathcad работают с функцией *cspline*, что сделали и мы – см. рис. 8.5.

[1] От англ. spline — гибкое лекало, гибкая плазовая рейка — полоса металла, используемая для черчения кривых линий.

А как ведется сплайн-интерполяция в среде Mathcad!? Ведь смежные точки необходимо сединить отрезками кривых полинома третьего порядка так, чтобы на стыках вся интерполирующая функция *y(x)* оставалоась гладкой.

Функция *spline возвращает вектор, в котором первые три элемента – это служебная информация для функции *interp*, а последующие элементы – это численные значения интерполирующей функции в узловых точках, что видно из рисунка 8.6.

$$k^{T} = [0 \;\; 3 \;\; 2 \;\; -1.272 \;\; 1 \;\; 3.272 \;\; -14.24 \;\; 16.104 \;\; -2.688 \;\; 15.834 \;\; -6.417 \;\; -3.091 \;\; -14.364 \;\; 13 \;\; 40.364]$$

$$y''(x) := \frac{d^2}{dx^2} y(x) \qquad y''(X_3) = 3.272 \qquad y''(X_6) = -2.688 \qquad y''(X_{12}) = 40.364$$

Рис. 8.6. Содержание вектора k, который генерируется функцией cspline

Зная численные значения кубического полинома и его вторых производных в двух смежных точках, несложно найти четыре его коэффициента. Для этого нужно решить четыре уравнения с четырьмя неизвестными – см. рис. 8.7.

$$cp(x, a, b, c, d) := a + b \cdot x + c \cdot x^2 + d \cdot x^3$$

$$cp''(x, c, d) := 6 \cdot d \cdot x + 2 \cdot c$$

Решить

Начальные приближения

$$a := 1 \qquad b := 1 \qquad c := 1 \qquad d := 1$$

Ограничения

$$cp(X_j, a, b, c, d) = Y_j \qquad cp(X_{j+1}, a, b, c, d) = Y_{j+1}$$

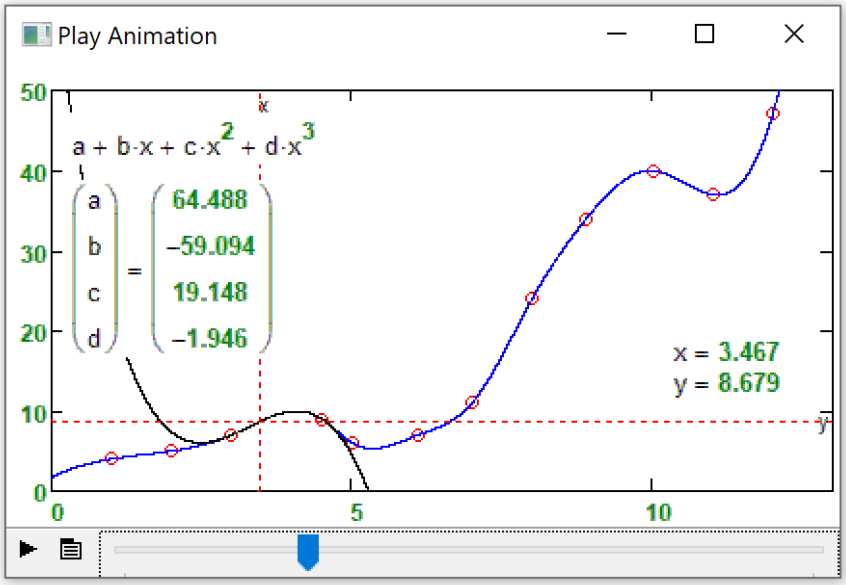
$$k_{j+3} = cp''(X_j, c, d) \qquad k_{j+1+3} = cp''(X_{j+1}, c, d)$$

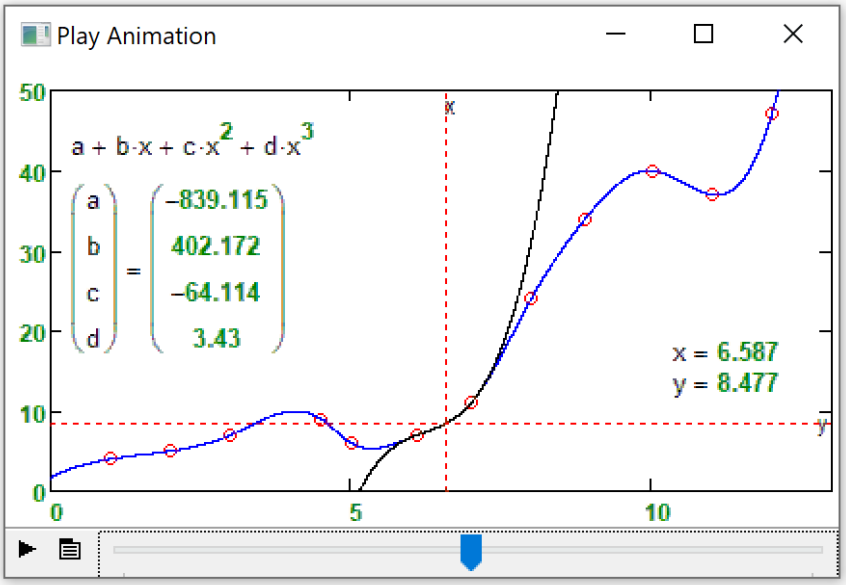
Решатель

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} := \text{Find}(a, b, c, d) = \begin{bmatrix} -5234.9968 \\ 1511.4385 \\ -143.9999 \\ 4.5606 \end{bmatrix}$$

Рис. 8.7 Нахождение коэффициентов сглаживающего кубического полинома

На рисунке 8.7 показано, как найдены коэффициенты кубического полинома для десятой (*j*) и одиннадцатой (*j*+1) точек. Меняя значения x, можно получить интересную анимацию, размещенную на сайте этой главы, три кадра которой показаны на рис. 8.8.


Play Animation
a + b·x + c·x^2 + d·x^3
a = 64.488
b = −59.094
c = 19.148
d = −1.946
x = 3.467
y = 8.679


Play Animation
a + b·x + c·x^2 + d·x^3
a = −839.115
b = 402.172
c = −64.114
d = 3.43
x = 6.587
y = 8.477

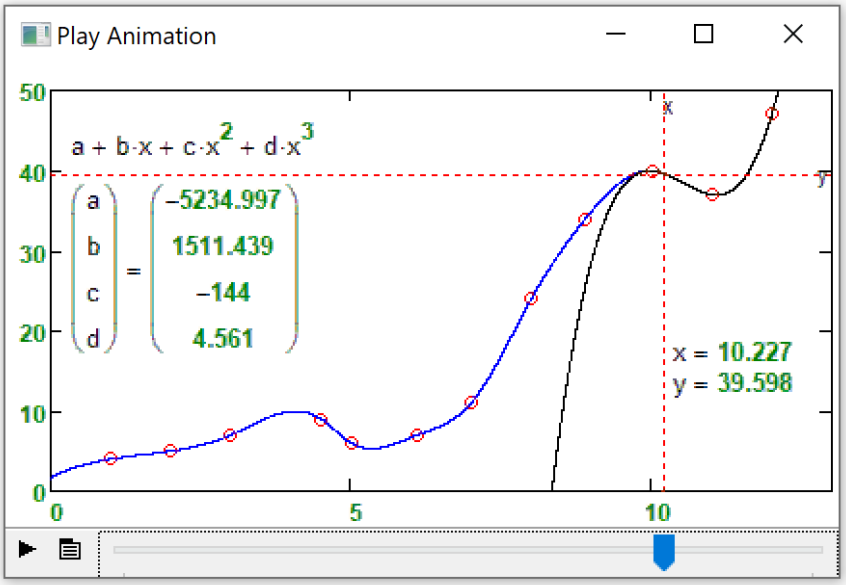

Play Animation
a + b·x + c·x^2 + d·x^3
a = −5234.997
b = 1511.439
c = −144
d = 4.561
x = 10.227
y = 39.598

Рис. 8.8 Три кадра анимации работы кубического сплайна

На рисунке 8.8 прорисованы и пронумерованы отрезки всех кубических парабол, формирующих интерполирующую функцию y(x), отображенную на рисунках 8.5 и 8.8.

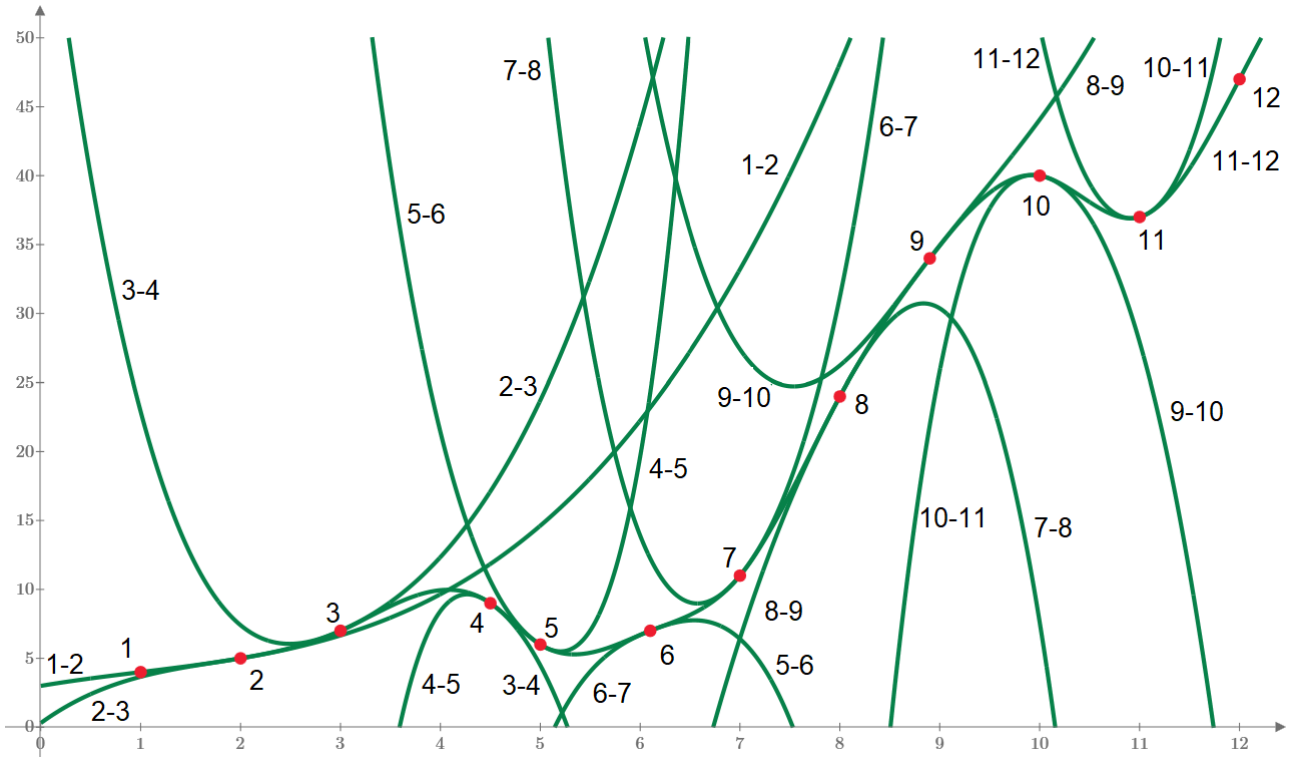


Рис. 8.8 Кубические параболы, сглаживающие табличную зависимость

Заросли кубических парабол на рис. 8.8 можно «подстричь» и получить то, что показано на рис. 8.9.

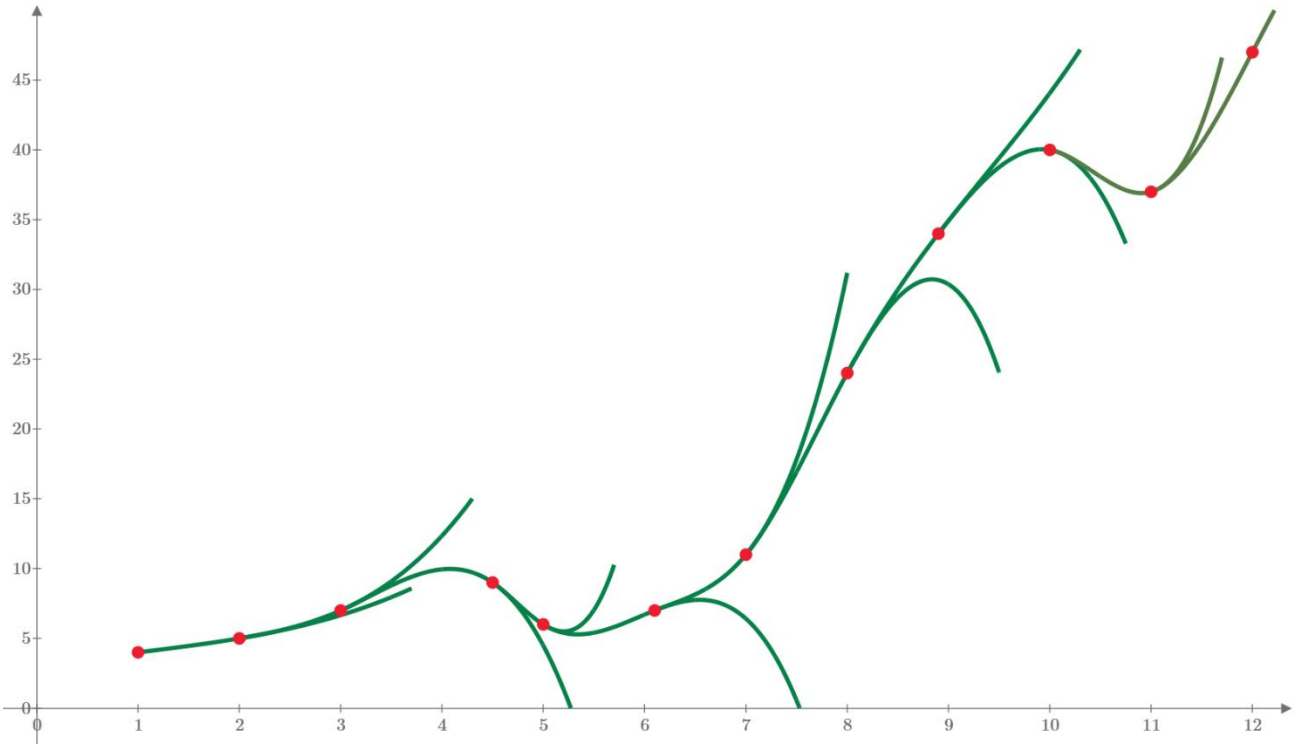


Рис. 8.9 Заготовка рисунка ветки дерева

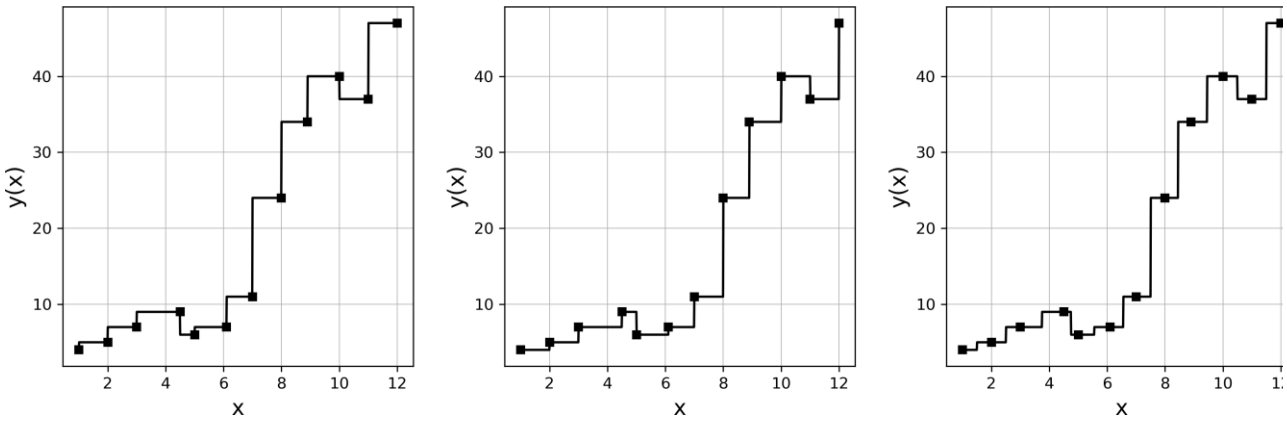
К ветви дерева, показанного на рис. 8.9, остается пририсовать листья и ягоды и получить «произведение изобразительного искусства с математической загадкой» – см. рис. 8.1.

Те же самые средства для интерполяции данных имеются в других системах для проведения научно-технических расчетов, в том числе и в экосистеме Python, где для этих целей используется библиотека scipy.interpolate. В частности, для проведения интерполяции достаточно создать интерполирующую функцию и предать ей массив значений по оси абсцисс, в которых необходимо получить значения интерполируемых данных [2]:

intrp = interp1d(x, y, kind=kind)

ynew = intrp(xnew)

здесь x, y – массивы данных по осям абсцисс и ординат, xnew – массив значений по оси абсцисс, для которого необходимо получить интерполируемые значения, a kind указывает, какой вид интерполяции необходимо использовать. Класс interp1d позволяет проводить сплайновую интерполяцию различных порядков. На рисунке 8.11 приводятся различные ломаные нулевого порядка, интерполирующие данные рис. 8.2.

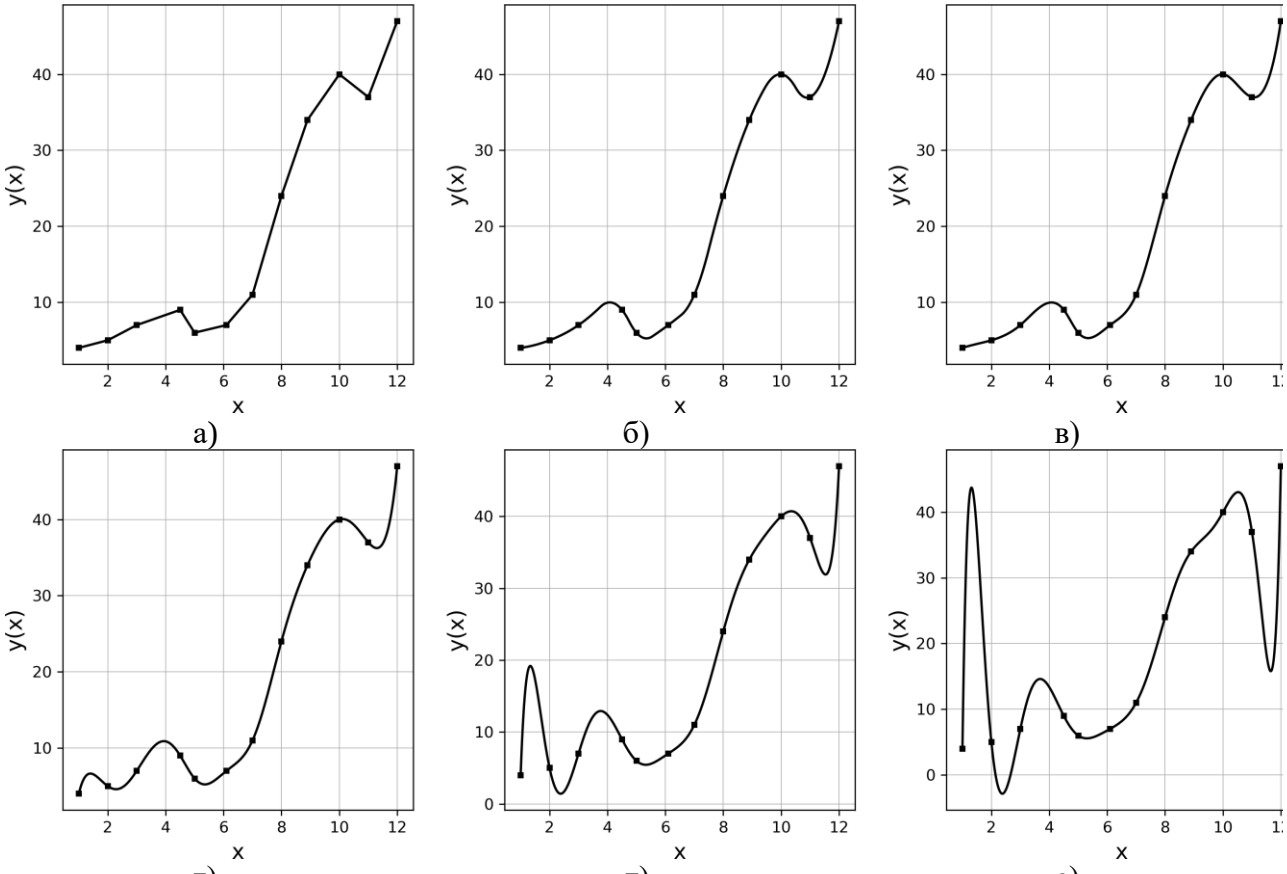


а) б) в)

Рис. 8.11 Интерполяция нулевого порядка

а) kind='next' – следующее значение, kind='previous' – предыдущее значение, kind='nearest' – ближайшее значение.

На рисунке 8.12 приводим результаты интерполяции сплайнами различных степеней.



а) б) в)

г) д) е)

Рис. 8.12 Сплайновая интерполяция различных степеней:

а) 1-я степень, б) 2-я степень; в) 3-я степень; г) 5-я степень; д) 7-я степень; е) 9-я степень

В большинстве практических случаев наилучшие результаты обеспечивает сплайновая интерполяция третьей степени, интерполяция более высоких порядков способствует образованию артефактов, но в любом случае при работе с данными рекомендуется провести

вычислительной эксперимент для выбора параметров интерполяции, тем более что для этого требуется минимум усилий.

До сих пор мы решали только задачу интерполяции, т.е. проводили кривые через заданный набор точек на плоскости. Можно снять требование на то, чтобы кривые точно проходили через заданные точки. Это позволит нам рисовать более гладкие кривые, чем изображенные на рис. 8.12 в, г. Такие задачи называются задачами аппроксимации, при их решении приходится искать компромисс между расстоянием от исходных данных до кривой и её гладкостью. В экосистеме Python для этого используется класс UnivariateSpline [3], создать экземпляр которого можно так:

spl = UnivariateSpline(x, y, k, s)

где x, y – набор исходных данных по осям координат, k – степень сплайна, k может принимать значения 1 –5, а s – допустимое расстояние между исходными данными и аппроксимирующим сплайном; при s=0 решается задача интерполяции. Чем больше s, тем более гладкой становится аппроксимирующая кривая.

Вычисление аппроксимированных значений осуществляется, как и ранее:

ynew = spl(xnew)

На рисунке 8.13 мы провели аппроксимацию исходной кривой для различных значений параметра s.

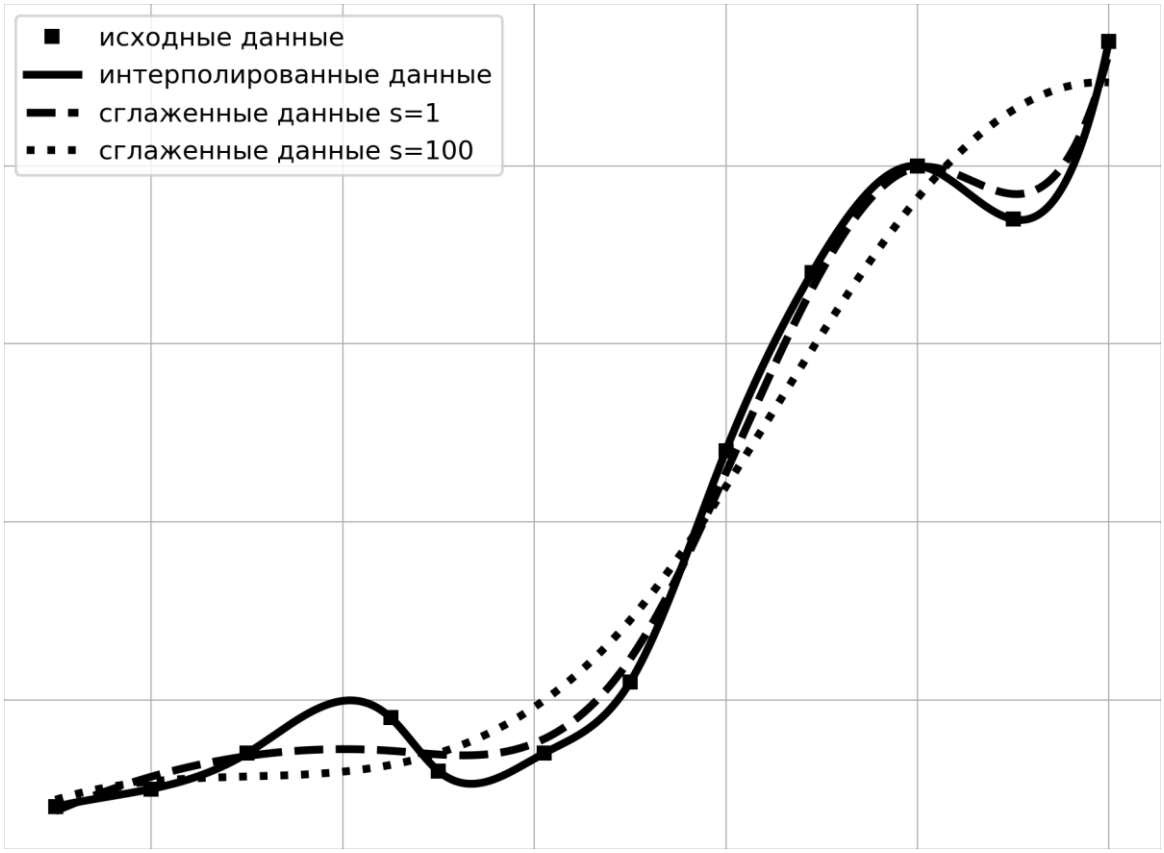


Рис. 8.13 Аппроксимация исходных данных при различных значения параметра s.

Для получения удовлетворительного результата аппроксимации значение параметра s необходимо подбирать.

Можно пойти немного дальше, имитируя зашумление исходных данных, для чего мы осуществили интерполяцию исходных данных в двухстах точках и прибавили к ним «шум» – нормально распределенные случайные числа с нулевым средним значением и средним квадратичным отклонением 1,5. Результаты аппроксимации зашумленных данных представлены на рис. 8.14.

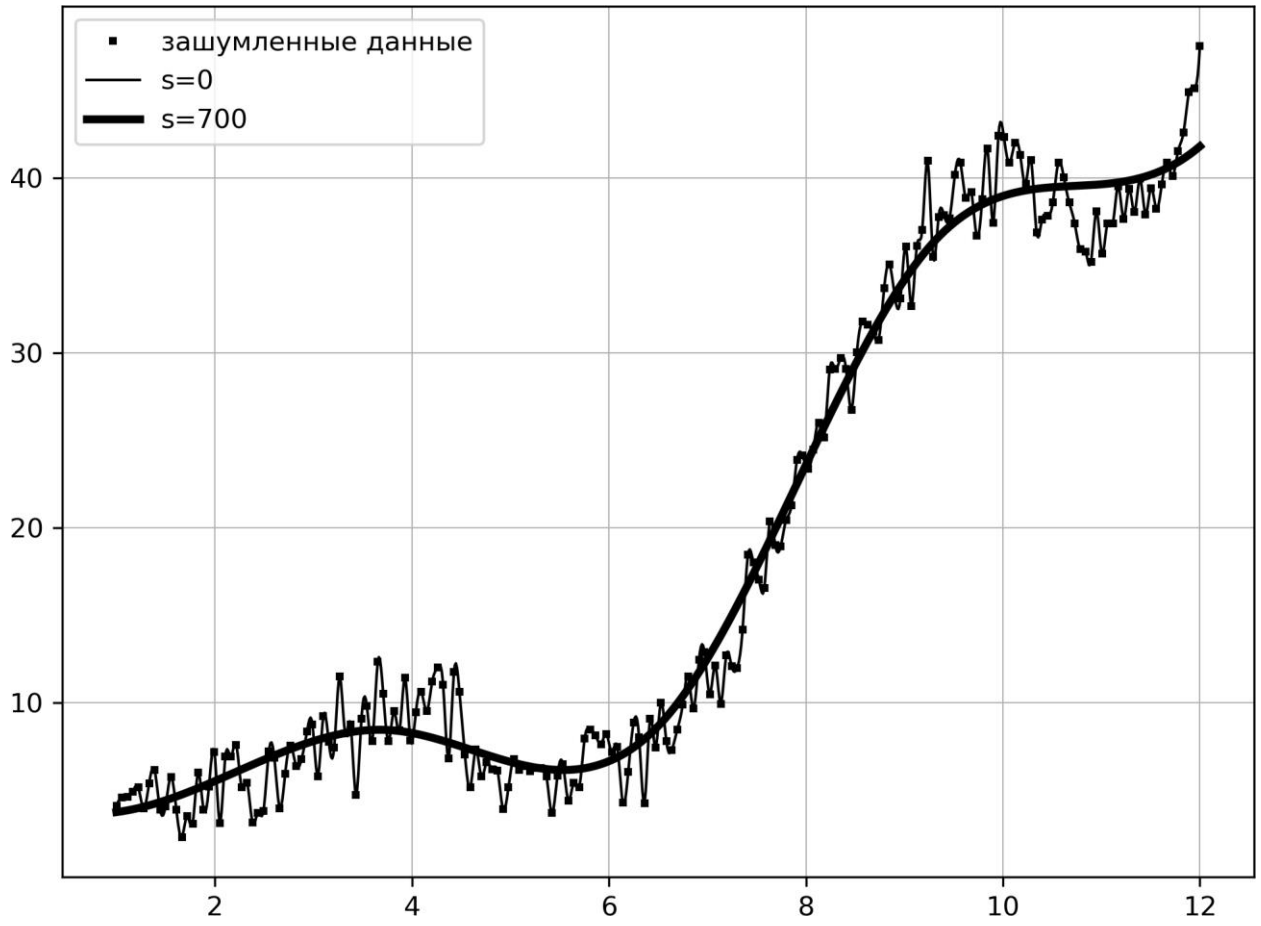


Рис. 8.14 Аппроксимация зашумленных данных

Меняя s, можно достичь компромисса между гладкостью кривой и расстоянием от исходных данных.

Решение задач интерполяции и аппроксимации позволяет получать значения на отрезке по оси абсцисс, где данные заданы. Для предсказания значений вне этого отрезка приходится решать задачу экстраполяции. Решение таких задач требует дополнительной информации о поведении экстраполируемой функции. В простейшем случае можно считать, что экстраполируемая функция не меняется и ее значения совпадают со значением на границе отрезка, на котором заданы данные. Другой гипотезой может быть предположение, что экстраполируемая функция ведет себя линейно. В любом случае гипотезу о поведении функции необходимо применять с широко открытыми глазами, иначе могут быть получены совершенно неожиданные результаты. На рис. 8.15 приводятся вычислительного

эксперимента по экстраполяции исходных данных рис. 8.2 с помощью UnivariateSpline и сплайнов различной степени.

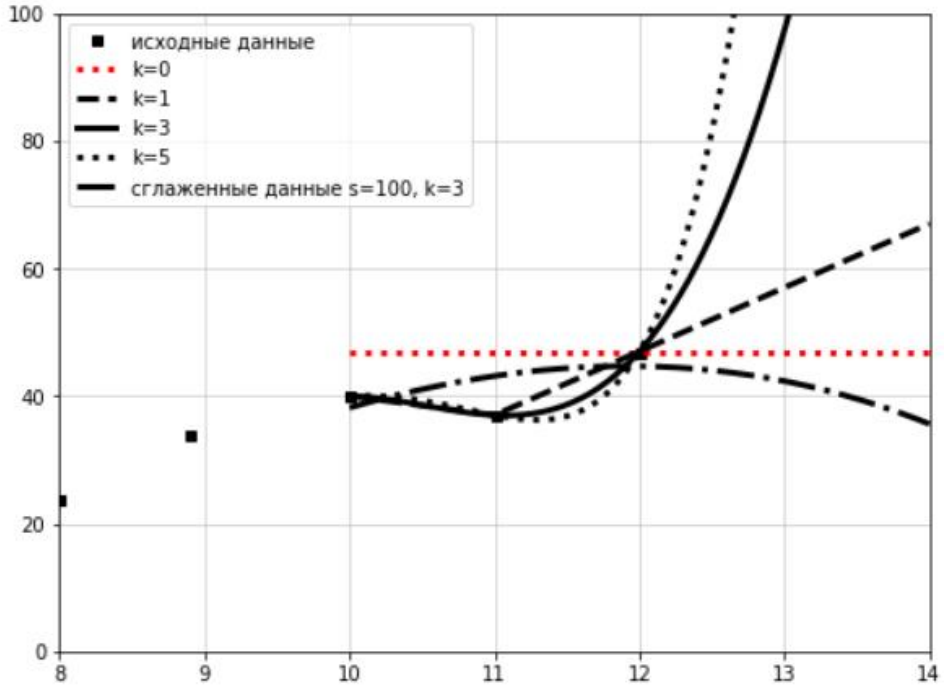


Рис. 8.15 Экстраполяция с помощью сплайнов различных степеней

На рисунке 8.15 показано, как ведет себя экстраполируемая функция при различных предположениях относительно её поведения, включая сплайновую интерполяцию от нулевой до пятой степени и сглаживание сплайном третьей степени. Как мы уже говорили раньше, выбор способа аппроксимации зависит от имеющейся в распоряжении информации и от человека, принимающего решение.

В пакете Mathcad есть средства интерполяции сплайном функций не только одного, но и двух аргументов. Для этого необходимо и достаточно в функциях **spline* и *interp* (рис. 8.5) вектор X заменить на матрицу с двумя столбцами, хранящую дискретные значения двух аргументов, а вектор Y – на квадратную матрицу дискретных значений функции для значений аргументов, хранящихся в векторе X. Таблица исходных дискретных данных должна быть квадратной, что не очень удобно. В связи с этим авторами была создана функция двойной интерполяции для исходных данных любой конфигурации – см. рис. 8.16.

$$z(x,y) := \left\| \begin{array}{l} M \leftarrow \begin{bmatrix} \text{“x\y”} & 22 & 23 & 24 & 25 & 26 & 27 & 28 & 29 & 30 & 31 & 32 & 33 & 34 & 35 & 36 \\ 23 & 107 & 98 & 90 & 82 & 76 & 71 & 66 & 61.5 & 58 & 54 & 51 & 48 & 45 & 42 & 40 \\ 65 & 230 & 212 & 194 & 179 & 165 & 153 & 142 & 132 & 124 & 116 & 109 & 102 & 97 & 91 & 87 \end{bmatrix} \\ X \leftarrow \text{submatrix}(M, 2, \text{rows}(M), 1, 1) \\ Y \leftarrow (\text{submatrix}(M, 1, 1, 2, \text{cols}(M)))^{\text{T}} \\ Z \leftarrow \text{submatrix}(M, 2, \text{rows}(M), 2, \text{cols}(M)) \\ \text{for } i \in 1 .. \text{cols}(Z) \\ \quad \left\| Zv_i \leftarrow \text{linterp}(X, Z^{(i)}, x) \right. \\ \text{interp}(\text{cspline}(Y, Zv), Y, Zv, y) \end{array} \right.$$

$$z(40,22) = 156.786 \qquad z(50,27.5) = 119.237$$

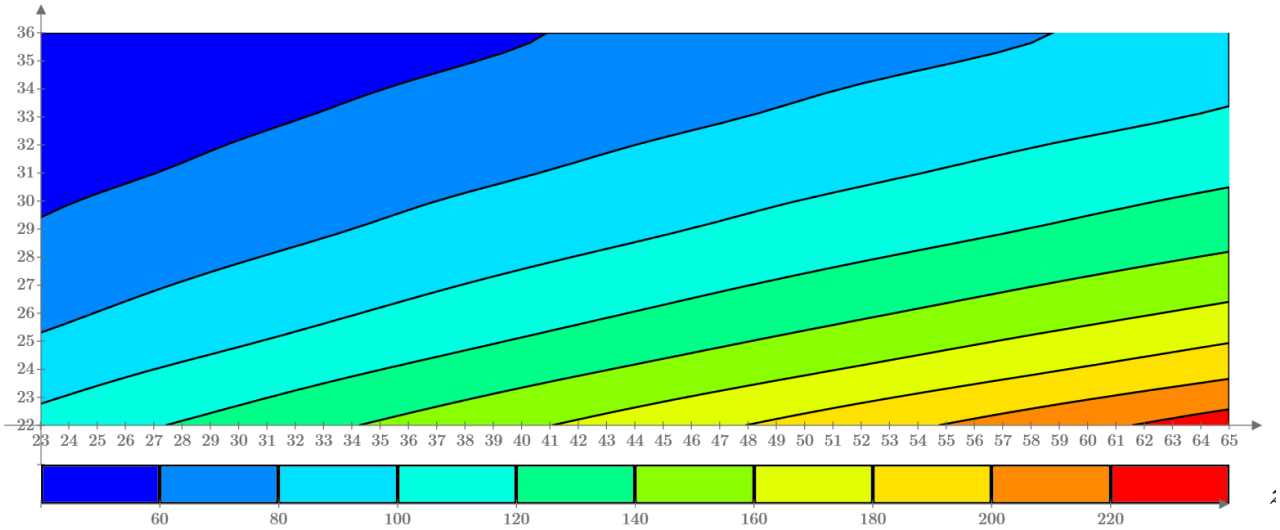


Рис. 8.16 Трехмерная комбинированная интерполяция

Функция *z(x, y)* удобна и тем, что в ней можно сочетать разные методы интерполяции. На рисунке 8.11 показана интерполяция данных с двумя строками по аргументу *x* и с пятнадцатью столбцами по аргументу *y*. Интерполяция сплайнами по двум точкам невозможна, поэтому там применяется линейная интерполяция. Если же в таблице *M* увеличить число строк, то можно перейти к интерполяции сплайнами, которая применена для аргумента *y*.

В [4] авторская методика интерполяции зависимостей и для случаев не полностью заполненных матриц, а также интерполяция вида *f(x, y, z)*.

Экосистема Python и в этом отношении не отстает от Mathcad. Если зависимость, изображенную на рис. 8.5, обозначить как f(x), то функцию двух переменных $g(x,y) = \sqrt{f(x) \cdot f(y)}$ можно представить как двумерную поверхность в трехмерном пространстве (рис. 8.17). Для рисования значения координат задавались в 10000 точек на регулярной сетке 100 на 100.

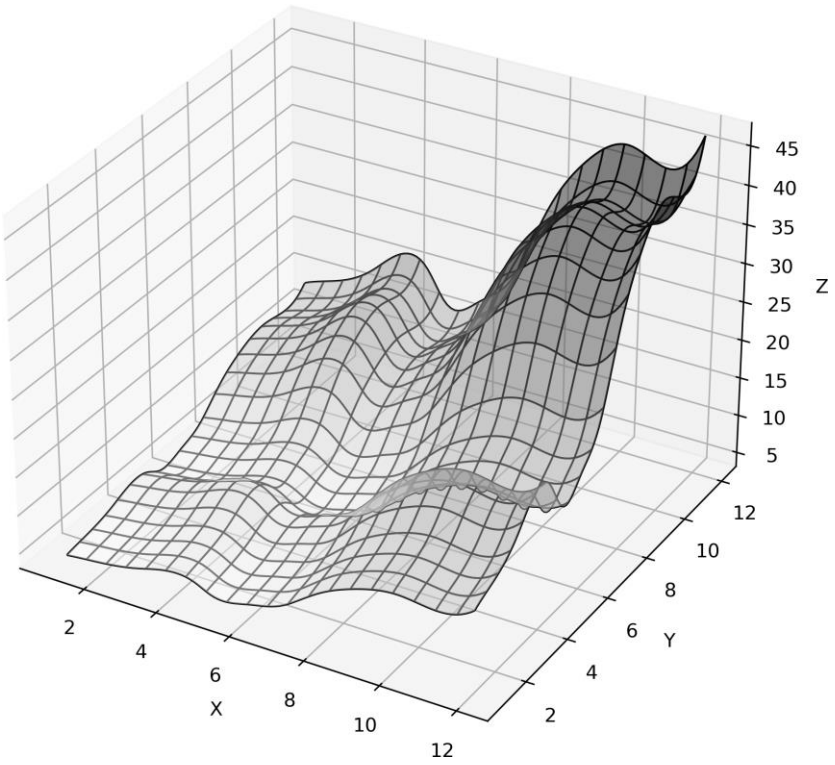
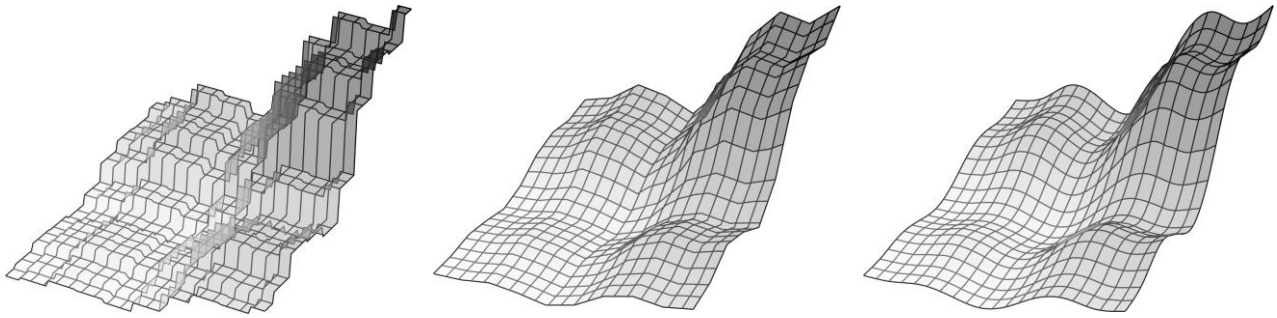


Рис. 8.17 График функция $g(x, y)$

Сплайновая интерполяция поверхности может быть выполнена с помощью griddata [5]. При этом сплайновую интерполяцию можно осуществить как на регулярной сетке, так и на произвольном наборе точек (x, y, z). Начнем мы с регулярной сетки. На рисунке 8.18 представлена сплайновая интерполяция на сетке 10 на 10.



а) б) б)

Рис. 8.18 Интерполяция сплайнами функции $g(x, y)$ на регулярной сетке 10 на 10:
а) интерполяция нулевого порядка; б) кусочно-линейная интерполяция;
в) кубическая сплайн-интерполяция

Очевидно, что восстановление формы поверхности на регулярной сетке даже с небольшим числом разбиений осуществляется вполне удовлетворительно.

Все существенно меняется, если интерполяция осуществляется на случайном наборе точек, как это показано на рис 19.

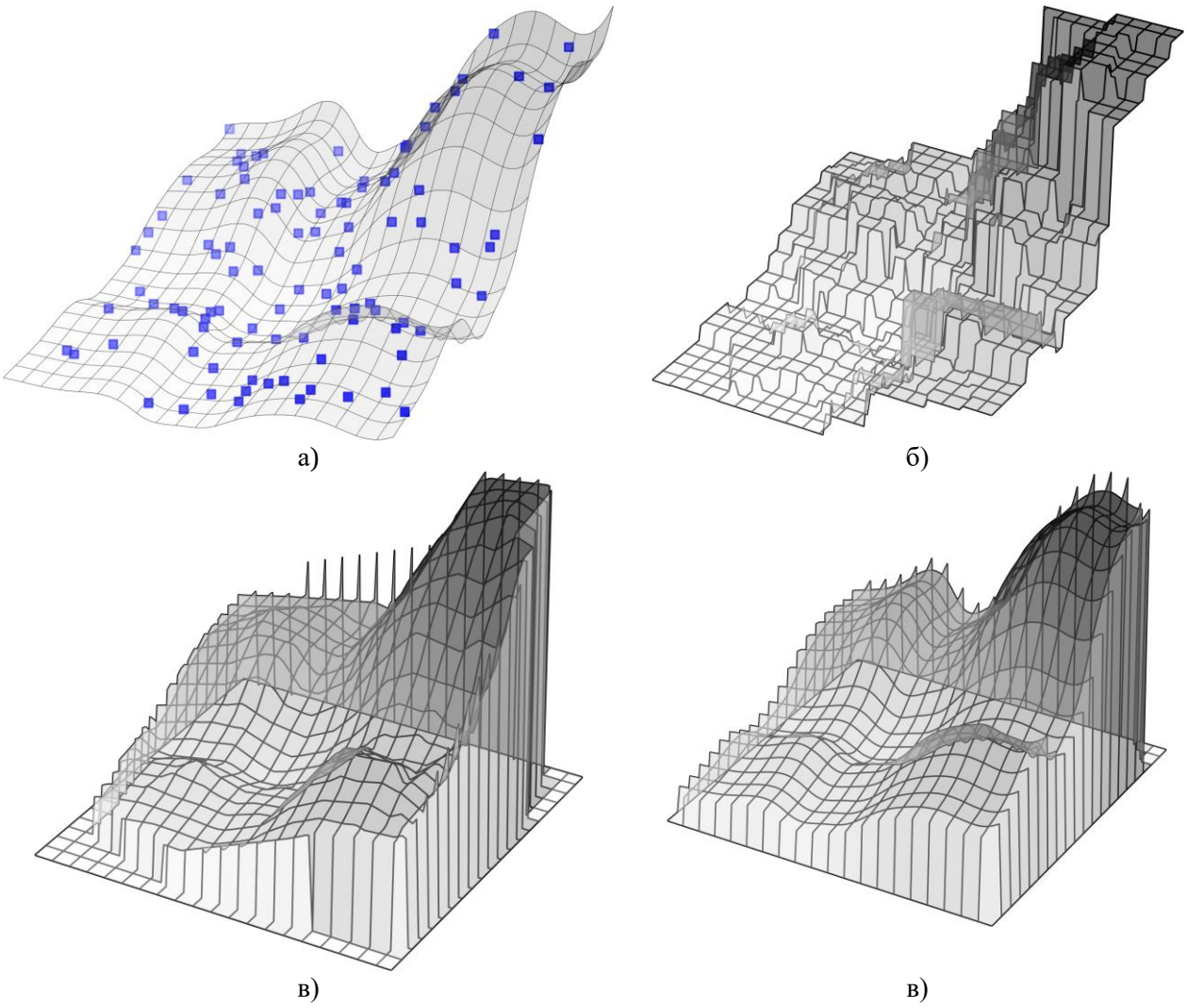


Рис. 19 Интерполяция $g(x, y)$ на случайном наборе из ста точек:
а) случайный набор точек, на котором осуществлялась интерполяция; а) интерполяция нулевого порядка; б) кусочно-линейная интерполяция; в) кубическая сплайн-интерполяция

Для случайного набора точек восстановить поверхность удалось не на всей области, в тех точках, где это не удалось сделать, значения функции приравнивались нулю. Кроме того, интерполяция на нерегулярной сетке приводит к появлению артефактов.

Литература

1. Половко А.М., Бутусов П.Н. Интерполяция. Методы и компьютерные технологии их реализации. СПб.: БХВ-Петербург, 2002, – 320 с.: с илл. URL: https://www.studmed.ru/view/polovko-am-butusov-pn-interpolyaciya-metody-i-kompyuternye-tehnologii-ih-realizacii_e938ef58445.html (дата обращения: 21.11.2020)
2. scipy.interpolate.interp1d [Электронный ресурс]. URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.interp1d.html (дата обращения 24.11.2020)

3. scipy.interpolate.UnivariateSpline. [Электронный ресурс]. URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.UnivariateSpline.html (дата обращения 24.11.2020)
4. Теплотехнические расчеты на компьютере / Александров А. А., Аунг Ту Ра Тун, Гаряев А. Б. [и др.] – М.: Издательство МЭИ, 2019. - 447 с. : цв. ил. ISBN 978-5-7046-2211-6. URL: http://twt.mpei.ac.ru/ochkov/Therm-Studies.pdf (дата обращения: 21.11.2020)
5. scipy.interpolate.griddata [Электронный ресурс]. URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.griddata.html (дата обращения 24.11.2020)