

## *Лекция № 4*

Методы аппроксимации, интер- и экстрополяции в математическом пакете Mathcad

## Аппроксимация

Замена одних математических объектов другими, в том или ином смысле близкими к исходным (в частности, приближенное выражение сложной *функции* с помощью более простых).

## Интерполяция

Нахождение промежуточного значения какой-либо изменяющейся величины по ряду соседних известных значений

## Экстраполяция

Вычисление по нескольким членам какого-либо статистического ряда таких его значений, которые находятся вне пределов данного ряда.

Аппроксимация функций заключается в приближенной замене заданной функции  $f(x)$  некоторой функцией  $\varphi(x)$  так, чтобы отклонение функции  $\varphi(x)$  от  $f(x)$  в заданной области было наименьшим. Функция  $\varphi(x)$  при этом называется *аппроксимирующей*.

Типичной задачей аппроксимации функций является задача *интерполяции*.

### Необходимость интерполяции функций

Функция  $f(x)$  имеет сложное аналитическое описание, вызывающее определенные трудности при его использовании (например,  $f(x)$  является спецфункцией: гамма-функцией, эллиптической функцией и др.).

Аналитическое описание функции  $f(x)$  неизвестно, т. е.  $f(x)$  задана таблично. При этом необходимо иметь аналитическое описание, приближенно представляющее  $f(x)$  (например, для вычисления значений  $f(x)$  в произвольных точках, определения интегралов и производных от  $f(x)$ )

## Простейшая задача интерполяции

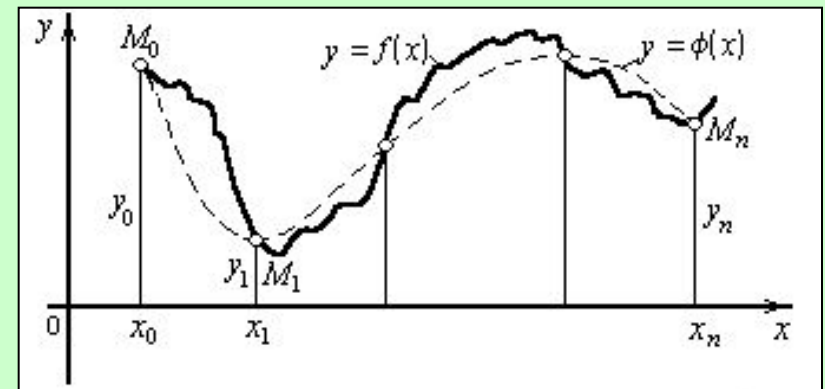
На отрезке  $[a, b]$  заданы  $n + 1$  точки  $x_i = x_0, x_1, \dots, x_n$ , которые называются *узлами интерполяции*, и значения некоторой функции  $f(x)$  в этих точках

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

Требуется построить функцию  $F(x)$  (*интерполяционная функция*), принадлежащую известному классу и принимающую в узлах интерполяции те же значения, что и  $f(x)$ , т. е. такую, что

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

Геометрически это означает, что нужно найти кривую  $y = F(x)$  некоторого определенного типа, проходящую через заданную систему точек  $M(x_i, y_i)$  ( $i = 0, 1, \dots, n$ )



В такой общей постановке задача может иметь бесконечное множество решений или совсем не иметь решений.

Однако эта задача становится однозначной, если вместо произвольной функции  $F(x)$  искать полином  $\varphi(x)$  (*интерполяционный полином*) степени не выше  $n$ , удовлетворяющий условиям (2), т. е. такой, что

$$\varphi(x_0) = y_0, \varphi(x_1) = y_1, \dots, \varphi(x_n) = y_n.$$

Полученную интерполяционную формулу

$$\varphi(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

Обычно используют для приближенного вычисления значений данной функции для значений аргумента  $x$ , отличных от узлов интерполяции. Такая операция называется *интерполяцией функций*.

# Виды интерполяции

## глобальная

соединение всех точек единым интерполяционным полиномом

Параболическая интерполяция  
Интерполяционная формула Лагранжа  
Интерполяционные формулы Ньютона

## локальная

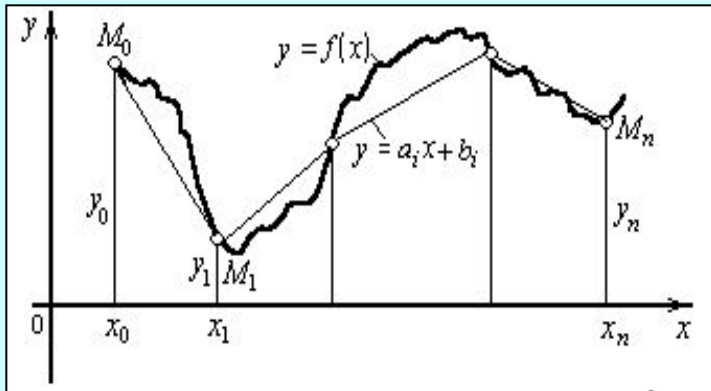
соединение точек отрезками прямой (по двум точкам), отрезками параболы (по трем точкам)

Линейная интерполяция  
Квадратичная интерполяция  
Кубическая сплайн-интерполяция



# Линейная интерполяция

Является простейшим и часто используемым видом локальной интерполяции. Заданные точки  $M(x_i, y_i)$  ( $i = 0, 1, \dots, n$ ) соединяются прямолинейными отрезками, и функция  $f(x)$  приближается к ломаной с вершинами в данных точках



Уравнения каждого отрезка ломаной линии в общем случае разные. Поскольку имеется  $n$  интервалов  $(x_i, x_{i+1})$ , то для каждого из них в качестве уравнения интерполяционного полинома используется уравнение прямой, проходящей через две точки.

В частности, для  $i$ -го интервала можно написать уравнение прямой, проходящей через точки  $(x_i, y_i)$  и  $(x_{i+1}, y_{i+1})$ , в виде:

$$\frac{y - y_i}{y_{i+1} - y_i} = \frac{x - x_i}{x_{i+1} - x_i}.$$

Отсюда

$$y = a_i x + b_i, \quad x_i \leq x \leq x_{i+1},$$
$$a_i = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}, \quad b_i = y_i - a_i x_i.$$

Следовательно, при использовании линейной интерполяции сначала нужно определить интервал, в который попадает значение аргумента  $x$ , а затем подставить его в формулу и найти приближенное значение функций в этой точке.



## Линейная интерполяция в MathCAD

При линейной интерполяции MathCAD соединяет существующие точки данных прямыми линиями (соединяет точки данных отрезками прямых, создавая таким образом ломаную). Это выполняется функцией *linterp*.

### *linterp* (vx, vy, x)

Использует векторы данных vx и vy, чтобы возвратить интерполируемое значение u, соответствующее третьему аргументу x. Аргументы vx и vy должны быть векторами одинаковой длины. Вектор vx должен содержать вещественные значения, расположенные в порядке возрастания.

Для значений x, расположенных перед первой точкой в векторе vx, MathCAD продолжает ломаную прямой линией, проходящей через первые две точки данных. Для значений x, расположенных за последней точкой vx, MathCAD продолжает ломаную прямой линией, проходящей через последние две точки данных.

Для получения наилучших результатов x должно находиться между самыми большими и самыми маленькими значениями vx. Маловероятно, что будут полезны значения, вычисленные для x вне этого диапазона.

Функция *linterp* предназначена для интерполяции, а не для экстраполяции.

# Линейная интерполяция в MathCAD

**Mathcad Professional - [Линейная интерполяция.MCD]**

Файл Правка Вид Вставка Формат Математика Символы Окно ?

Линейная интерполяция

$f(x) := x^2 \cdot e^{-x^2}$  - Исходная функция. Зададим функцию  $f(x)$  таблично на отрезке  $[0, 2.5]$  в 6 узловых точках. Для этого:

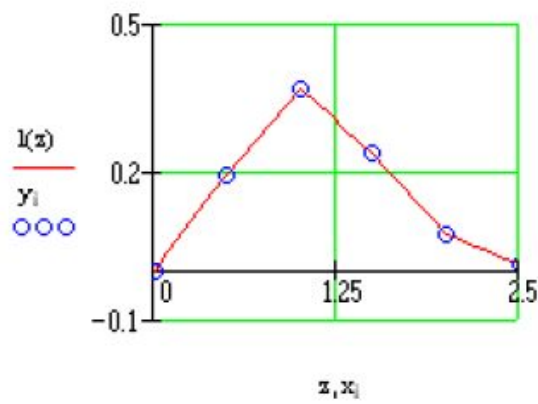
$a := 0$        $b := 2.5$        $h := \frac{a + b}{5}$

$i := 0..5$

$x_i := a + h \cdot i$        $y_i := f(x_i)$

$l(z) := \text{linterp}(x, y, z)$

$z := 0, 0.1.. 2.5$  - переменная  $z$ , определена на отрезке  $[0, 2.5]$ , но с более частым шагом, чем  $x$ , для построения графика функции  $l(z)$  в промежуточных точках между узлами интерполяции  $x_i$



$l(z)$   
—  $l(z)$   
○  $y_i$

$z, x_i$

$\text{linterp}(x, y, 0.5) = 0.195$

$\text{linterp}(x, y, 1.3) = 0.289$

$\text{linterp}(x, y, 2.2) = 0.049$

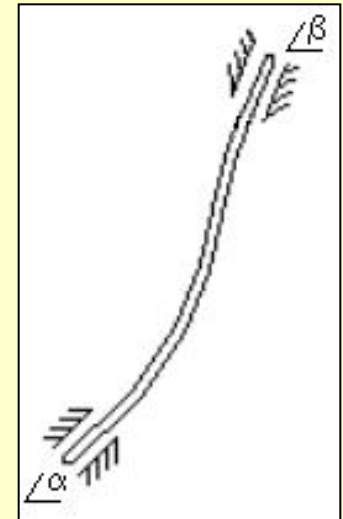
# Кубическая сплайн-интерполяция

Теория сплайнов - новый раздел современной вычислительной математики, интенсивно развивающийся в последние годы.

Линейная интерполяция, по существу, является простейшим сплайном первой степени, квадратичная интерполяция - второй.

Сплайны позволяют эффективно решать задачи обработки экспериментальных зависимостей между параметрами, имеющих достаточно сложную структуру.

Наиболее широкое практическое применение, в силу их простоты, нашли кубические сплайны. Основные идеи теории кубических сплайнов сформировались в результате попыток математически описать гибкие рейки из упругого материала (механические сплайны), которыми издавна пользовались чертежники в тех случаях, когда возникала необходимость проведения через заданные точки достаточно гладкой кривой. Известно, что рейка из упругого материала, закрепленная в некоторых точках и находящаяся в положении равновесия, принимает форму, при которой ее энергия является минимальной.



Это фундаментальное свойство позволяет эффективно использовать сплайны при решении практических задач обработки экспериментальной информации.

В общем случае для функции  $y = f(x)$  требуется найти приближение  $y = \varphi(x)$  таким образом, чтобы  $f(x_i) = \varphi(x_i)$  в точках  $x = x_i$ , а в остальных точках отрезка  $[a, b]$  значения функций  $f(x)$  и  $\varphi(x)$  были близкими между собой.

При малом числе экспериментальных точек (например, 6-8) для решения задачи интерполяции можно использовать один из методов построения интерполяционных полиномов.

При большом числе узлов интерполяционные полиномы становятся практически непригодными, так как степень интерполяционного полинома лишь на единицу меньше числа экспериментальных значений функций. Разбиение отрезка, на котором определена функция, на участки, содержащие малое число экспериментальных точек, и построение для каждого из них интерполяционного полинома приведет к появлению в аппроксимирующей функции точек, где производная не является непрерывной.

Кубические сплайны лишены этого недостатка. Функции  $\varphi(x)$ ,  $\varphi'(x)$ ,  $\varphi''(x)$  непрерывны на отрезке  $[a, b]$ .

Кубическим интерполяционным сплайном, соответствующим данной функции  $f(x)$  и данным узлам  $x_i$ , называется функция  $S(x)$ , удовлетворяющая следующим условиям:

- 1) на каждом сегменте  $[x_{i-1}, x_i]$ ,  $i = 1, 2, \dots, N$  функция  $S(x)$  является полиномом третьей степени,
- 2) функция  $S(x)$ , а также ее первая и вторая производные непрерывны на отрезке  $[a, b]$ ,
- 3)  $S(x_i) = f(x_i)$ ,  $i = 0, 1, \dots, N$ .

На каждом из отрезков  $[x_{i-1}, x_i]$ ,  $i = 1, 2, \dots, N$  будем искать функцию  $S(x) = S_i(x)$  в виде полинома третьей степени:

$$S_i(x) = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2 + d_i(x - x_{i-1})^3,$$

$$x_{i-1} \leq x \leq x_i$$

где  $a_i, b_i, c_i, d_i$  - коэффициенты, подлежащие определению на всех  $n$  элементарных отрезках. Чтобы система алгебраических уравнений имела решение, нужно, чтобы число уравнений точно равнялось числу неизвестных. Поэтому мы должны получить  $4n$  уравнения.

$$S_i(x_{i-1}) = a_i = y_{i-1},$$

$$S_i(x_i) = a_i + b_i h_i + c_i h_i^2 + d_i h_i^3 = y_i, \quad \text{где } h_i = x_i - x_{i-1}, i = 1, 2, \dots, n.$$

$$d_i = \frac{c_{i+1} - c_i}{3h_i}, i = 1, 2, \dots, n - 1, \quad d_n = -\frac{c_n}{3h_n}$$

$$b_i = \frac{y_i - y_{i-1}}{h_i} - \frac{h_i}{3} (c_{i+1} + 2c_i), i = 1, 2, \dots, n - 1, \quad b_n = \frac{y_n - y_{n-1}}{h_n} - \frac{2}{3} (h_n c_n)$$

$$h_{i-1} c_{i-1} + 2(h_{i-1} + h_i) c_i + h_i c_{i+1} = 3 \left( \frac{y_i - y_{i-1}}{h_i} - \frac{y_{i-1} - y_{i-2}}{h_{i-1}} \right), \quad i = 2, 3, \dots, n.$$

$$c_1 = 0 \quad \text{и} \quad c_{n+1} = 0.$$

## Кубическая сплайн-интерполяция в MathCAD

Проводит кривую через набор точек таким образом, что первые и вторые производные кривой непрерывны в каждой точке. Кривая образуется путем создания ряда кубических полиномов, проходящих через наборы из трех смежных точек. Кубические полиномы затем состыковываются друг с другом, чтобы образовать одну кривую.

**lspline (vx, vy)** генерирует кривую сплайна, которая приближается к прямой линии в граничных точках;

**pspline (vx, vy)** генерирует кривую сплайна, которая приближается к параболе в граничных точках.

**cspline (vx, vy)** генерирует кривую сплайна, которая может быть кубическим полиномом в граничных точках.

Функции возвращают вектор коэффициентов вторых производных (будем называть его **vs**). Аргументы **vx** и **vy** должны быть вещественными векторами одинаковой длины. Значения вектора **vx** должны быть расположены в порядке возрастания.

**interp(vs, vx, vy, x)** - возвращает интерполируемое значение  $y$ , соответствующее аргументу  $x$ . Вектор **vs** вычисляется на основе векторов данных **vx** и **vy** по одной из функций **pspline**, **lspline** или **cspline**.



## Построения кубического сплайна через набор точек:

Создать векторы  $\mathbf{vx}$  и  $\mathbf{vy}$ , содержащие координаты  $\mathbf{x}$  и  $\mathbf{y}$ , через которые нужно провести кубический сплайн. Элементы  $\mathbf{vx}$  должны быть расположены в порядке возрастания.

Вычислить вектор  $\mathbf{vs} = \mathbf{cspline}(\mathbf{vx}, \mathbf{vy})$ . Вектор  $\mathbf{vs}$  содержит вторые производные интерполяционной кривой в рассматриваемых точках. Чтобы найти интерполируемое значение в произвольной точке  $\mathbf{x0}$ , вычислите  $\mathbf{interp}(\mathbf{vs}, \mathbf{vx}, \mathbf{vy}, \mathbf{x0})$ , где  $\mathbf{vx}, \mathbf{vy}$  - векторы, описанные ранее.

*или*

Вычислить  $\mathbf{interp}(\mathbf{cspline}(\mathbf{vx}, \mathbf{vy}), \mathbf{vx}, \mathbf{vy}, \mathbf{x0})$

Для значений  $\mathbf{x}$ , расположенных перед первой точкой вектора  $\mathbf{vx}$ , MathCAD продолжает сплайн первой из составляющих его кубических парабол. Для значений  $\mathbf{x}$ , расположенных за последней точкой  $\mathbf{vx}$ , MathCAD продолжает сплайн первой из составляющих его кубических парабол.

Для получения наилучших результатов  $\mathbf{x}$  должно находиться между самыми большими и самыми маленькими значениями  $\mathbf{vx}$ . Маловероятно, что будут полезны значения, вычисленные для  $\mathbf{x}$  вне этого диапазона.

Сплайны также предназначены для интерполяции, а не для экстраполяции.

# Кубическая сплайн-интерполяция в MathCAD

**Mathcad Professional - [Кубическая интерполяция.MCD]**

Файл Правка Вид Вставка Формат Математика Символы Окно ?

Интерполяция кубическим сплайном

$f(x) := x^2 \cdot e^{-x^2}$  - Исходная функция. Зададим функцию  $f(x)$  таблично на отрезке  $[0, 2.5]$  в 6 узловых точках. Для этого:

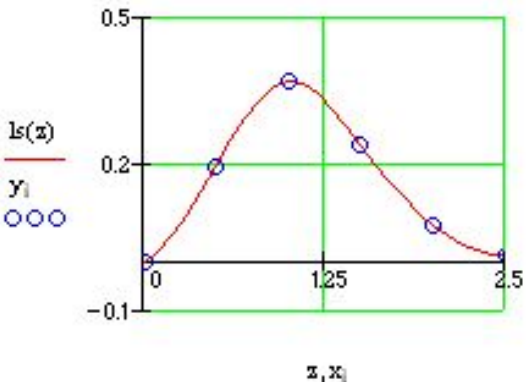
$a := 0$        $b := 2.5$        $h := \frac{a + b}{5}$

$i := 0..5$

$x_i := a + h \cdot i$        $y_i := f(x_i)$

$s := \text{cspline}(x, y)$        $ls(z) := \text{interp}(s, x, y, z)$

$z := 0, 0.1..2.5$  - переменная  $z$ , определена на отрезке  $[0, 2.5]$ , но с более частым шагом, чем  $x$ , для построения графика функции  $ls(z)$  в промежуточных точках между узлами интерполяции  $x_i$



$\text{interp}(s, x, y, 0.5) = 0.195$

$\text{interp}(s, x, y, 1.3) = 0.313$

$\text{interp}(s, x, y, 2.2) = 0.033$



## Интерполяция вектора точек

Можно использовать *оператор векторизации*, чтобы вычислить сразу целый набор интерполируемых значений, соответствующих вектору заданных точек. Это можно сделать и с *interp*, и с *linter*.

На примере показано, как выполнить эту операцию. Чтобы применить оператор векторизации к функции, щелкните мышью на имени функции и нажимайте клавишу пробел, пока в рамку не попадет нужная функция. Затем нажмите Ctrl + - (минус).

$$f(x) := x^2 \cdot e^{-x^2} \quad a := 0 \quad b := 2.5 \quad h := \frac{a+b}{5}$$

$$i := 0..5 \quad x_i := a + h \cdot i \quad y_i := f(x_i)$$

$$s := \text{cspline}(x, y) \quad z := 0, 0.1 .. 2.5$$

$$\xrightarrow{\hspace{2cm}} \\ \text{interp}(s, x, y, z) =$$

0
0.02
0.052
0.095
0.143
0.195
0.245
0.291
0.329
0.356
0.368
0.363
0.343
0.313
0.276
0.237

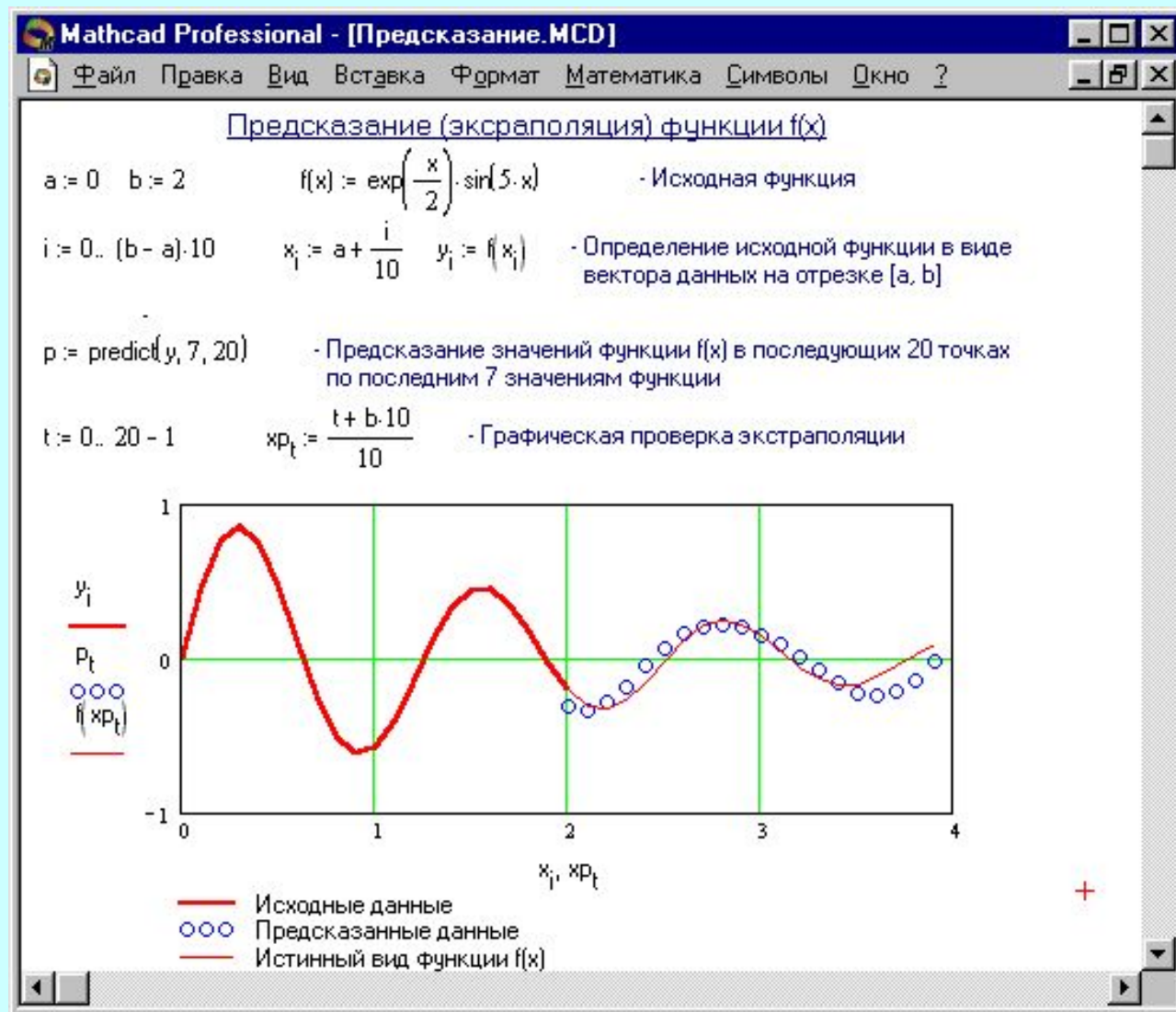
## Линейное предсказание (экстраполяция)

Функция **predict** в Mathcad позволяет оценить значения формул в точках, находящихся вне области расположения сетки, на которой заданы значения функции. Эта функция использует линейный алгоритм предсказания, который является полезным, когда экстраполируемая функция является гладкой и осциллирующей, хотя не обязательно периодической. Линейное предсказание можно рассматривать как разновидность экстраполяции, но нельзя путать с линейной или полиномиальной экстраполяцией.

**predict(v,m,n)** возвращает **n** предсказанных значений, основанных на **m** последовательных значениях вектора данных **v**. Элементы вектора **v** должны представлять собой значения, взятые через равные интервалы.

Функция **predict** использует последние **m** исходных значений данных, чтобы вычислять коэффициенты предсказания. Как только это сделано, она использует последние **m** точек, чтобы предсказать координаты **(m + 1)**-ой точки, фактически создавая скользящее окно шириной в **m** точек.

# Линейное предсказание (экстраполяция) в Mathcad



## Двумерная сплайн-интерполяция

Используют для повышения качества построения 3D-графиков. Она позволяет существенно повысить представительность сложных графиков функций, в том числе и контурных.

Выполняется кубическими сплайнами функции двух переменных тем же самым образом, как и в одномерном случае. Через сетку узлов проводится поверхность, составленная из кубических полиномов от  $x$  до  $y$ , таким образом, что первые и вторые производные являются непрерывными в каждом узле сетки.

**lspline(Mxy, Mz)** генерирует сплайн, который приближается к плоскости на краях сетки;

**pspline(Mxy, Mz)** генерирует сплайн, который представляет полином второй степени от  $x$  и  $y$  на краях сетки;

**cspline(Mxy, Mz)** генерирует сплайн, который представляет полином третьей степени от  $x$  и  $y$  на краях сетки.

Возвращают вектор коэффициентов вторых производных (будем называть **vs**) -  $n \times 2$  матрица, чьи элементы **Mxy** <sub>$i,0$</sub>  и **Mxy** <sub>$i,1$</sub> , определяют точки на диагонали  $n \times n$  сетки.  $(ij)$ -элемент  $n \times n$  матрицы **Mz** определяет значение интерполирующей функции в **(Mxy** <sub>$i,0$</sub> , **Mxy** <sub>$j,1$</sub> **)**.

**interp(vs, Mxy, Mz, x)**. Возвращает интерполируемое значение **z**, соответствующее точкам  $x = v_0$  и  $y = v_1$ . Вектор **vs** вычисляется **pspline**, **lspline** или **cspline** на основе векторов данных **Mxy** и **Mz**.

## Построения двумерной сплайн-интерполяции

Первый шаг точно такой же, как и в одномерном случае, но теперь нужно определить сетку узлов, через которые должна пройти поверхность.

Создать  $n * 2$  матрицу  $M_{xy}$ , чьи элементы,  $M_{xy} i,0$  и  $M_{xy} i,1$  определяют  $x$  и  $y$  координаты по диагонали прямоугольной сетки. Эта матрица играет ту же самую роль, что и  $v_x$  в одномерном случае. Так как эти узлы описывают диагональ, элементы в каждом столбце  $M_{xy}$  должны быть расположены в порядке возрастания ( $M_{xy} i, k < M_{xy} j, k$  всякий раз, когда  $i < j$ );

Создать  $n * n$  матрицу  $M_z$ , чей  $(ij)$ - элемент есть координата  $z$  соответствующая точке  $x = M_{xy} i,0$  и  $y = M_{xy} j,1$ . Она играет ту же самую роль, что и  $v_y$  в одномерном случае.

### Второй шаг

Вычислить вектор  $vs = \text{cspline}(M_{xy}, M_z)$  Вектор  $vs$  содержит вторые производные приближающей поверхности в узлах, определенных  $M_{xy}$  и  $M_z$ . Чтобы найти интерполируемое значение в произвольной точке  $(x_0, y_0)$ ,

вычислите

$\text{interp}(vs, M_{xy}, M_z, \begin{bmatrix} x_0 \\ y_0 \end{bmatrix})$

Результатом является значение интерполирующей функции, соответствующее точке  $(x_0, y_0)$

Можно сделать то же самое, вычисляя

$\text{interp}(\text{cspline}(M_{xy}, M_z), M_{xy}, M_z, \begin{bmatrix} x_0 \\ y_0 \end{bmatrix})$

***КОНЕЦ ЛЕКЦИИ!***