Тема №8 «Математическая статистика. Обработка данных»

Лекция 2 Интерполяция

1. **Линейная интерполяция**
2. **Кубическая интерполяция**
3. **Полиномиальная сплайн-интерполяция**
4. **Экстраполяция функцией предсказания**
5. **Многомерная интерполяция**

1 Линейная интерполяция

Когда Вы имеете дело с выборкой экспериментальных данных, то они, чаще всего, представляются в виде массива, состоящего из пар чисел *(x_i, y_i)*. Поэтому возникает задача аппроксимации дискретной зависимости $Y(x_j)$ непрерывной функцией *f(x)*. Функция *f(x),* в зависимости от специфики задачи, может быть получена тремя способамми:

- *Интерполяция,* тогда *f(x)* должна проходить через точки *(x_i)*, т.е *.f (x_i), i=1…n*. В этом случае говорят об интерполяции данных функцией *f (x)* во внутренних точках между x_i, или экстраполяции за пределами интервала, содержащего все x_i*;*
- *Сглаживание (регрессия)*. Выбор аппроксимирующей функции *f(x)* осуществляется таким образом, чтобы она сглаживала, усредняла опытные данные.
- *Сглаживание с фильтрацией данных.* Подобр аппроксимирующей функции *f(x)* осуществляется, отбросив систематическую погрешность, так называемые шумы, наложившиеся на экспериментальные данные.

Различные виды построения аппроксимирующей зависимости *f (x)* иллюстрирует рис. 1. На нем исходные данные обозначены кружками, *интерполяция* отрезками прямых линий – *пунктиром*, *линейная регрессия* – наклонной прямой линией, а *фильтрация* – жирной гладкой кривой. Эти зависимости приведены в качестве примера и отражают лишь малую часть возможностей Mathcad по обработке данных. Вообще говоря, в Mathcad имеется целый арсенал встроенных функций, позволяющий осуществлять самую различную регрессию, интерполяцию-экстраполяцию и сглаживание данных.

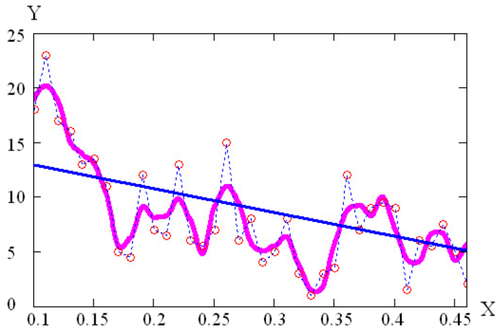


Рис. 1. Разные задачи аппроксимации данных

Для построения интерполяции-экстраполяции в Mathcad имеются несколько встроенных функций, позволяющих "соединить" точки выборки данных (x_i, y_i) кривой разной степени гладкости. По определению интерполяция означает построение функции *A(x)*, аппроксимирующей зависимость *y(x)* в промежуточных точках (между *xi*). Поэтому интерполяцию еще по-другому называют аппроксимацией. В точках x_i значения интерполяционной функции должны совпадать с исходными данными, т. е. $A(x_i) = y(x_i)$.

- **lspline(x,y)** – вектор значений коэффициентов линейного сплайна;
- **pspline(x,y)** – вектор значений коэффициентов квадратичного сплайна;
- **cspline(x,y)** – вектор значений коэффициентов кубического сплайна;
  - *x, y* – векторы данных.

Везде в этом разделе при рассказе о различных типах интерполяции будем использовать вместо обозначения *A(X)* другое имя ее аргумента *A(t)*, чтобы не путать вектор данных *x* и скалярную переменную *t*.

Линейная интерполяция

Самый простой вид интерполяции – линейная, которая представляет искомую зависимость *A(t)* в виде ломаной линии. Интерполирующая функция *A(t)* состоит из отрезков прямых, соединяющих точки (рис. 2).

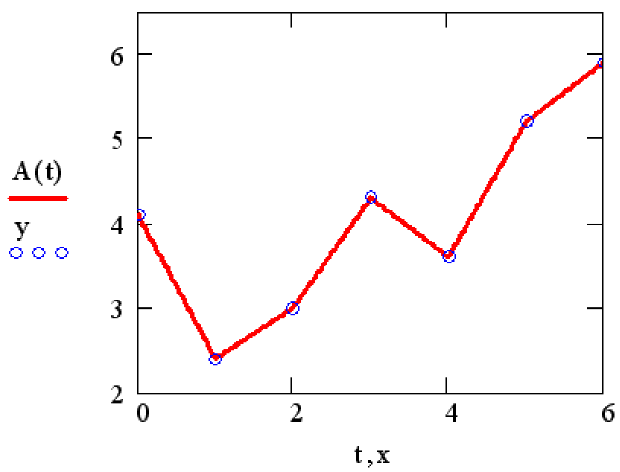


Рис. 2. Линейная интерполяция (листинг 1)

Для построения линейной интерполяции служит встроенная функция **linterp** (листинг 1).

- **linterp(x, y, t)** – функция, аппроксимирующая данные векторов x и y кусочно-линейной зависимостью;

- **x** – вектор действительных данных аргумента;
- **y** – вектор действительных данных значений того же размера;
- **t** – значение аргумента, при котором вычисляется интерполирующая функция.

Элементы вектора *x* должны быть определены в порядке возрастания, т. е. $X_1<X_2<X_3< \ldots <Xn$.

Листинг 1. Линейная интерполяция:

$$x := (0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6)^T$$

$$y := (4.1 \quad 2.4 \quad 3 \quad 4.3 \quad 3.6 \quad 5.2 \quad 5.9)^T$$

$$A(t) := \mathrm{linterp}(x, y, t)$$

Как видно из листинга, чтобы осуществить линейную интерполяцию, надо выполнить следующие действия:

- Введите векторы данных *x* и *y* (первые две строки листинга).
- Определите функцию **linterp(x,y, t).**
- Вычислите значения этой функции в требуемых точках, например linterp(x,y, 2.4)=3.52 или linterp(x,y,6) =5.9, или постройте ее график, как показано на рис. 2.

Обратите внимание, что функция *A(t)* на графике имеет аргумент *t*, а не *x*. Это означает, что функция *A(t)* вычисляется не только при значениях аргумента (т. е. в семи точках), а при гораздо большем числе аргументов в интервале (0,6), что автоматически обеспечивает Mathcad. Просто в данном случае эти различия незаметны, т. к. при обычном построении графика функции *A(x)* от векторного аргумента *x* (рис.3) Mathcad, по умолчанию, соединяет точки графика прямыми линиями (т. е. скрытым образом осуществляет их линейную интерполяцию).

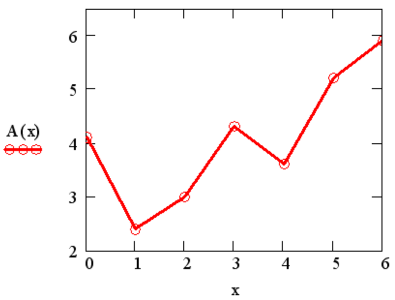


Рис. 3. Обычное построение графика функции от векторной переменной *x* (листинг 1)

2 Кубическая сплайн-интерполяция

В большинстве практических приложений желательно соединить экспериментальные точки не ломаной линией, а гладкой кривой. Лучше всего для этих целей подходит интерполяция кубическими сплайнами, т. е. отрезками кубических парабол (рис. 4).

- **interp(s,x,y,t)** – функция, аппроксимирующая данные векторов *x* и *y* кубическими сплайнами;
  - *s* – вектор вторых производных, созданный одной из сопутствующих функций cspline, pspline или lspline;
  - *x* – вектор действительных данных аргумента, элементы которого расположены в порядке возрастания;
  - *y* – вектор действительных данных значений того же размера;
  - *t* – значение аргумента, при котором вычисляется интерполирующая функция.

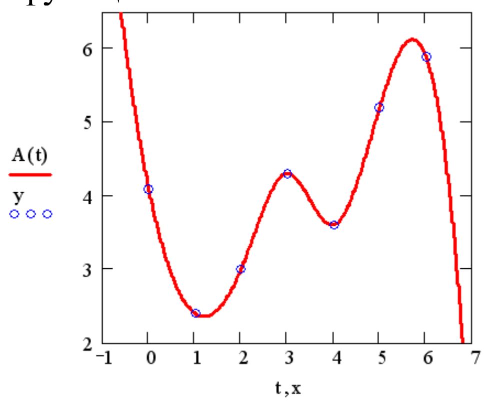


Рис. 4. Сплайн-интерполяция (см. листинг 2)

Сплайн-интерполяция в Mathcad реализована чуть сложнее линейной. Перед применением функции **interp** необходимо предварительно определить первый из ее аргументов – векторную переменную *s*. Делается это при помощи одной из трех встроенных функций тех же аргументов *(x,y)*.

Пример сплайн-интерполяции приведен в листинге 2.

Листинг 2. Кубическая сплайн-интерполяция:

$$x := (0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6)^T$$

$$y := (4.1 \quad 2.4 \quad 3 \quad 4.3 \quad 3.6 \quad 5.2 \quad 5.9)^T$$

$$s := \mathrm{cspline}(x, y)$$

$$A(t) := \mathrm{interp}(s, x, y, t)$$

Смысл сплайн-интерполяции заключается в том, что в промежутках между точками осуществляется аппроксимация в виде зависимости

$A(t) = a\,t^3 + bt^2 + ct + d$. Коэффициенты *a, b, c, d* рассчитываются независимо для каждого промежутка, исходя из значений *y(x)* в соседних точках. Этот процесс скрыт от пользователя, поскольку смысл задачи интерполяции состоит в выдаче значения *A(t)* в любой точке *t* (рис.4).

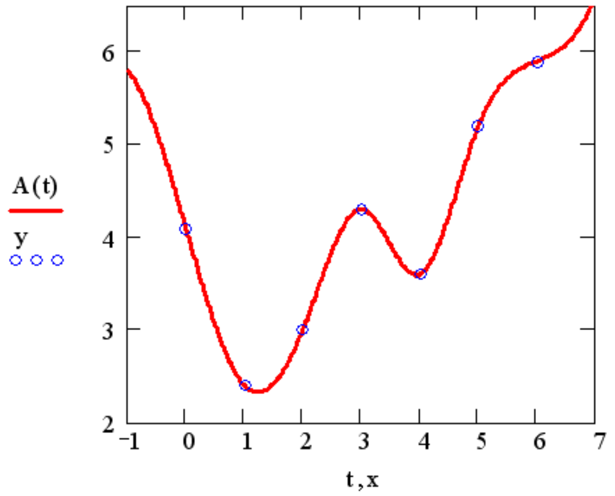
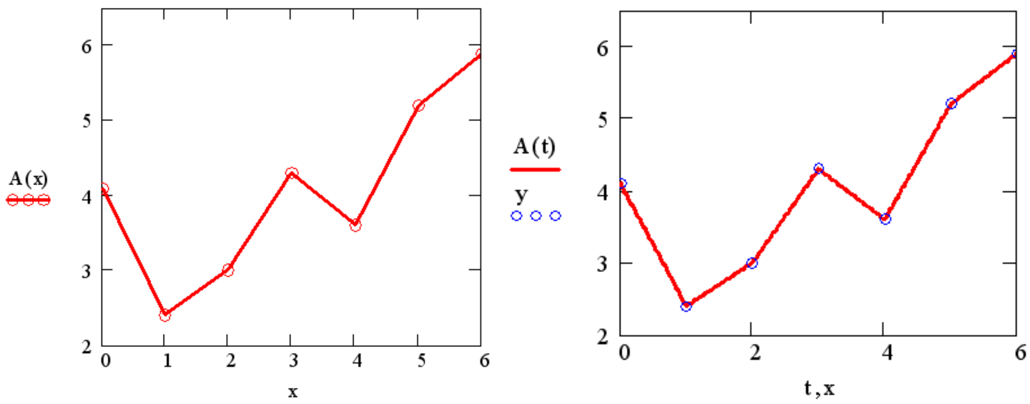


Рис. 5. Сплайн-интерполяция с выбором коэффициентов линейного сплайна lspline

Чтобы подчеркнуть различия, соответствующие разным вспомогательным функциям **cspline**, **pspline**, **Ispline**, покажем результат действия листинга 2 при замене функции **cspline** в предпоследней строке на линейную **ispline** (рис.5). Как видно, выбор вспомогательных функций существенно влияет на поведение *A(t)* вблизи граничных точек рассматриваемого интервала (0, 6) и особенно разительно меняет результат экстраполяции данных за его пределами.



а б

Рис. 6. а - Ошибочное построение графика сплайн-интерполяции (см. листинг 2), б - правильное

В заключение остановимся на уже упоминавшейся в вопросе 1 распространенной ошибке при построении графиков интерполирующей функции (см. рис.3). Если на графике, например являющемся продолжением листинга 2, задать построение функции *A(X)* вместо *A(t)*, то будет получено просто соединение исходных точек ломаной (рис. 6). Так происходит потому, что в промежутках между точками вычисления интерполирующей функции не производятся.

3. Полиномиальная сплайн-интерполяция

Более сложный тип интерполяции – так называемая интерполяция B-сплайнами. В отличие от обычной сплайн-интерполяции, сшивка элементарных *B*-сплайнов производится не в точках x_i, а в других точках u_i, координаты которых предлагается ввести пользователю. Сплайны могут быть полиномами 1, 2 или 3 степени (линейные, квадратичные или кубические). Применяется интерполяция *B*-сплайнами точно так же, как и обычная сплайн-интерполяция, различие состоит только в определении вспомогательной функции коэффициентов сплайна.

- **interp(s,x,y, t)** – функция, аппроксимирующая данные векторов *x* и *y* с помощью *B*-сплайнов;
- *bspline(x,y,u,n)* – вектор значений коэффициентов B-сплайна;
  - *s* – вектор вторых производных, созданный функцией bspline;
  - *x* – вектор действительных данных аргумента, элементы которого расположены в порядке возрастания;
  - *y* – вектор действительных данных значений того же размера;
  - *t* – значение аргумента, при котором вычисляется интерполирующая функция;
  - *u* – вектор значений аргумента, в которых производится сшивка B-сплайнов;
  - *n* – порядок полиномов сплайновой интерполяции (1, 2 или 3).

Размерность вектора и должна быть на 1, 2 или на 3 меньше размерности векторов *x* и *y*. Первый элемент вектора и должен быть меньше или равен первому элементу вектора *x*, а последний элемент – больше или равен последнему элементу *x*.

Интерполяция *B*-сплайнами иллюстрируется листингом 3 и рис. 7.

Листинг 3. Интерполяция B-сплайнами:

$$x := (0 \;\; 1 \;\; 2 \;\; 3 \;\; 4 \;\; 5 \;\; 6)^T$$

$$y := (4.1 \;\; 2.4 \;\; 3 \;\; 4.3 \;\; 3.6 \;\; 5.2 \;\; 5.9)^T$$

$$u := (-0.5 \;\; 2.2 \;\; 3.3 \;\; 4.1 \;\; 5.5 \;\; 7)^T$$

$$s := \mathrm{bspline}(x, y, u, 2)$$

$$A(t) := \mathrm{interp}(s, x, y, t)$$

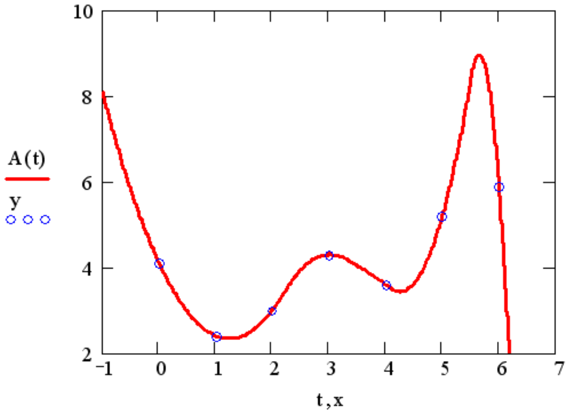


Рис. 7. B-сплайн-интерполяция (листинг 3)

4. Экстраполяция функцией предсказания

Все рассмотренные выше функции осуществляли экстраполяцию данных за пределами их интервала с помощью соответствующей зависимости, основанной на анализе расположения нескольких исходных точек на границах интервала. В Mathcad имеется более развитый инструмент экстраполяции, который учитывает распределение данных вдоль всего интервала. В функцию **predict** встроен линейный алгоритм предсказания поведения функции, основанный на анализе, в том числе осцилляции.

- **predict (y,m,n)** – функция предсказания вектора, экстраполирующего выборку данных;
  - *y* – вектор действительных значений, взятых через равные промежутки значений аргумента;
  - *m* – количество последовательных элементов вектора y, согласно которым строится экстраполяция;
  - *n*– количество элементов вектора предсказаний.

Пример использования функции предсказания на примере экстраполяции осциллирующих данных y_i с меняющейся амплитудой приведен в листинге 4. Полученный график экстраполяции, наряду с самой функцией, показан на рис. 8. Аргументы и принцип действия функции **predict** отличаются от рассмотренных выше встроенных функций интерполяции-экстраполяции. Значений аргумента для данных не требуется, поскольку по определению функция действует на данные, идущие друг за другом с равномерным шагом. Обратите внимание, что результат функции **predict** вставляется "в хвост" исходных данных.

Листинг 4. Экстраполяция при помощи функции предсказания:

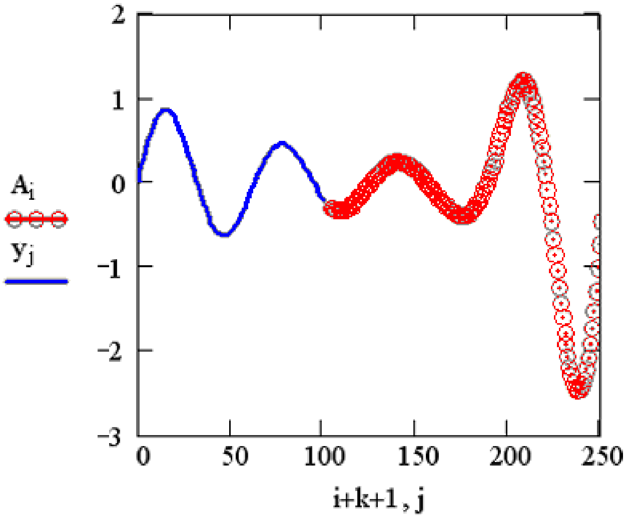
$$k := 100 \qquad\qquad j := 0..k$$

$$y_j := e^{\frac{-j}{100}} \cdot \sin\left(\frac{j}{10}\right)$$

$$m := 50 \qquad\qquad n := 150$$

$$A := \mathrm{predict}(y, m, n)$$

$$i := 0..n+k$$



A =

	0
0	-0.228
1	-0.252
2	-0.274
3	-0.293
4	-0.308
5	-0.32
6	-0.328
7	-0.333
8	-0.335
9	-0.333
10	-0.328
11	-0.319
12	-0.308
13	-0.294
14	-0.277
15	-0.258

Рис. 8. Экстраполяция при помощи функции предсказания (листинг 4)

Как видно из рис. 9, функция предсказания может быть полезна при экстраполяции данных на небольшие расстояния. Вдали от исходных данных результат часто бывает неудовлетворительным. Кроме того, функция **predict** хорошо работает в задачах анализа подробных данных с четко прослеживающейся закономерностью (рис. 8), в основном осциллирующего характера.

Если данных мало, то предсказание может оказаться бесполезным. В листинге 5 приведена экстраполяция небольшой выборки данных (из примеров, рассмотренных в предыдущих разделах). Соответствующий результат показан на рис. 9 для различных крайних точек массива исходных данных, для которых строится экстраполяция.

Листинг 5. Экстраполяция при помощи функции предсказания:

$$y := (4.1 \;\; 2.4 \;\; 3 \;\; 4.3 \;\; 3.6 \;\; 5.2 \;\; 5.9)^T$$

$$k := \mathrm{rows}(y) - 1$$

$$m := 5 \qquad\qquad n := 10$$

$$A := \mathrm{predict}(y, m, n)$$

$$i := 0..n+k \qquad j := 0..k$$

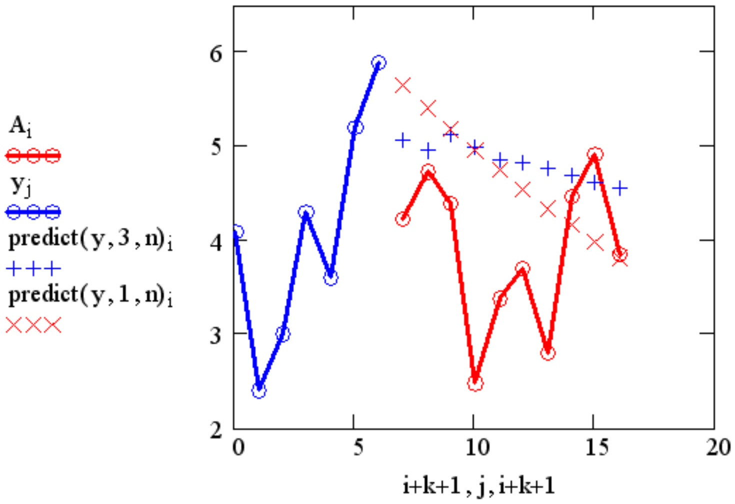


Рис. 9. Работа функции предсказания в случае малого количества данных (листинг 5)

5. Многомерная интерполяция

Двумерная сплайн-интерполяция приводит к построению поверхности *z(x,y)*, проходящей через массив точек, описывающий сетку на координатной плоскости *(x,y)*. Поверхность создается участками двумерных кубических сплайнов, являющихся функциями *(x,y)* и имеющих непрерывные первые и вторые производные по обеим координатам.

Многомерная интерполяция строится с помощью тех же встроенных функций, что и одномерная, но имеет в качестве аргументов не векторы, а соответствующие матрицы. Существует одно важное ограничение, связанное с возможностью интерполяции только квадратных *N* x *N* массивов данных.

- **interp(s,x,z,v)** – скалярная функция, аппроксимирующая данные выборки двумерного поля по координатам *x* и *y* кубическими сплайнами;
  - *s* – вектор вторых производных, созданный одной из сопутствующих функций *cspline*, *pspline* или *lspline*;
  - *x* – матрица размерности *N* x *2* (в две строки), определяющая диагональ сетки значений аргумента (элементы обоих столбцов соответствуют меткам *x* и *y* и расположены в порядке возрастания);
  - *z* – матрица действительных данных размерности *N* x *N*;

- v – вектор из двух элементов, содержащий значения аргументов x и y, для которых вычисляется интерполяция.

Вспомогательные функции построения вторых производных имеют те же матричные аргументы, что и interp: Ispline (X,Y), pspline (X, У), cspline(X,Y).

Пример исходных данных приведен на рис. 10 в виде графика линий уровня, программная реализация двумерной интерполяции показана в листинге 6, а ее результат – на рис. 11.

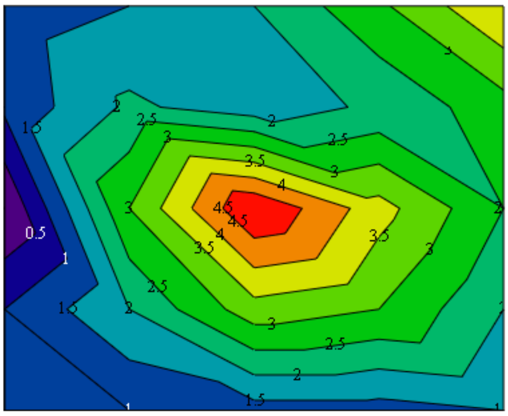


Рис. 10. Исходное двумерное поле данных (листинг 6)

Листинг 6. Двумерная интерполяция:

$$X := \begin{pmatrix} 0 & 0 \\ 1 & 10 \\ 2 & 20 \\ 3 & 30 \\ 4 & 40 \end{pmatrix} \qquad Y := \begin{pmatrix} 1 & 1 & 0 & 1.1 & 1.2 \\ 1 & 2 & 3 & 2.1 & 1.5 \\ 1.3 & 3.3 & 5 & 1.7 & 2 \\ 1.3 & 3 & 3.7 & 2.1 & 2.9 \\ 1.5 & 2 & 2.5 & 2.8 & 4 \end{pmatrix}$$

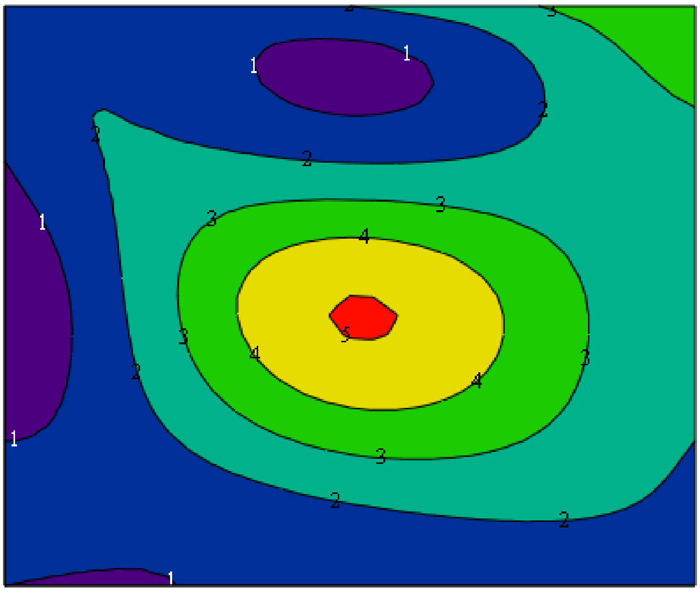
$$S := \mathrm{cspline}(X, Y)$$

$$V := \begin{pmatrix} 3.7 \\ 2.2 \end{pmatrix} \qquad \mathrm{interp}(S, X, Y, V) = 1.636$$

$$k := 30$$

$$i := 0..k \qquad j := 0..k$$

$$A_{i,j} := \mathrm{interp}\left[S, X, Y, \begin{pmatrix} \frac{i}{k} \cdot 4 \\ \frac{j}{k} \cdot 40 \end{pmatrix}\right]$$



А

Рис. 11. Результат двумерной интерполяции (листинг 6)

Таким образом, в лекции рассмотрены различные способы аппроксимации данных (интерполяции), которые находят свое применение при решении таких задач как анализ и синтез избирательных радиотехнических устройств, реализации алгоритмов восстановления цифровых сигналов и т.д.