Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное   
учреждение высшего профессионального образования

Нижегородский государственный университет им. Н.И. Лобачевского

Факультет информационных технологий математики и механики

**Отчет по лабораторной работе**

**Построение кубических сплайнов**

**Выполнил**:студент группы 381606-1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кутовой В.Н.

Подпись

**Научный руководитель**:

к.ф.-м.н.,

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Эгамов А.И

Подпись

Нижний Новгород

2018

Содержание

[Введение 3](#_Toc530936925)

[Изложение метода 4](#_Toc530936926)

[Интерполяция кубическими сплайнами 4](#_Toc530936927)

[Метод прогонки 5](#_Toc530936928)

[Блок-схема алгоритма 7](#_Toc530936929)

[Пример работы программы 8](#_Toc530936930)

[Заключение 9](#_Toc530936931)

[Список литературы 10](#_Toc530936932)

# Введение

**Сплайн** — функция, область определения которой разбита на конечное число отрезков, на каждом из которых она совпадает с некоторым алгебраическим многочленом (полиномом). Максимальная из степеней использованных полиномов называется степенью сплайна. Разность между степенью сплайна и получившейся гладкостью называется дефектом сплайна. Например, непрерывная ломаная есть сплайн степени 1 и дефекта 1. В современном понимании сплайны — это решения многоточечных краевых задач сеточными методами.

Другими словами сплайн — это кусочно заданная функция, то есть совокупность нескольких функций, каждая из которых задана на каком-то множестве значений аргумента, причём эти множества попарно непересекающиеся.

Сплайны имеют многочисленные применения как в математической теории, так и в прикладной математике (в частности, в разнообразных вычислительных программах). В частности, сплайны двух переменных интенсивно используются для задания поверхностей в различных системах компьютерного моделирования. Сплайны двух аргументов называют би-сплайнами (например, бикубический сплайн), которые являются двумерными сплайнами, моделирующими поверхности. Их часто путают с B-сплайнами (базисными сплайнами), которые являются одномерными и в линейной комбинации составляют кривые — каркас для «натягивания» поверхностей. Также из базисных сплайнов возможно составить трёхмерную конструкцию для моделирования объёмных тел.

**Кубический сплайн** — функция, область определения которой разбита на конечное число отрезков, на каждом из которых она совпадает с некоторым кубическим многочленом (полиномом).

Передо мной стоит задача разработать программу, которая бы обрабатывала набор точек и отрисовывала полученный кубический сплайн.

# Изложение метода

## Интерполяция кубическими сплайнами

Интерполяция кубическими сплайнами является частным случаем кусочно-полиномиальной интерполцией. В этом специальном случае между любыми двумя соседними узлами функция интерполируется кубическим полиномом. его коэффициенты на каждом интервале определяются из условий сопряжения в узлах:

f_i=y_i,

f'(x_i-0)=f'(x_i+0),

f''(x_i-0)=f''(x_i+0), i=1, 2, \cdots, n-1.

Кроме того, на границе при x=x_0 и x=x_n ставятся условия

( 2 ) f''(x_0)=0, f''(x_n)=0.

Будем искать кубический полином в виде

( 3 ) f(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}\le \x\le \x_i.

Из условия f_i=y_i имеем

( 4 )

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

f(x_i)=a_i+b_ih_i+c_ih_i^2+d_ih_i^3=y_i,

h_i=x_i-x_{i-1}, i=1, 2, \cdots, n-1.

Вычислим производные:

f'(x)=b_i+2c_i(x-x_{i-1})+3d_i(x-x_{i-1})^2,

f''(x)=2c_i+6d_i(x-x_{i-1}), x_{i-1}\le \x\le \x_i,

и потребуем их непрерывности при x=x_i:

( 5 ) b_{i+1}=b_i+2c_ih_i + 3d_ih_i^2,

c_{i+1}=c_i+3d_ih_i, i=1, 2, \cdots, n-1.

Общее число неизвестных коэффициентов, очевидно, равно 4n, число уравнений [(4)](http://www.machinelearning.ru/wiki/index.php?title=%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BF%D0%BE%D0%BB%D1%8F%D1%86%D0%B8%D1%8F_%D0%BA%D1%83%D0%B1%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%BC%D0%B8_%D1%81%D0%BF%D0%BB%D0%B0%D0%B9%D0%BD%D0%B0%D0%BC%D0%B8#eq:4) и [(5)](http://www.machinelearning.ru/wiki/index.php?title=%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BF%D0%BE%D0%BB%D1%8F%D1%86%D0%B8%D1%8F_%D0%BA%D1%83%D0%B1%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%BC%D0%B8_%D1%81%D0%BF%D0%BB%D0%B0%D0%B9%D0%BD%D0%B0%D0%BC%D0%B8#eq:5) равно 4n-2. Недостающие два уравнения получаем из условия [(2)](http://www.machinelearning.ru/wiki/index.php?title=%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BF%D0%BE%D0%BB%D1%8F%D1%86%D0%B8%D1%8F_%D0%BA%D1%83%D0%B1%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%BC%D0%B8_%D1%81%D0%BF%D0%BB%D0%B0%D0%B9%D0%BD%D0%B0%D0%BC%D0%B8#eq:2) при x=x_0 и x=x_n:

c_1=0, c_n+3d_nh_n=0.

Выражение из (5) d_i=\frac{c_{i+1}-c_i}{3h_i}, подставляя это выражение в (4) и исключая a_i=y_{i-1}, получим

b_i=\[\frac{y_i-y_{i-1}}h_i\]-\frac{1}{3}h_i(c_{i+1}+2c_i),  i=1, 2, \cdots, n-1,

b_n=\[\frac{y_n-y_{n-1}}h_n\]-\frac{2}{3}h_nc_n,.

Подставив теперь выражения для b_i, b_{i+1} и d_i в первую формулу (5), после несложных преобразований получаем для определения c_i разностное уравнение второго порядка

( 6 )

h_ic_i+2(h_i+h_{i+1})c_{i+1}+h_{i+1}c_{i+2}=3\left(\frac{y_{i+1}-y_i}{h_{i+1}} - \frac{y_i-y_{i-1}}{h_i}\right), i=1, 2, \cdots, n-1.

С краевыми условиями

( 7 ) c_1=0, c_{n+1}=0.

Условие c_{n+1}=0 эквивалентно условию c_n+3d_nh_n=0 и уравнению c_{i+1} = c_i+d_ih_i. Разностное уравнение (6) с условиями (7) можно решить методом прогонки, представив в виде системы линейных алгебраических уравнений вида ~A*x=F, где вектор x соответствует вектору \{c_i\}, вектор F поэлементно равен правой части уравнения (6).  Разностное уравнение (6) с условиями (7) можно решить методом прогонки, представив в виде системы линейных алгебраических уравнений вида ~A*x=F, где вектор x соответствует вектору \{c_i\}, вектор F поэлементно равен правой части уравнения (6), а матрица ~A имеет следующий вид:

где A_i=h_i,  i=2, \cdots, n,  B_i = h_{i+1},  i=1, \cdots, n-1 и C_i=2(h_i+h_{i+1}), i =1, \cdots, n.

## 

## Метод прогонки

**Метод прогонки**, основан на предположении, что искомые неизвестные связаны рекуррентным соотношением:

( 8 ) x_i = \alpha_{i+1}x_{i+1} + \beta_{i+1}\, i=1,\cdots,n-1

Используя это соотношение, выразим x_{i-1} и x_i через x_{i+1} и подставим в i-e уравнение:

\left(A_i\alpha_i\alpha_{i+1} + C_i\alpha_{i+1} + B_i\right)x_{i+1} + A_i\alpha_i\beta_{i+1} + A_i\beta_i + C_i\beta_{i+1} - F_i = 0

где F_i - правая часть i-го уравнения. Это соотношение будет выполняться независимо от решения, если потребовать

A_i\alpha_i\alpha_{i+1} + C_i\alpha_{i+1} + B_i = 0

A_i\alpha_i\beta_{i+1} + A_i\beta_i + C_i\beta_{i+1} - F_i = 0

Отсюда следует:

\alpha_{i+1} = {-B_i \over A_i\alpha_i + C_i}\

\beta_{i+1} = {F_i - A_i\beta_i \over A_i\alpha_i + C_i}

Из первого уравнения получим:

\alpha_2 = {-B_1 \over C_1}

\beta_2 = {F_1 \over C_1}

После нахождения прогоночных коэффициентов \alpha и \beta, используя уравнение (1), получим решение системы. При этом,

x_n = {F_n-A_n\beta_n \over C_n+A_n\alpha_n} 

# Блок-схема алгоритма

СЧИТЫВАНИЕ КООРДИНАТ

ПРЯМАЯ ПРОГОНКА

ОБРАТНАЯ ПРОГОНКА

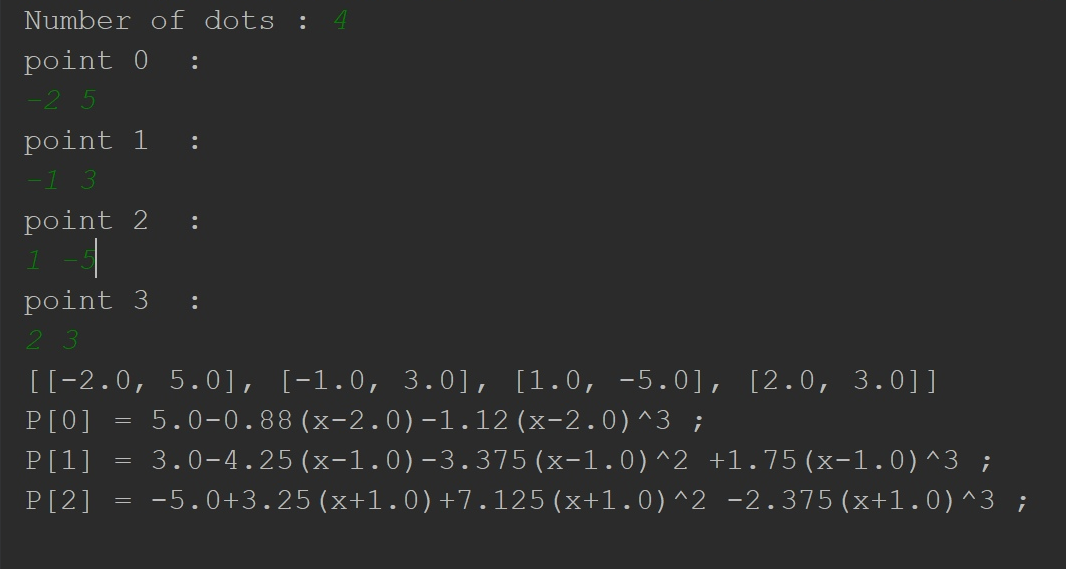
ВЫВОД ПОЛИНОМОВ

ОТРИСОВКА СПЛАЙНА

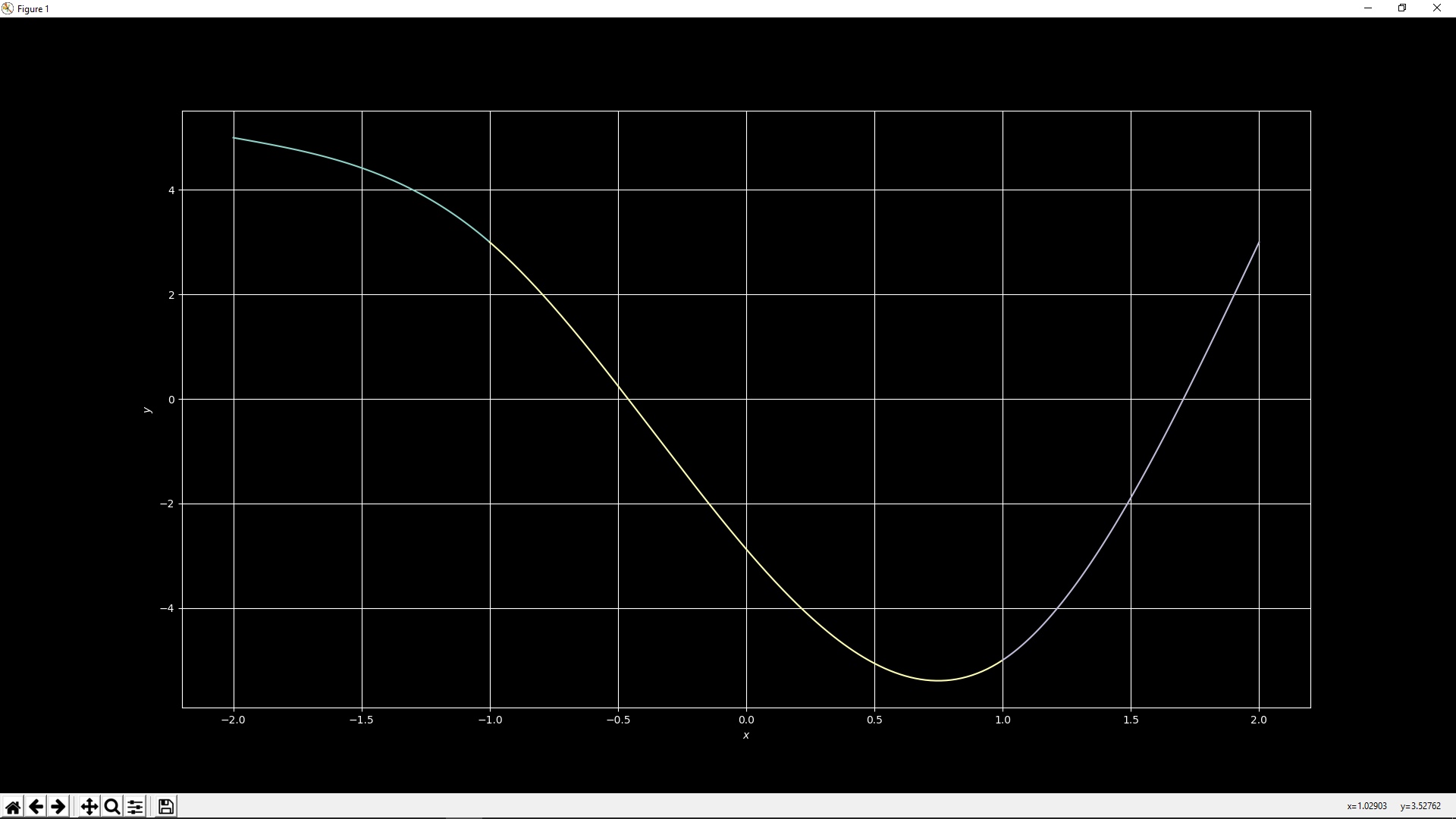
# Пример работы программы

В начале работы программы пользователь выбирает тип ввода точек: чтение из файла или ввод вручную через консоль.

Ввод точек пользователем:



Результат:



# Заключение

Мне удалось выполнить поставленную задачу. Программа может считывать набор точек как из консоли, так и из файла, а потом выводить полученные результаты в другой файл и на экран в удобной для пользователя форме.

# Список литературы

* *А.А.Самарский.*  Введение в численные методы М.: Наука, 1982
* А.А.Самарский, А.В.Гулин. Численные методы М.: Наука, 1989
* http://www.machinelearning.ru/