

Министерство науки и высшего образования Российской Федерации  
НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

---

А.Г. ЗАДОРЖНЫЙ, Д.С. КИСЕЛЕВ

# ПОСТРОЕНИЕ СПЛАЙНОВ С ИСПОЛЬЗОВАНИЕМ БИБЛИОТЕКИ OPENGL

Утверждено  
Редакционно-издательским советом университета  
в качестве учебного пособия

НОВОСИБИРСК  
2019

УДК 004.92(075.8)  
3-156

Рецензенты:  
канд. техн. наук, доцент *В.С. Карманов*  
доктор техн. наук, профессор *М.Э. Рояк*

Работа подготовлена на кафедре прикладной математики НГТУ

**Задорожный А.Г.**

З-156 Построение сплайнов с использованием библиотеки OpenGL:  
учебное пособие / А.Г. Задорожный, Д.С. Киселев. – Новоси-  
бирск: Изд-во НГТУ, 2019. – 88 с.

ISBN 978-5-7782-4069-8

В данном учебном пособии рассмотрены элементы теории построения сплайнов и соответствующие функции графической библиотеки OpenGL. Пособие может быть рекомендовано как для самостоятельного изучения курсов «Компьютерная графика» и «Численные методы», так и для подготовки к лабораторным, практическим и расчетно-графическим заданиям.

УДК 004.92(075.8)

**ISBN 978-5-7782-4069-8**

© Задорожный А.Г., Киселев Д.С, 2019

© Новосибирский государственный  
технический университет, 2019

## ОБЩИЕ СВЕДЕНИЯ

Исторически, сплайн (spline) – это металлическая линейка (лекало) для соединения точек плавной кривой (интерполяции) на чертеже, которое невозможно построить при помощи циркуля или линейки (например, эллипс или спираль). В этом случае для каждого участка требуемой кривой по контрольным точкам подбирается подходящая кривая на лекале и проводится нужная линия (на рис.1 приведено схематическое изображение подобной гибкой линейки). Изучением форм подобных лекал занимался еще Леонардо да Винчи. Лекало постоянной кривизны представляет собой шаблон, содержащий различные кривые переменного радиуса, а лекало переменной кривизны – линейку с изменяемой кривизной, примеры таких лекал приведены на рис.2. Сейчас подобные лекала используются, например, для составления выкроек одежды

В математике же под сплайном понимают кривую, описывающую деформацию гибкой линейки, зафиксированной в контрольных точках. Таким образом, сплайн-функция является математической моделью гибкой линейки. Родоначальником теории сплайнов стал американский математик Исаак Шенберг, который данными математическими объектами занимался в середине двадцатого столетия. Сплайны оказались очень удобным инструментом для задач интерполяции в теории приближения. К достоинствам сплайном можно отнести их прекрасные аппроксимационные свойства, универсальность и простоту реализации.

Однако сплайны используются не только для описания некоторых кривых, во многих случаях сплайны являются еще и решением дифференциальных уравнений, описывающих вполне реальные физические процессы. Например, сплайнами описываются траектория искусственного объекта с активными и инерционными отрезками движения, распространение тепловых и электромагнитных полей через разнородные среды, диффузия через слои различных веществ и т.п. Таким образом, в современном понимании сплайны – это решения многоточечных

краевых задач сеточными методами. В частности, решения тепловых и электромагнитных задач методом конечных элементов (МКЭ) представляют собой обычно линейные, квадратичные или кубические сплайн-функции.

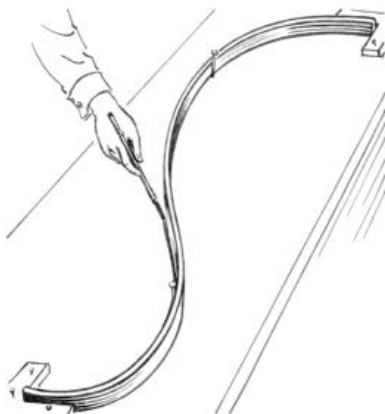


Рис.1. Гибкая металлическая линейка для рисования кривых ([https://ru.qwertyu.wiki/wiki/Non-uniform\\_rational\\_B-spline](https://ru.qwertyu.wiki/wiki/Non-uniform_rational_B-spline)).



Fig. 6. Kurvenlineale von Gebrüder Wichmann, Berlin.



Рис.2. Примеры наборов лекал (<https://ru.wikipedia.org/wiki/Лекало>).

## МНОГОЧЛЕНЫ

*Одночлен (моном)* – это произведение, состоящее из числового множителя и  $m$  переменных, взятых в неотрицательной целой степени:

$$O(x_1, x_2, \dots, x_m) = c \cdot \prod_{k=1}^m x_k^{s_k} = c \cdot x_1^{s_1} \cdot x_2^{s_2} \cdot \dots \cdot x_m^{s_m},$$

где множитель  $c \neq 0$  называется *коэффициентом одночлена*, а множество индексов  $S = \{s_1, s_1, \dots, s_m\}$  – *мультииндексом*.

*Степенью* одночлена называется сумма показателей степеней всех его переменных (сумма всех элементов мультииндекса):

$$s_1 + s_2 + \dots + s_m.$$

Одночлен с  $m = 0$  называется *свободным членом* и представляет собою вещественное число, при этом число 0 является одночленом с неопределенной степенью (*нулевым одночленом*).

*Многочлен (полином)* от  $m$  переменных – это линейная комбинация  $k$  одночленов:

$$M(x_1, x_2, \dots, x_n) = \sum_{i=1}^k \left( c_i \cdot \prod_{j=1}^n x_j^{s_{ij}} \right).$$

*Степенью многочлена* называется максимальная из степеней его одночленов.

При  $k = 2$  многочлен состоит из двух одночленов и называется *двучленом* или *биномом*, классическим примером является бином Ньютона. В случае  $k = 3$  многочлен состоит из трех одночленов и называется *трехчленом*.

В задачах геометрического программирования понятие монома расширяется (в частности, разрешается использование вещественных чисел в показателях), а линейная комбинация таких расширенных мономов называется *позиномом*.

## БИНОМ НЬЮТОНА

*Бином Ньютона* – это формула построения многочлена для целой неотрицательной степени  $n$  суммы двух переменных  $x$  и  $y$ :

$$(x + y)^n = \sum_{k=0}^n C_n^k \cdot x^{n-k} \cdot y^k,$$

где  $C_n^k = \frac{n!}{k! \cdot (n-k)!}$  – это *биномиальные коэффициенты*, которые могут быть обобщены на область вещественных чисел.

Поскольку непосредственное вычисление коэффициента  $C_n^k$  имеет факториальную сложность расчетов, то были разработаны различные алгоритмы его получения. Один из них – это использование формулы  $C_n^k = C_{n-1}^k + C_{n-1}^{k-1}$  при условии хранения всех получаемых значений. Другой вариант – использование *формулы Стирлинга* (Муавра-Стирлинга), которая является приближением, полученным из разложения факториала в ряд Стирлинга с использованием классических чисел  $\pi$  и  $e$ :

$$n! = \sqrt{2 \cdot \pi \cdot n} \cdot \left(\frac{n}{e}\right)^n \cdot \left(1 + \frac{1}{12 \cdot n} + \frac{1}{288 \cdot n^2} - \frac{139}{51840 \cdot n^3} - \frac{571}{2488320 \cdot n^4}\right) + O(n^{-5}).$$

Бином Ньютона, в свою очередь, может быть обобщен до *полинома Ньютона*, представляющего собой возведение в степень  $n$  суммы  $m$  слагаемых:

$$\left(\sum_{i=1}^m x_i\right)^n = \sum_{\{k_i\}} \left(C_n^{k_1, k_2, \dots, k_m} \cdot \prod_{j=1}^m x_j^{k_j}\right),$$

где:

$\{k_i \geq 0, i = \overline{1, m}\}$  – это всевозможные уникальные комбинации целых неотрицательных чисел  $k_i$ , сумма которых равна  $n$ ;

$C_n^{k_1, k_2, \dots, k_m} = \frac{n!}{k_1! \cdot k_2! \cdot \dots \cdot k_m!}$  – это *мультиномиальные (полиномиальные) коэффициенты*, частным случаем которых и являются биномиальные коэффициенты.

## АЛГЕБРАИЧЕСКИЕ ПОЛИНОМЫ

*Алгебраический полином степени  $n$*  — это многочлен степени  $n$  от одной переменной:

$$P_n(x) = \prod_{k=0}^n a_k \cdot x^k = a_0 + a_1 \cdot x + a_2 \cdot x^2 + \dots + a_n \cdot x^n,$$

причем  $a_n \neq 0$ .

Коэффициент  $a_n$  называется *старшим коэффициентом*, одночлен  $a_n \cdot x^n$  — *старшим членом*, а коэффициент  $a_0$  — *свободным членом*. Если  $a_n = 1$ , тогда алгебраический полином называется *нормированным*.

Число  $\hat{x}$  является *корнем* полинома  $P_n(x)$ , если  $P_n(\hat{x}) = 0$ . По теореме Гаусса любой алгебраический полином имеет хотя бы один корень (действительный или комплексный).

По теореме Безу для  $\hat{x}$  выполняется следующее равенство (деление многочлена без остатка):

$$P_n(x) = (x - \hat{x}) \cdot Q_{n-1}(x),$$

где  $Q_{n-1}(x)$  — это алгебраический полином степени  $n-1$ .

Таким образом, число  $\hat{x}$  является *корнем кратности  $r$* , если  $P_n(\hat{x})$  можно представить в виде

$$P_n(x) = (x - \hat{x})^r \cdot Q_{n-r}(x),$$

причем  $Q_{n-r}(x) \neq 0$ .

Можно убедиться, что корень кратности  $r$  по отношению к производной полинома  $P_n(x)$  является корнем кратности  $r-1$ .

Исходя из вышеизложенного, полином  $P_n(x)$  может быть разложен на линейные множители:

$$P_n(x) = \prod_{k=0}^m p_k \cdot (x - x_k)^{r_k},$$

где  $x_j$  — это корни кратности  $r_j$ , причем  $\sum_{k=1}^m r_k = n$ .

Отношение (дробь) двух алгебраических полиномов  $Q_m(x)$  и  $P_n(x)$  называется *дробно-рациональной функцией*  $f(x)$ :

$$f(x) = \frac{Q_m(x)}{P_n(x)}.$$

Соответственно, при  $m < n$  такая дробь называется *правильной*, а при  $m = n = 1$  такая функция становится *дробно-линейной*.

## МНОГОЧЛЕНЫ ЧЕБЫШЕВА

*Многочлены Чебышева* — это две последовательности ортогональных многочленов  $T_n(x)$  и  $U_n(x)$ , названные в честь русского математика Пафнутия Чебышева.

*Многочлен Чебышева первого рода*  $T_n(x)$  характеризуется как многочлен степени  $n$  со старшим коэффициентом  $2^{n-1}$ , который меньше всего отклоняется от нуля на отрезке  $[-1, 1]$ :

$$T_n(x) = \sum_{k=0}^{[n/2]} C_{2k}^n \cdot (x^2 - 1)^k \cdot x^{n-2k}.$$

*Многочлен Чебышева второго рода*  $U_n(x)$  характеризуется как многочлен степени  $n$  со старшим коэффициентом  $2^n$ , интеграл от абсолютной величины которого по отрезку  $[-1, 1]$  принимает минимально возможное значение:

$$U_n(x) = \sum_{k=0}^{[n/2]} C_{2k+1}^{n+1} \cdot (x^2 - 1)^k \cdot x^{n-2k}.$$

Многочлены Чебышева первого рода играют важную роль в теории приближений, поскольку их корни используются в качестве узлов в интерполяции алгебраическими многочленами. Для интервала  $[a, b]$  можно построить  $n + 1$  точку по следующей формуле:

$$x_{n-i} = \frac{a+b}{2} + \frac{b-a}{2} \cdot \cos\left(\frac{2 \cdot i + 1}{2 \cdot n} \pi\right), i = \overline{0, n}.$$



## МНОГОЧЛЕН ТЕЙЛОРА

Если функция  $f(x)$  бесконечно дифференцируема в некоторой окрестности точки  $a$ , тогда для нее можно выписать *ряд Тейлора* – разложение функции в бесконечную сумму степенных функций:

$$\sum_{k=0}^{\infty} \frac{f^{(k)}(a)}{k!} \cdot (x - a)^k,$$

который при  $a = 0$  становится *рядом Маклорена*.

Эта формула Тейлора описывает поведение функции в окрестности точки  $a$  и используется при доказательстве теорем в дифференциальном исчислении. Ряды Тейлора применяются при аппроксимации функции многочленами, в частности, линеаризация уравнений происходит путем разложения в ряд Тейлора и отбрасывания всех членов выше первого порядка.

В теории приближений важную роль играет *многочлен Тейлора*  $T_n(x)$  степени  $n$  для приближения функции  $f(x)$  в окрестности точки  $a$ , представляющий собой сумму первых  $n$  членов ряда Тейлора:

$$T_n(x) = \sum_{k=0}^n \frac{f^{(k)}(a)}{k!} \cdot (x - a)^k.$$

По теореме Тейлора функцию  $f(x)$  можно представить в виде суммы многочлена Тейлора  $T_n(x)$  и некоторого остаточного члена  $R_n(x)$ :

$$f(x) = T_n(x) + R_n(x).$$

Если  $f(x)$  дифференцируема  $n + 1$  раз, тогда для нее остаточный член при  $b \in (a, x)$  можно выписать в форме Лагранжа

$$R_n(x) = \frac{f^{(n+1)}(b)}{(n+1)!} \cdot (x - a)^{n+1},$$

а его величину можно оценить по следующей формуле (при условии, что  $|f^{(n+1)}(x)| \leq M$ ):

$$|R_n(x)| \leq M \cdot \frac{|x - a|^{n+1}}{(n+1)!}.$$

## МНОГОЧЛЕНЫ БЕРНШТЕЙНА

Многочлены Бернштейна (многочлены в форме Бернштейна) – это алгебраические полиномы, представляющие собой линейную комбинацию базисных многочленов Бернштейна  $b_{k,n}(x)$  и коэффициентов Бернштейна  $\beta_k$ :

$$B_n(x) = \sum_{k=0}^n \beta_k \cdot b_{k,n}(x),$$

где  $b_{k,n}(x) = C_n^k \cdot x^k \cdot (1-x)^{n-k}$  – базисный многочлен Бернштейна.

Легко убедиться, что сумма всех базисных полиномов Бернштейна в любой точке равна 1, а начальные полиномы имеют следующий вид:

$$\begin{aligned} b_{0,0}(x) &= 1, \\ b_{0,1}(x) &= 1-x, \\ b_{1,1}(x) &= x, \\ b_{0,2}(x) &= (1-x)^2, \\ b_{1,2}(x) &= 2x(1-x)^2, \\ b_{2,2}(x) &= x^2. \end{aligned}$$

Первая производная полинома  $b_{k,n}(x)$  рассчитывается по следующей формуле:

$$b'_{k,n}(x) = n \cdot b_{k-1,n-1}(x) - n \cdot b_{k,n-1}(x).$$

Полиномы Бернштейна были описаны еще в начале XX века российским математиком С. Н. Бернштейном, а спустя полвека они стали использоваться при построении кривых Безье.

Для повышения устойчивости при расчете  $B_n(x)$  французским математиком Полем де Фаже де Кастельжо был предложен рекурсивный метод (алгоритм де Кастельжо):

$$B_n(x) = \beta_0^{(n)}(x),$$

где  $\beta_i^{(j)}(x) = \beta_i^{(j-1)}(x) \cdot (1-x) + \beta_{i+1}^{(j-1)}(x) \cdot x$ ,  $i = \overline{0, n-j}$ ,  $j = \overline{1, n}$ , причем  $\beta_i^{(0)}(x) = \beta_i$ .

## ПОНЯТИЯ СПЛАЙНОВ

Кривые и поверхности, встречающиеся в практических задачах, часто имеют весьма сложную форму, поэтому обычно их аппроксимируют набором достаточно простых гладких фрагментов, описываемых при помощи элементарных функций одной или двух переменных.

Вполне естественно потребовать, чтобы данные функции имели схожую природу. Классическим примером являются фрагменты в виде алгебраических многочленов одинаковой степени, коэффициенты которых подобраны так, чтобы обеспечить непрерывность первой производной на стыке фрагментов.

В приложении приведены результаты построения некоторых из рассматриваемых сплайнов.

### *ЗАДАЧИ СПЛАЙНОВ*

Сплайны могут быть использованы при решении следующих задач вычислительной математики:

1. *Задача приближенного представления функции* (заменена исходной сложной функции более простой, но близкой по значениям);
2. *Задача приближенного восстановления функции* (восстановление исходной функции с некоторой точностью на данном интервале);
3. *Задача сглаживания функции* (построение достаточно гладкой функции, которая в контрольных точках незначительно отличается от значений исходной функции);
4. *Задача приближенного вычисления по функции функционалов и операторов* (поиск значения интеграла от функции в заданной системе точек или поиск производной от заданной функции).

Данная работа посвящена построению часто используемых сплайнов для задач восстановления и сглаживания.

## ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

*Сплайн* – это непрерывная кусочно-заданная функция, каждый фрагмент (сегмент) которой является достаточно простой функцией и строится по контрольным точкам. В большинстве случаев сплайн представляет собой гладкую функцию, а в качестве фрагментов используются алгебраические полиномы.

*Контрольные (опорные) точки* – это точки, в которых известно (точно или приближенно) значение исходной функции (функции, для которой строится сплайн). Контрольные точки используются для задания фрагментов сплайна.

*Контрольная ломаная* – это ломаная (фактически, линейный сплайн), построенная по контрольным точкам.

*Степень сплайна* – максимальная из степеней использованных полиномов. *Гладкость сплайна* – максимальный порядок непрерывной производной. *Дефект сплайна* – разность между степенью сплайна и его гладкостью. Исходя из приведенных определений, непрерывная ломаная есть сплайн степени 1 и дефекта 1, поскольку производная ломаной – это кусочно-постоянная функция (полином степени ноль).

Поскольку сплайн есть совокупность функций, каждая из которых задана на своем подмножестве значений аргумента, то сплайн  $S(x)$  в общем случае можно записать следующим образом:

$$S(x) = \sum_{i=0}^n \alpha_i \cdot \psi_i(x) + R(x),$$

где:

$\alpha_i$  – коэффициенты, зависящие от значений исходной функции  $f(x)$  в контрольных точках  $x_i$ ,

$\psi_i(x)$  – базисные функции сплайна,

$R(x)$  – некоторые регулирующие добавки.

## КЛАССИФИКАЦИЯ СПЛАЙНОВ

Сплайны можно классифицировать по множеству признаков, но обычно выделяют следующие классы:

- по назначению;
- по особенностям фрагментов;
- по виду краевых условий;
- по мерности пространства;
- по локальности.

Также у сплайнов могут быть некоторые дополнительные ограничения, нередко вытекающие из физики процессов (например, требование равенства моментов или площадей). Такие условия могут как упростить построение сплайна, так и существенно его затруднить.

### ПО МЕРНОСТИ ПРОСТРАНСТВА

По мерности пространства сплайны разделяются на *одномерные*, зависящие только от одной переменной, и на *двумерные*, зависящие уже от двух переменных. В первом случае сплайн является кривой, а во втором – поверхностью.

Для построения двумерных сплайнов обычно используется область определения в виде прямоугольной сетки, узлы которой определяются контрольными точками. В этом случае построение двумерного сплайна (называемого *бисплайном*) можно выполнить через произведение одномерных сплайнов по каждой из переменных:

$$S(x, y) = \sum_{i=0}^{n_x} \sum_{j=0}^{n_y} f(x_i, y_j) \cdot \psi_i^x(x) \cdot \psi_j^y(y).$$

## ПО НАЗНАЧЕНИЮ

*Интерполяционный сплайн* – это сплайн, обязательно проходящий через все контрольные точки. Обычно такие сплайны используются для приближенного вычисления значений исходной функции при значениях аргумента, отличных от узлов интерполирования. Если же значения аргумента выходят за диапазон исходной области определения, тогда сплайн становится *экстраполяционным*. На рис.3 приведен пример линейного и кубического интерполяционных сплайнов.



Рис.3. Пример интерполяционных сплайнов.

*Сглаживающий сплайн* – это сплайн, проходящий достаточно близко к контрольным точкам. Такие сплайны используют, например, для описания зависимостей физических экспериментов с известной погрешностью измерений или для получения гладкой функции из осциллирующей (сглаживание колебаний). На рис.4 приведен пример двух сглаживающих сплайнов: построенного по контрольным точкам и сглаживающего осцилляции.

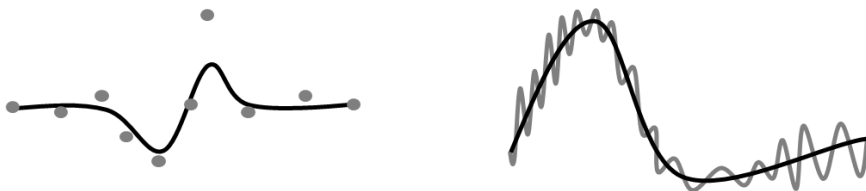


Рис.4. Пример сглаживающих сплайнов.

*Корреляционный сплайн* – это сплайн, проходящий через корреляционное множество точек и отображающий его генеральную зависимость (тренд, тенденцию, регрессию). Такие сплайны используют в качестве нелинейных графиков регрессии, простейшими из которых можно считать описание зависимости ступенчатой и кусочно-линейной функцией. На рис.5 приведен пример линейного и квадратичного корреляционных сплайнов.

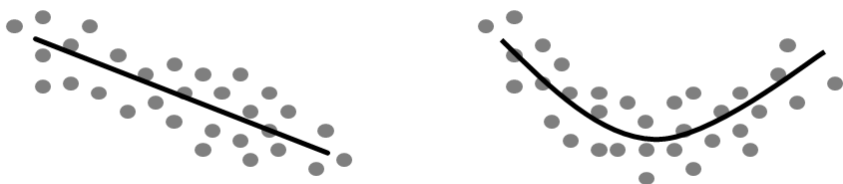


Рис.5. Пример корреляционных сплайнов.

## ПО ЛОКАЛЬНОСТИ

*Глобальные сплайны* – это сплайны, у которых базисные функции в общем случае ненулевые на всей области определения сплайна. Это приводит к тому, что для расчета сплайна в конкретной точке необходимо учитывать все базисные функции, что приводит к быстрому росту вычислительных ресурсов при увеличении числа контрольных точек. Как следствие, при добавлении новой контрольной точки или изменении существующей сплайн полностью пересчитывается.

*Локальные сплайны* – это сплайны, у которых базисные функции ненулевые лишь на части области определения сплайна, т.е. являются финитными. Такой способ задания приводит к тому, что для расчета сплайна в конкретной точке необходимо учитывать лишь соседние базисные функции, а при добавлении новой контрольной точки или изменении существующей сплайн перестраивается лишь частично.

## ПО ТИПУ КРАЕВЫХ УСЛОВИЙ

Поскольку на границах области определения  $[a, b]$  функции  $f(x)$  не всегда можно однозначно построить сплайн  $S(x)$ , то в этих точках обычно накладываются дополнительные (краевые) условия. Также в зависимости от цели построения на сплайн могут быть наложены дополнительные ограничения.

Всего существует 4 типа классических краевых условий и ряд неклассических.

Краевые условия *первого рода* задают значения первой производной на краях сплайна:

$$S'(a) = f'(a),$$

$$S'(b) = f'(b).$$

Краевые условия *второго рода* задают значения второй производной на краях сплайна:

$$S''(a) = f''(a),$$

$$S''(b) = f''(b).$$

Краевые условия *третьего рода* используются для состыковки крайних фрагментов замкнутого или периодического сплайна с периодом  $T = b - a$ , такие краевые условия называются *периодическими*:

$$S'(a) = S'(b),$$

$$S''(a) = S''(b).$$

Краевые условия *четвертого рода* используются, когда на краях сплайна неизвестны ни первая, ни вторая производные. В этом случае стыкуются (приравниваются) соседние пары крайних фрагментов (первого со вторым и последнего с предпоследним) по третьим производным:

$$S'''(a + h - 0) = S'''(a + h + 0).$$

$$S'''(b - h - 0) = S'''(b - h + 0).$$



Необходимость первых и вторых условий обычно вызвана недостаточностью данных для однозначного построения крайних фрагментов сплайна, которых не с чем состыковывать. Краевые условия с нулевой производной часто называют *естественными*, для вторых краевых условий это эквивалентно свободности концов изгибаемой линейки.

К неклассическим краевым условиям относятся, например, изменение на паре соседних фрагментах сплайна его третьей производной по линейному закону.

### *ПО ШИРИНЕ ФРАГМЕНТОВ*

Обычно ширина фрагментов различна (расстояние между контрольными точками области определения неодинаково), но в ряде случаев ширина может оказаться постоянной, что позволяет оптимизировать алгоритм построения сплайна.

Вообще говоря, существуют сплайны и с нулевой шириной фрагментов, что приводит к кратности узлов и возможности приближать сплайны с неразрывными фрагментами разрывных функций.

Также у сплайнов крайние фрагменты могут быть бесконечной ширины, что позволяет естественно задать краевые условия.

### *ПО КОЛИЧЕСТВУ ФРАГМЕНТОВ*

Очевидно, что минимальное количество фрагментов равно одному, а максимально должно быть каким-то разумным числом. Но, в принципе, можно строить сплайны и с бесконечным числом фрагментов – представителями этих сплайнов являются *кардинальные сплайны*.

В интерполяционном полиноме Лагранжа всегда используется только один фрагмент, но при кусочной аппроксимации количество фрагментов зависит от количества контрольных точек.

При использовании сглаживающих сплайнов количество фрагментов обычно можно варьировать, но, например, кривая Безье всегда содержит только один фрагмент.

## ПО ВИДУ ФРАГМЕНТОВ

Сплайн представляет собой кусочно-заданную функцию, каждый фрагмент которой является некоторой функцией одного и того же вида. Соответственно, в зависимости от типа фрагмента можно выделить и соответствующие классы сплайнов.

*Алгебраические сплайны* – это сплайны, чьи фрагменты являются алгебраическими полиномами. Обычно используются полиномы не выше третьей степени, поскольку увеличение степени полинома приводит к быстрому росту вычислительных ресурсов и осцилляций. Достоинствами таких сплайнов являются простота построения и анализа, что и привело к их высокой распространенности. К недостаткам можно отнести то, что относительно мало реальных физических процессов описывается данными зависимостями.

*Трансцендентные сплайны* – это сплайны, чьи фрагменты являются трансцендентными полиномами. К таким сплайнам относятся экспоненциальные, логарифмические, тригонометрические и т.п. Данные сплайны намного лучше описывают реальные процессы по сравнению с алгебраическими, в частности, экспоненциальные сплайны хорошо описывают многие физические процессы в динамических системах. Недостатком таких сплайнов является трудоемкость расчетов. Очень часто для интерполяции и решения дифференциальных уравнений используются тригонометрические многочлены, являющиеся периодическими и бесконечно дифференцируемыми функциями вида:

$$T(x) = c + \sum_{k=1}^n (a_k \cdot \cos(k \cdot x) + b_k \cdot \sin(k \cdot x)) = \sum_{k=-n}^n (c_k \cdot e^{i \cdot k \cdot x}).$$

К *специальным сплайнам* относятся такие специфические сплайны, у которых фрагменты могут быть, например, рациональными функциями или гамма-функциями. К таким сплайнам относится и *монослайн*, используемый в задачах поиска наилучших квадратурных формул для дифференцируемых функций, который является суммой степенной функции и алгебраического полинома:

$$M(x) = \frac{x^m}{m!} + P_{m-1}(x).$$

## ПО СОСТЫКОВКЕ ФРАГМЕНТОВ

В общем случае фрагменты должны состыковываться гладко, это требование достигается требованием непрерывности первой (а иногда и второй) производной:

$$S'(x - 0) = S'(x + 0) \quad \forall x \in [a, b],$$

очевидно, что это условие для линейных сплайнов невыполнимо.

Понятие дефекта сплайна напрямую связано с числом непрерывных производных у самого фрагмента, и числом производных, непрерывность которых гарантирована в узлах. Для большинства задач достаточно непрерывности первой и второй производных, поскольку влияние старших производных несущественно, при том, что их учет приводит к значительным затруднениям при построении сплайна.

Условие на первую производную в точках стыка фрагментов может реализовываться по-разному. При эрмитовой интерполяции значение производной уже дано или же его можно примерно рассчитать из требования равенства производных слева и справа на границах рассматриваемого фрагмента. В другом случае значения первой производной выбирается так, чтобы обеспечилась непрерывность второй.

Как бы то ни было, обязательно должно выполняться условие непрерывности сплайна, в том числе и в местах стыковки фрагментов:

$$S(x - 0) = S(x + 0) \quad \forall x \in [a, b].$$

## ПО ПАРАМЕТРИЗАЦИИ

*Параметрические сплайны* – это сплайны, координаты которых выражаются через некоторый параметр  $t$ , что позволяет, в отличие от обычных (непараметрических) сплайнов, строить функции, у которых одному значению аргумента может соответствовать несколько значений функции.

Обычно параметризацию проводят по хордам, соответствующим участкам контрольной ломаной. В этом случае сплайн параметризуется по длине фрагментов функции.

## ИНТЕРПОЛЯЦИОННЫЙ МНОГОЧЛЕН

*Интерполяция (интерполирование)* – это нахождения промежуточных значений некоторой величины по имеющемуся дискретному набору известных значений. В общем случае задача интерполяции заключается в необходимости построения сплайн-функции определенного вида, проходящей через контрольные точки (*узлы интерполяции*), образующих *интерполяционную сетку*. Вид получаемого сплайна зависит как от используемых базисных функций, так и, например, от ограничений на его производные.

*Интерполяция алгебраическими многочленами* функции  $f(x)$  на интервале  $[a, b]$  – это задача построение многочлена  $S^m(x)$ , принимающего в узлах интерполяции  $\{a = x_0 < x_1 < \dots < x_n = b, n \geq m\}$  значения  $\{f_i = f(x_i), i = 0, n\}$ :

$$S^m(x) = \sum_{i=0}^n f_i \cdot \psi_i^m(x),$$

где  $\psi_i^m(x)$  – это базисные функции, являющиеся алгебраическими полиномами степени  $m$ . Соответственно, условие интерполяции выглядит следующим образом:

$$S^m(x_i) = f_i.$$

При *эрмитовой интерполяции* в узлах  $x_i$  помимо значений  $f(x_i)$  задаются еще и значения  $k$  первых производных  $f^{(k)}(x_i)$ :

$$\begin{array}{ccccccc} f^{(1)}(x_0), & f^{(1)}(x_1), & \dots & f^{(1)}(x_n), \\ f^{(2)}(x_0), & f^{(2)}(x_1), & \dots & f^{(2)}(x_n), \\ & \dots & & \\ f^{(k)}(x_0), & f^{(k)}(x_1), & \dots & f^{(k)}(x_n). \end{array}$$

В результате эрмитова интерполяция строит многочлен, значения которого в контрольных точках совпадают со значениями исходной функции (как и в предыдущем случае), а производные многочлена совпадают со значениями производных функции вплоть до порядка  $k$ .

## ИНТЕРПОЛЯЦИОННЫЕ ФОРМУЛЫ НЬЮТОНА

Если узлы интерполяции расположены равномерно с шагом  $h$ , т.е.

$$x_i = x_0 + i \cdot h, \quad i = \overline{0, n}$$

тогда интерполяционный многочлен можно записать в форме Ньютона. Достоинством ньютоновской интерполяции является то, что добавление нового узла приводит лишь к добавлению нового слагаемого к построенному ранее полиному, однако, изменение значения исходной функции хотя бы в одном узле приводит к необходимости пересчета построенного полинома.

В случае единичного расстояния ( $h = 1$ ) между точками интерполяции справедлива *короткая форма интерполяционной формулы Ньютона*:

$$S_n(x) = \sum_{i=0}^n C_x^i \sum_{j=0}^i (-1)^{i-j} C_i^j f_j.$$

Если обозначить за  $f_i^{(j)}$  конченную разность порядка  $j$  в точке  $x_i$  тогда можно записать первую и вторую интерполяционные формулы Ньютона, которые фактически являются разложением в ряд Тейлора функции  $f(x)$  в окрестности точек  $x_0$  и  $x_n$  соответственно.

*Прямая (первая) интерполяционная формула Ньютона*, применяемая для интерполирования "вперед" (когда точка  $u$  находится в начале области определения):

$$S_n(x) = f_0 + r \cdot f_0^{(1)} + \frac{r \cdot (r-1)}{2!} \cdot f_0^{(2)} + \dots + \frac{r \cdot (r-1) \cdot \dots \cdot (r-n+1)}{n!} \cdot f_0^{(n)},$$

где  $r = (x - x_0)/h$ .

*Обратная (вторая) интерполяционная формула Ньютона*, применяемая для интерполирования "назад" (когда точка  $u$  находится в конце области определения):

$$S_n(x) = f_n + s \cdot f_{n-1}^{(1)} + \frac{s \cdot (s+1)}{2!} \cdot f_{n-2}^{(2)} + \dots + \frac{s \cdot (s+1) \cdot \dots \cdot (s+n-1)}{n!} \cdot f_0^{(n)},$$

где  $s = (x - x_n)/h$ .

## О ЕСТЕСТВЕННЫХ КУБИЧЕСКИХ СПЛАЙНАХ

Как уже упоминалось, изначально сплайн описывал поведение гибкой тонкой линейки, закрепленной в определенных точках с помощью свинцовых грузиков. В этом случае отклонение  $y(x)$  по оси  $OY$  определяется уравнением Эйлера для момента изгиба  $M(x)$  и при малых отклонений ( $y' \ll 1$ ) принимает вид:

$$y'' \approx M(x) / \gamma.$$

где  $\gamma$  – это произведение модуля Юнга на момент инерции.

Если рассматривать грузики как простые подпорки, тогда момент изгиба между ними изменяется линейно:

$$M(x) = \alpha \cdot x + \beta.$$

Соответственно, после двойного интегрирования по  $x$  получается кубическая функция, описывающая профиль линейки:

$$y(x) = c_0 + c_1 \cdot x + c_2 \cdot x^2 + c_3 \cdot x^3,$$

таким образом, форма сплайна задается алгебраическим полиномом третьего порядка, расчет коэффициентов  $c_i$  которого зависит от способа построения сплайна.

Естественные краевые условия  $y''(a) = y''(b) = 0$  в этом случае означают свободу (отсутствие изгибов) концов искривляемой линейки, следовательно, начало и конец описываются линейными функциями. Профиль самой линейки в этом случае принимает форму, при которой потенциальная энергия минимальна, в линейном приближении это выражается следующим соотношением:

$$\int_a^b [S''(x)]^2 dx \leq \int_a^b [f''(x)]^2 dx,$$

причем равенство интегралов выполняется лишь когда  $S(x) = f(x)$ .

Кубические интерполяционные сплайны являются многочленами наименьшей степени, имеющие непрерывные вторые производные. Еще одним достоинством таких сплайнов является простота алгоритмов их построения и реализации, причем влияние ошибок округления при вычислениях оказывается незначительным.

## ФЕНОМЕН РУНГЕ

*Феномен Рунге* – это эффект нежелательных осцилляций (колебаний), возникающий при интерполяции полиномами высоких степеней. Это явление было открыто немецким математиком Карлом Рунге при изучении ошибок полиномиальной интерполяции для приближения некоторых функций.

Данный эффект зависит не только от свойств самой интерполируемой функции, но и от правильного выбора последовательности узлов интерполяции (согласно аппроксимационной теореме Вейерштрасса, для любой непрерывной на отрезке функции можно подобрать последовательность полиномов, равномерно к ней сходящихся). Часто в качестве узлов интерполяции берут узлы Чебышева, погрешность интерполяции по которым равномерно убывает для любой абсолютно непрерывной функции.

Например, для функции  $f(x) = |x|$  при использовании равномерной интерполяционной сетки возникает эффект быстрого роста колебаний сплайн-функции (рис.6). Использование же узлов Чебышева помогает избежать нежелательных колебаний (рис.7).



Рис.6. Рост осцилляций на равномерной сетке.



Рис.7. Уменьшение осцилляций на сетке Чебышева.

## ПАРАМЕТРИЗАЦИЯ СПЛАЙНА

Стандартные алгоритмы построения сплайнов позволяют по дискретному набору точек  $x_i$  строить кривые вида  $y = y(x)$ . Однако такой способ представления не работает в ситуациях, когда одному значению аргумента соответствует несколько значений функции, что характерно, например, для замкнутых и самопересекающихся кривых. Такие ситуации типичны, например, в машиностроении, поскольку кривые инженерных объектов могут иметь вертикальные касательные.

В этом случае сплайн строят как параметрическую функцию.

### *ФУНКЦИИ, ЗАДАННЫЕ ПАРАМЕТРИЧЕСКИ*

У функции, заданной параметрически, связь между аргументом и значением может быть записана через дополнительную переменную:

$$y = F(x): \begin{cases} x = x(t) \\ y = y(t) \end{cases}.$$

Параметрическое уравнение данной кривой в векторной форме имеет вид:

$$\vec{r}(x, y) = \vec{r}(t) = x(t) \cdot \vec{e}_1 + y(t) \cdot \vec{e}_2,$$

где  $\vec{r}(x, y)$  – радиус-вектор точки  $(x, y)$ , а  $\vec{e}_{1,2}$  – координатные орты.

Например, для окружности  $x^2 + y^2 = r^2$  параметрическое представление имеет следующий вид:

$$\begin{cases} x = r \cdot \cos(t) \\ y = r \cdot \sin(t) \end{cases}, t \in [0, 2\pi].$$

Если у функций  $x(t)$  и  $y(t)$  существуют производные, а у функции  $x(t)$  существует еще и обратная, тогда явное представление функции  $y(x)$  выражается через параметрическое как  $y(x(t)) = F(x)$ . В этом случае производная функции  $y(x)$  считается следующим образом:

$$y'_x(x) = \frac{y'_t(t)}{x'_t(t)} \Big|_{t=x^{-1}(x)}.$$



## ПОСТРОЕНИЕ ПАРАМЕТРИЧЕСКОГО СПЛАЙНА

Параметризация может выполняться различными способами, но обычно применяется параметризация по длине траектории: кривая, которую можно рассматривать как траекторию непрерывного движения точки, задается двумя уравнениями  $x = x(t)$  и  $y = y(t)$ , зависящими от некоторого общего параметра  $t \in [t_0, t_1]$ .

Для этого по контрольным точкам  $(x_i, y_i)$  строится контрольная ломаная (задающая ориентацию параметризованной кривой), а в качестве параметра  $t$  берется ее длина:

$$t_{i+1} = t_i + d_i,$$

где  $d_i = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$  – расстояние между точками  $(x_i, y_i)$  и  $(x_{i+1}, y_{i+1})$ , причем начало пути  $t_0 = 0$ , пример на рис.8.

Далее по наборам значений  $(t_i, x_i)$  и  $(t_i, y_i)$  строятся соответствующие одномерные сплайны  $S_x(t)$  и  $S_y(t)$  нужного вида. Результирующий сплайн  $S(x, y)$  может быть записан в векторном виде через орты  $\vec{e}_x$  и  $\vec{e}_y$ :

$$\vec{S}(x, y) = \vec{S}(t) = S_x(t) \cdot \vec{e}_x + S_y(t) \cdot \vec{e}_y.$$

Для построения графика сплайна нужно для каждого  $t \in [t_0, t_1]$  получить значения координаты  $(x, y)$ , подставляя  $t$  в  $S_x(t)$  и  $S_y(t)$ .

К сожалению, такие объекты, как эллипсы и гиперболы не могут быть представлены параметрически с использованием многочленов.

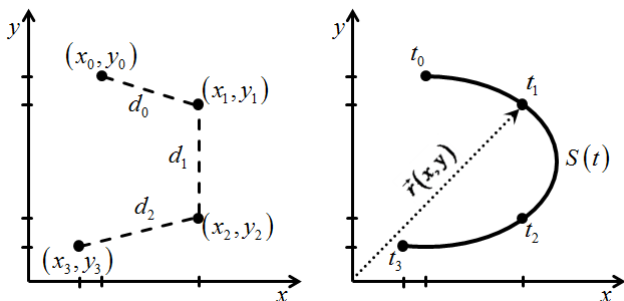


Рис.8. Пример параметризации для четырех контрольных точек.

## ПРИМЕР ПАРАМЕТРИЗАЦИИ ДЛЯ ЛИНЕЙНОГО СПЛАЙНА

Параметризация для линейной функции  $y(x) = \alpha + \beta \cdot x$  на интервале  $[a, b]$  выглядит следующим образом при  $t \in [0, 1]$ :

$$x(t) = a \cdot (1 - t) + b \cdot t,$$

$$y(t) = y(a) \cdot (1 - t) + y(b) \cdot t.$$

Рассмотрим пример построения линейного сплайна  $S(x)$ , состоящего из семи фрагментов  $S_i(x)$ , для набора из восьми контрольных точек  $P_i = (x_i, y_i)$ , показанных на рис.9. Особенность данного набора в том, что -координаты некоторых точек совпадают.

При попытке построить стандартный линейный полином на интервалах  $[x_1, x_2]$  и  $[x_5, x_6]$  происходит сбой, поскольку длина этих интервалов равна нулю, и производная уходит в бесконечность.

При параметризации сплайн  $S(x)$  на каждом сегменте  $[P_i, P_{i+1}]$  можно представить в следующем виде:

$$S_i(t) = P_i \cdot (1 - t) + P_{i+1} \cdot t,$$

где  $t = \frac{t - t_i}{t_{i+1} - t_i}$ .

Таким образом, значения координат  $(x(t), y(t))$  сплайна  $S_i(t)$  на интервале  $[x_i, x_{i+1}]$  рассчитываются по следующим формулам:

$$x(t) = x_i \cdot (1 - t) + x_{i+1} \cdot t,$$

$$y(t) = y_i \cdot (1 - t) + y_{i+1} \cdot t.$$

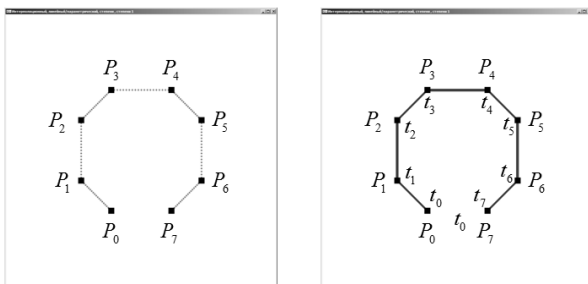


Рис.9. Пример параметризации для набора из 8 контрольных точек.

# ЛАГРАНЖЕВЫ СПЛАЙНЫ

При построении глобального сплайна ( $n = m$ ) получается система из  $n + 1$  уравнений

$$\begin{cases} c_0 + c_1 \cdot x_0 + c_2 \cdot x_0^2 + c_3 \cdot x_0^3 + \dots + c_n \cdot x_0^n = f_0 \\ c_0 + c_1 \cdot x_1 + c_2 \cdot x_1^2 + c_3 \cdot x_1^3 + \dots + c_n \cdot x_1^n = f_1 \\ \dots \\ c_0 + c_1 \cdot x_n + c_2 \cdot x_n^2 + c_3 \cdot x_n^3 + \dots + c_n \cdot x_n^n = f_n \end{cases},$$

которая имеет единственное решение при условии, что узлы интерполяции не равны друг другу.

В матричном виде данную систему можно записать следующим образом:

$$\begin{bmatrix} 1 & x_0 & x_0^1 & \dots & x_0^n \\ 1 & x_1 & x_1^1 & \dots & x_1^n \\ \dots & & & & \\ 1 & x_n & x_n^1 & \dots & x_n^n \end{bmatrix} \cdot \begin{bmatrix} c_0 \\ c_1 \\ \dots \\ c_n \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ \dots \\ f_n \end{bmatrix},$$

причем определитель матрицы данной такой СЛАУ является определителем Вандермонда:

$$\begin{vmatrix} 1 & x_0 & x_0^1 & \dots & x_0^n \\ 1 & x_1 & x_1^1 & \dots & x_1^n \\ \dots & & & & \\ 1 & x_n & x_n^1 & \dots & x_n^n \end{vmatrix} = \prod_{i=0}^n \prod_{j=i+1}^n (x_i - x_j).$$

Поскольку решение такой системы затруднительно, то французский математик Жозеф Луи Лагранж предложил способ построения интерполяционного полинома без предварительного вычисления коэффициентов  $c_i$ , т. е. без решения системы уравнений – так называемый интерполяционный многочлен в форме Лагранжа.

## ИНТЕРПОЛЯЦИОННЫЙ ПОЛИНОМ ЛАГРАНЖА

*Интерполяционный многочлен Лагранжа (в форме Лагранжа)* – это многочлен минимальной степени, принимающий данные значения  $f_i$  в данном наборе точек  $x_i$ . Часто интерполяционный многочлен Лагранжа называют просто *полиномом Лагранжа*.

Полином Лагранжа  $L^n(x)$  степени  $n$  можно представить в виде линейной комбинации базисных полиномов также степени  $n$ :

$$L^n(x) = \sum_{i=0}^n f_i \cdot l_i^n(x),$$

где  $l_i^n(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j}$  – *множитель Лагранжа*.

Поскольку для базисных полиномов Лагранжа  $l_i^n(x)$  выполняется условие

$$l_i^n(x_j) = \begin{cases} 1, & \text{если } i = j \\ 0 & \text{если } i \neq j \end{cases},$$

то и для полинома  $L^n(x)$  также выполняется условие

$$L^n(x_i) = f_i, \quad i = 0, \dots, n.$$

Однако на практике возникают проблемы с тем, что поиск всех коэффициентов многочлена высокой степени приводит к большим вычислительным затратам и возрастанию погрешности вычислений.

**Замечание.** Полином Лагранжа можно использовать и для интегрирования. Например, если функцию  $f(x)$  аппроксимировать как линейную комбинацию базисных полиномов  $l_i(x)$ , тогда:

$$\int_a^b f(x) dx \approx \sum_{i=0}^n \left( f(x_i) \cdot \int_a^b l_i^n(x) dx \right).$$

## ДВУХТОЧЕЧНАЯ ИНТЕРПОЛЯЦИЯ

Поскольку глобальная интерполяция по Лагранжу затруднена при большом числе контрольных точек, то в таких ситуациях часто применяется *локальная интерполяция*, когда вместо полинома степени  $n$  строится последовательность полиномов меньшей степени.

В данном случае на каждом интервале  $\Omega_i = [x_i, x_{i+1}]$  длиной  $h_i$  расчетной области  $\Omega = \bigcup_{i=0}^{n-1} \Omega_i$  строится локальный алгебраический полином  $P_i^m(x)$  заданной степени  $m \leq n$ , который на границах интервала равен значениям  $f_i$  и  $f_{i+1}$  соответственно (условие интерполяции), а вне границ своего интервала данный полином не существует:

$$P_i^m(x) = \begin{cases} 0, & x \notin [x_i, x_{i+1}] \\ f_i, & x = x_i \\ f_{i+1}, & x = x_{i+1} \\ \sum_{k=0}^m c_{i,k} \cdot x^k, & x \in [x_i, x_{i+1}] \end{cases}$$

Объединение построенных таким образом полиномов  $P_i^m(x)$  и дает требуемый сплайн  $S^m(x)$ :

$$S^m(x) = \bigcup_{i=0}^{n-1} P_i^m(x) = \sum_{i=0}^{n-1} P_i^m(x),$$

при этом условие непрерывности выглядит как  $P_i^m(x_{i+1}) = P_{i+1}^m(x_{i+1})$ , а условие гладкости полученного сплайна заключается в требовании равенства производных слева и справа для каждой внутренней точки.

Соответственно, вычисление сплайна в некоторой точке интервала  $\Omega_i$  заключается в поиске всех коэффициентов на всей области  $\Omega$ , с последующим вычислением значения базисного полинома на  $\Omega_i$ .

Обычно при локальной интерполяции используются линейные ( $m = 1$ ), квадратичные ( $m = 2$ ) или кубические ( $m = 3$ ) полиномы.

## КУСОЧНО-ЛИНЕЙНАЯ

При линейной интерполяции на интервале  $[x_i, x_{i+1}]$  длиной  $h_i$  базисный полином  $P_i^1(x) = S_i(x)$  является функцией вида:

$$S_i(x) = a_i + b_i \cdot x.$$

Для однозначного нахождения обоих коэффициентов данной линейной функции достаточно системы всего лишь из двух уравнений, которые берутся из условия интерполяции:

$$\begin{cases} S_i(x_i) = f_i \\ S_i(x_{i+1}) = f_{i+1} \end{cases}.$$

В результате из системы  $2 \times 2$  элементарно находятся формулы для коэффициентов  $a_i$  и  $b_i$ :

$$a_i = f_i - b_i \cdot x_i,$$

$$b_i = \frac{1}{h_i} \cdot (f_{i+1} - f_i),$$

В форме Тейлора полином  $P_i^1(x)$  можно представить следующем образом:

$$S_i(x) = f_i + \frac{f_{i+1} - f_i}{h_i} \cdot (x - x_i).$$

В ряде же случаев удобно пользоваться другой формой записи, через линейные лагранжевы базисные функции:

$$S_i(x) = \frac{x_{i+1} - x}{h_i} f_i + \frac{x - x_i}{h_i} f_{i+1}.$$

Достоинствами линейного сплайна являются простота и отсутствие осцилляций, к недостаткам же можно отнести разрывность даже первых производных.

## КУСОЧНО-КВАДРАТИЧНАЯ

При квадратичной интерполяции на интервале  $[x_i, x_{i+1}]$  длиной  $h_i$  базисный полином  $P_i^2(x) = S_i(x)$  является функцией вида:

$$S_i(x) = a_i + b_i \cdot x + c_i \cdot x^2.$$

Для однозначного нахождения всех трех коэффициентов данной квадратичной функции требуется система из трех уравнений. Первые два уравнения берутся из условия интерполяции, а третье – из условия гладкости (на границах интервалов производные слева и справа должны быть равны). Соответствующая система уравнений выглядит следующим образом:

$$\begin{cases} S_i(x_i) = f_i \\ S_i(x_{i+1}) = f_{i+1} \\ \left. \frac{dS_i}{dx} \right|_{x=x_i} = \left. \frac{dS_{i-1}}{dx} \right|_{x=x_i} \end{cases}$$

В результате можно получить следующие формулы для коэффициентов  $a_i$ ,  $b_i$  и  $c_i$ :

$$a_i = f_i - \beta_i \cdot x_i + c_i \cdot x_i^2,$$

$$b_i = \beta_i - 2 \cdot c_i \cdot x_i,$$

$$c_i = \frac{1}{h_i^2} \cdot (f_{i+1} - f_i) - \frac{1}{h_i} \cdot \beta_i,$$

где  $\beta_i = b_{i-1} + 2 \cdot c_{i-1} \cdot x_i$  – это "производная слева" в точке  $x_i$ .

Как можно видеть, в точке  $x_0$  посчитать коэффициент  $\beta_0$  невозможно, соответственно, если нет информации о значении производной функции  $f(x)$  в точке  $x_0$ , тогда  $\beta_0$  можно аппроксимировать через правую конечную разность:

$$\beta_0 = \frac{f_1 - f_0}{h_0}.$$

## КУСОЧНО-КУБИЧЕСКАЯ

При выводе формул кубический сплайн часто расписывают в многочлен Тейлора третьего порядка в окрестности точки  $x_i$ :

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

на интервале  $[x_{i-1}, x_i]$  длиной  $h_i$ ,  $i \in [1, n]$ .

Условия интерполяции, непрерывности и гладкости можно представить следующим образом:

$$\begin{aligned} S_i(x_i) &= f_i, \\ S_i(x_{i-1}) &= f_{i-1}, \\ S_i(x_{i-1}) &= S_{i-1}(x_{i-1}), \\ S'_i(x_{i-1}) &= S'_{i-1}(x_{i-1}), \\ S''_i(x_{i-1}) &= S''_{i-1}(x_{i-1}). \end{aligned}$$

Таким образом можно получить лишь  $4n - 2$  уравнения, поскольку на каждом интервале сплайн задается четырьмя коэффициентами, а для начального интервала не хватает информации. При построении естественного сплайна недостающие два уравнения для границ задаются как естественные краевые условия:

$$S''(x_0) = S''(x_n) = 0.$$

Теперь при подстановке и приравнивании коэффициентов можно получить следующие формулы для трех коэффициентов:

$$\begin{aligned} a_i &= f_i, \\ b_i &= \frac{a_i - a_{i-1}}{h_i} + \frac{2 \cdot c_i + c_{i-1}}{3} \cdot h_i, \\ d_i &= \frac{c_i - c_{i-1}}{3 \cdot h_i}, \end{aligned}$$



Для расчета же коэффициента  $c_i$  необходимо решить следующее разностное уравнение второго порядка:

$$c_{i-1} \cdot h_i + 2 \cdot c_i \cdot (h_i + h_{i+1}) + c_{i+1} \cdot h_{i+1} = 3 \cdot \left( \frac{a_{i+1} - a_i}{h_{i+1}} + \frac{a_i - a_{i-1}}{h_i} \right),$$

с краевыми условиями  $c_n = c_0 = 0$ .

Данное уравнение можно представить в виде следующей СЛАУ:

$$\begin{bmatrix} 1 & 0 & & & & & \\ h_1 & \delta_1 & h_2 & & & & \\ & h_2 & \delta_2 & h_3 & & & \\ & & h_3 & \delta_3 & h_4 & & \\ & & & \dots & \dots & \dots & \\ & & & & h_{n-1} & \delta_{n-1} & h_n \\ & & & & & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ \dots \\ c_{n-1} \\ c_n \end{bmatrix} = \begin{bmatrix} 0 \\ \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \dots \\ \varepsilon_{n-1} \\ 0 \end{bmatrix},$$

где:  $\delta_i = 2 \cdot (h_i + h_{i+1})$ ,  $\varepsilon_i = 3 \cdot \left( \frac{f_{i+1} - f_i}{h_{i+1}} - \frac{f_i - f_{i-1}}{h_i} \right)$ ,  $i \in [1, n - 1]$ .

Как можно видеть, матрица СЛАУ для расчета коэффициентов  $c_i$  является трехдиагональной, что и позволяет применить метод прогонки для решения СЛАУ и нахождения коэффициентов  $c_i$ .

При наличии дополнительных условий, например, на вид второй производной, в данную СЛАУ вносятся соответствующие добавки.

Использование кубической аппроксимации позволяет строить кривые с непрерывной не только первой, но и второй производной. При использовании же многочленов более высоких степеней увеличивается сложность расчетов, и возникают нежелательные осцилляции кривых. Эти факторы и определили популярность кубической аппроксимации.

## МНОГОТОЧЕЧНАЯ ИНТЕРПОЛЯЦИЯ

Рассмотренные ранее способы задания фрагментов сплайна выполнялись на каждом интервале  $[x_i, x_{i+1}]$ . Теперь рассмотрим ситуацию построения фрагментов сплайна с использованием базисных функций степени  $m$  на элементах  $\Omega_k = [x_{k \cdot m}, x_{(k+1) \cdot m}]$ , которые содержат не два узла интерполяции, а  $m + 1$ . Очевидно, при таком подходе количество рассматриваемых элементов  $\Omega_k$  будет равно не  $n$ , а  $[n/m]$ , пример для случая  $m = 2$  приведен на (рис.10).

На каждом таком лагранжевом элементе  $\Omega_k$  задаются локальные базисные функции  $\hat{\psi}_i(x)$  в виде полиномов Лагранжа порядка  $m$ , которые в своем узле равны единице, а во всех остальных – нулю (эти базисные для МКЭ функции, как локальные, так и глобальные, подробно рассмотрены в [3]). Поскольку базис на каждом элементе  $\Omega_k$  задается независимо, то гладкость получаемого сплайна на границах лагранжевых элементов не гарантируется.

Получение значения сплайна в точке заключается в поиске нужного интервала  $\Omega_k$  и вычисления линейной комбинации локальных базисных функций  $\hat{\psi}_j(x)$ :

$$S(x) = \sum_{j=0}^m f_{k \cdot m + j} \cdot \hat{\psi}_j(x).$$

Для удобства введем следующую замену для преобразования исходного интервала  $[x_i, x_{i+m}]$  длиной  $h = x_{i+m} - x_i$  в интервал  $[0,1]$  с шаблонными базисными функциями  $\varphi_0(\xi)$ ,  $\varphi_1(\xi)$ , ...,  $\varphi_m(\xi)$ :

$$\xi(x) = \frac{x - x_i}{h}.$$

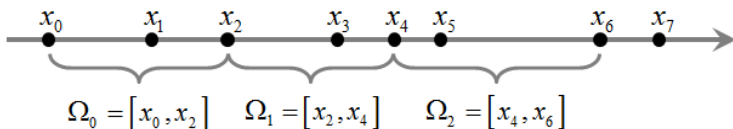


Рис. 10. Пример лагранжевого квадратичного элемента.

## КУСОЧНО-ЛИНЕЙНАЯ

На линейном лагранжевом элементе  $\Omega_k = [x_i, x_{i+1}]$  задаются две локальные линейные базисные функции  $\hat{\psi}_i(\xi(x))$  и  $\hat{\psi}_{i+1}(\xi(x))$ , графики которых приведены на рис.11:

$$\varphi_0(\xi) = \hat{\psi}_i(\xi) = 1 - \xi,$$

$$\varphi_1(\xi) = \hat{\psi}_{i+1}(\xi) = \xi.$$

Соответственно, на данном элементе значение сплайна рассчитывается через линейную комбинацию таких базисных функций:

$$S_k(x) = f_i \cdot \varphi_0(\xi) + f_{i+1} \cdot \varphi_1(\xi).$$

По факту, сплайн, построенный подобным образом, полностью совпадает с двухточечным линейным сплайном.

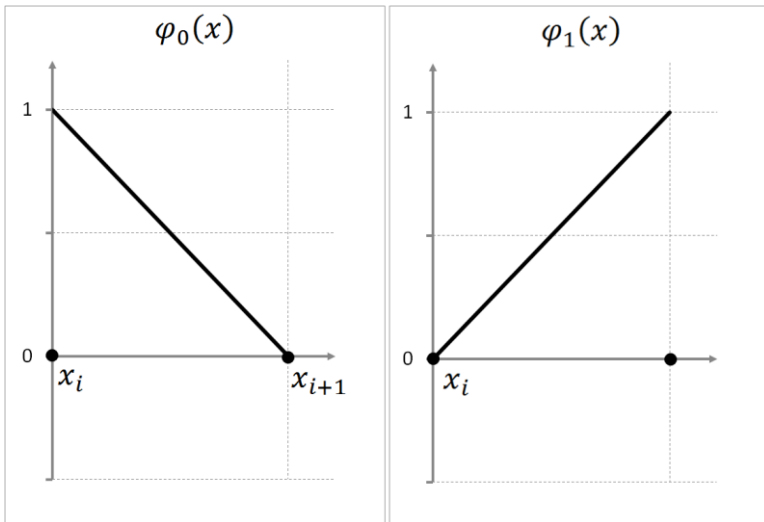


Рис.11. Линейные базисные функции на интервале  $[x_i, x_{i+1}]$ .

Из данных локальных функций можно собрать глобальные базисные функции  $\psi_i(x)$ :

$$\psi_i(x) = \begin{cases} \frac{x - x_{i-1}}{h_{i-1}} = \varphi_1(\xi(x)), & \text{при } x \in [x_{i-1}, x_i] \\ \frac{x_{i+1} - x}{h_i} = \varphi_0(\xi(x)), & \text{при } x \in [x_i, x_{i+1}] \\ 0, & \text{при } x \notin [x_{i-1}, x_{i+1}] \end{cases}$$

На рис.12 приведен пример глобальной базисной функции  $\psi_i(x)$  для части области определения. Как можно видеть, функция  $\psi_i(x)$  является финитной: она равна единице лишь в своем узле, на соседних интервалах  $[x_{i-1}, x_i]$  и  $[x_i, x_{i+1}]$  она убывает до нуля и остается равной нулю на всех остальных интервалах (в остальных узлах).

В случае использования глобальных базисных функций значение сплайна в точке будет рассчитывается через полную линейную комбинацию весов  $f_i$  и глобальных базисных функций  $\psi_i$ :

$$S_k(x) = \sum_{i=0}^n f_i \cdot \psi_i(x).$$

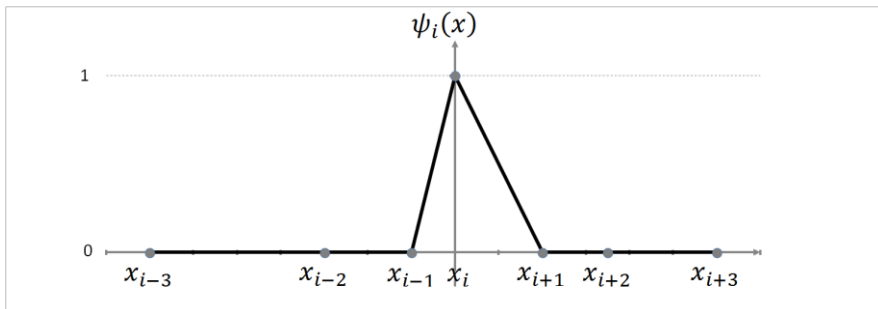


Рис.12. Пример глобальной линейной базисной функции для узла  $x_i$ .

## КУСОЧНО-КВАДРАТИЧНАЯ

На квадратичном лагранжевом элементе  $\Omega_k = [x_i, x_{i+2}]$  задаются три локальные квадратичные базисные функции  $\hat{\psi}_i(\xi(x))$ ,  $\hat{\psi}_{i+1}(\xi(x))$  и  $\hat{\psi}_{i+2}(\xi(x))$ , графики которых приведены на рис.13:

$$\varphi_0(\xi) = \hat{\psi}_i(\xi) = \frac{\xi - \alpha}{-\alpha} \cdot \frac{\xi - 1}{-1},$$

$$\varphi_1(\xi) = \hat{\psi}_{i+1}(\xi) = \frac{\xi}{\alpha} \cdot \frac{\xi - 1}{\alpha - 1},$$

$$\varphi_2(\xi) = \hat{\psi}_{i+2}(\xi) = \frac{\xi}{1} \cdot \frac{\xi - \alpha}{1 - \alpha},$$

где  $\alpha = \frac{x_{i+1} - x_i}{h}$ .

Соответственно, на данном элементе значение сплайна рассчитывается через линейную комбинацию таких базисных функций:

$$S_k(x) = f_i \cdot \varphi_0(\xi) + f_{i+1} \cdot \varphi_1(\xi) + f_{i+2} \cdot \varphi_2(\xi).$$

При желании можно, как и в линейном случае, перейти к глобальным базисным функциям.

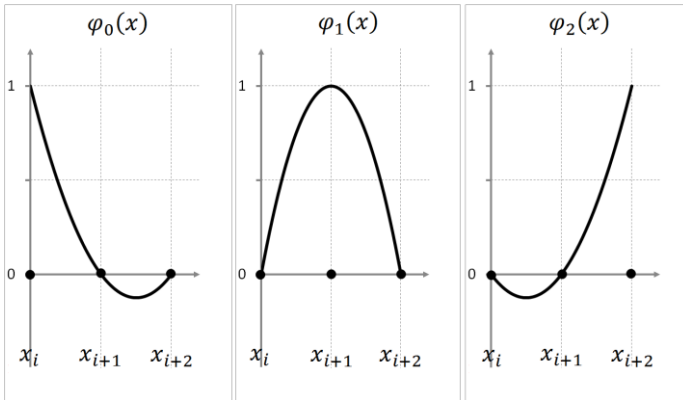


Рис.13. Квадратичные базисные функции на интервале  $[x_i, x_{i+2}]$ .

## КУСОЧНО-КУБИЧЕСКАЯ

На кубическом лагранжевом элементе  $\Omega_k = [x_i, x_{i+3}]$  задаются четыре локальные кубические базисные функции  $\hat{\psi}_i(\xi(x))$ ,  $\hat{\psi}_{i+1}(\xi(x))$ ,  $\hat{\psi}_{i+2}(\xi(x))$  и  $\hat{\psi}_{i+3}(\xi(x))$ , графики которых приведены на рис.14:

$$\varphi_0(\xi) = \hat{\psi}_i(\xi) = \frac{\xi - \beta}{-\beta} \cdot \frac{\xi - \gamma}{-\gamma} \cdot \frac{\xi - 1}{-1},$$

$$\varphi_1(\xi) = \hat{\psi}_{i+1}(\xi) = \frac{\xi}{\beta} \cdot \frac{\xi - \gamma}{\beta - \gamma} \cdot \frac{\xi - 1}{\beta - 1},$$

$$\varphi_2(\xi) = \hat{\psi}_{i+2}(\xi) = \frac{\xi}{\gamma} \cdot \frac{\xi - \beta}{\gamma - \beta} \cdot \frac{\xi - 1}{\gamma - 1},$$

$$\varphi_3(\xi) = \hat{\psi}_{i+3}(\xi) = \frac{\xi}{1} \cdot \frac{\xi - \beta}{1 - \beta} \cdot \frac{\xi - \gamma}{1 - \gamma},$$

где  $\beta = \frac{x_{i+1} - x_i}{h}$ ,  $\gamma = \frac{x_{i+2} - x_i}{h}$ .

Соответственно, на данном элементе значение сплайна рассчитывается через линейную комбинацию таких базисных функций:

$$S_k(x) = f_i \cdot \varphi_0(\xi) + f_{i+1} \cdot \varphi_1(\xi) + f_{i+2} \cdot \varphi_2(\xi) + f_{i+3} \cdot \varphi_3(\xi).$$

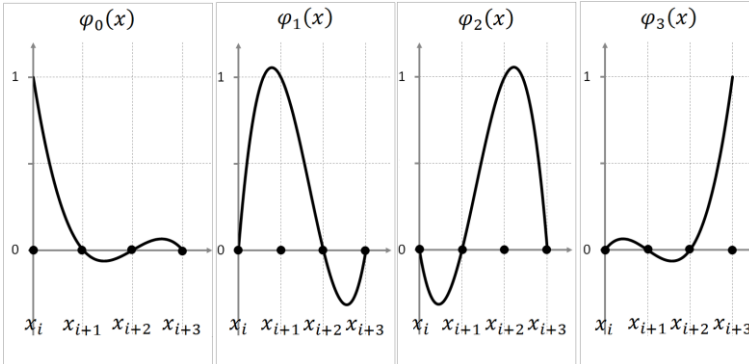


Рис.14. Кубические базисные функции на интервале  $[x_i, x_{i+3}]$ .

## ЭРМИТОВЫ СПЛАЙНЫ

*Эрмитова интерполяция* – это метод интерполяции полиномами с использованием производных функции, названный в честь французского математика Шарля Эрмита. Суть метода заключается в построении полинома, значения которого в точках интерполяции совпадают со значениями исходной функции, как совпадают и значения их первых производных. В общем же случае, эрмитова интерполяция предполагает совпадение производных вплоть до  $n$ -го порядка, но при этом количество известных в контрольных точках производных может оказаться переменным. Очевидно, что построенный таким образом сплайн автоматически получается гладким.

На практике же обычно применяются лишь производные первого порядка. Это связано с тем, что старшие производные редко когда бывают известны, более того, часто даже сами первые производные неизвестны (в этом случае их приходится каким-либо образом аппроксимировать для построения сплайна). С другой стороны, первая производная геометрически является касательной к функции в точке, поэтому в графических редакторах первые производные используются в качестве направляющих при построении гладких кривых, как проиллюстрировано на рис.15.

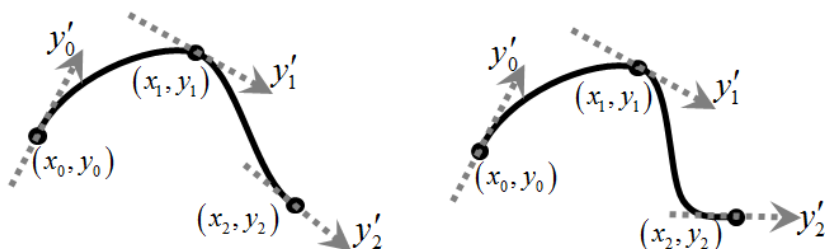


Рис.15. Пример эрмитовой интерполяции.

Таким образом, на *эрмитовом элементе*  $[x_i, x_{i+1}]$  задаются значения  $f_i, f_{i+1}, f'_i$  и  $f'_{i+1}$ , что позволяет однозначно построить кубический полином, который при этом не является многочленом Эрмита. Соответственно, *кубический эрмитов сплайн* – это сплайн, построенный из кубических полиномов с использованием эрмитовой интерполяции:

$$S_i(\xi) = f_i \cdot \varphi_0(\xi) + f'_i \cdot \varphi_1(\xi) + f_{i+1} \cdot \varphi_3(\xi) + f'_{i+1} \cdot \varphi_4(\xi),$$

где  $\varphi_i$  – это эрмитовые кубические базисные функции.

В том случае, когда значения производной исходной функции  $f(x)$  в узлах  $x_i$  неизвестны, возникает необходимость в получении их приближенных значений. Вообще говоря, существуют различные способы задания значений  $f'_i$ , влияющие на свойства получаемого сплайна.

Например, можно построить сплайн с непрерывными вторыми производными путем дифференцирования базисных функций (этот вариант подробно разобран в [3]).

Но в основном же методы расчета производных основаны на использование конечно-разностного подхода. Один из вариантов – это расчет через полусумму левой и правой конечных разностей:

$$f'_i = \frac{1}{2} \cdot \left( \frac{f_{i+1} - f_i}{h_i} + \frac{f_i - f_{i-1}}{h_{i-1}} \right).$$

Использование центральной конечной разности приводит к построению сплайна *Катмалла-Рома*:

$$f'_i = \frac{f_{i+1} - f_{i-1}}{h_i + h_{i-1}},$$

у которого касательная в точке  $x_i$  становится параллельной отрезку  $(P_{i-1}, P_{i+1})$ .

В *кардинальном сплайне* эта формула обобщается путем введения числового параметра  $c \in [0, 1]$ :

$$f'_i = (1 - c) \frac{f_{i+1} - f_{i-1}}{h_i + h_{i-1}}.$$

В данной же работе будет рассмотрен лишь вариант с использованием квадратичного полинома Лагранжа.



## ЭРМИТОВЫ БАЗИСНЫЕ ФУНКЦИИ

Как и в случае лагранжевых базисных функций, введем следующую замену для преобразования исходного интервала  $[x_i, x_{i+1}]$  длиной  $h_i$  в шаблонный  $[0,1]$ :

$$\xi(x) = \frac{x - x_i}{h_i}.$$

Зададим локальные эрмитовы базисные функции  $\varphi_i(\xi)$  (их вид приведен на рис.16) через полиномы Бернштейна  $b_{j,3}$  и глобальные базисные функции  $\psi_k$  следующим образом:

$$\begin{aligned} \varphi_0(\xi) &= b_{0,3} - b_{1,3} = \psi_{2i+0}(\xi) = 1 - 3 \cdot \xi^2 + 2 \cdot \xi^3, \\ \varphi_1(\xi) &= \frac{1}{3} b_{1,3} = \psi_{2i+1}(\xi) = h_i \cdot (\xi - 2 \cdot \xi^2 + \xi^3), \\ \varphi_2(\xi) &= b_{2,3} + b_{3,3} = \psi_{2i+2}(\xi) = 3 \cdot \xi^2 - 2 \cdot \xi^3, \\ \varphi_3(\xi) &= -\frac{1}{3} b_{2,3} = \psi_{2i+3}(\xi) = h_i \cdot (-\xi^2 + \xi^3). \end{aligned}$$

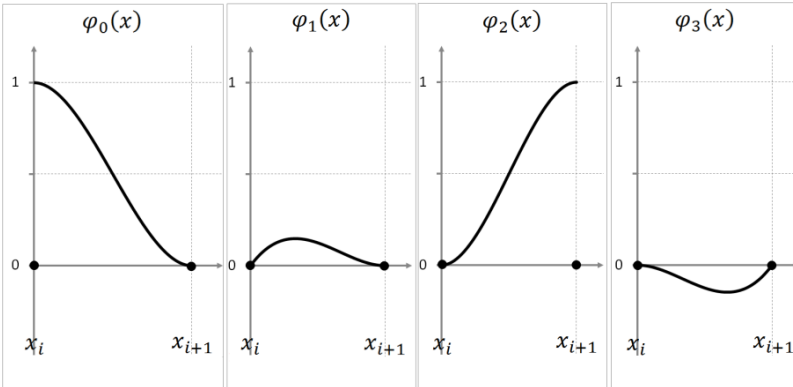


Рис.16. Локальные эрмитовы базисные функции.

Как можно видеть, функции  $\psi_{2i}(\xi)$  и  $\psi_{2i+2}(\xi)$  соответствуют значениям  $f_i$  и  $f_{i+1}$ , а функции  $\psi_{2i+1}(\xi)$  и  $\psi_{2i+3}(\xi)$  – значениям  $f'_i$  и  $f'_{i+1}$ , их графики приведены на рис.17 и рис.18. Видно, что, функция  $\psi_{2i}(x)$  равна единице лишь в своем узле, на соседних интервалах  $[x_{i-1}, x_i]$  и  $[x_i, x_{i+1}]$  она убывает до нуля и остается равной нулю на всех остальных интервалах (в остальных точках интерполяции). В свою очередь, функция  $\psi_{2i+1}(x)$  не равна нулю лишь на интервале  $[x_{i-1}, x_{i+1}]$ .

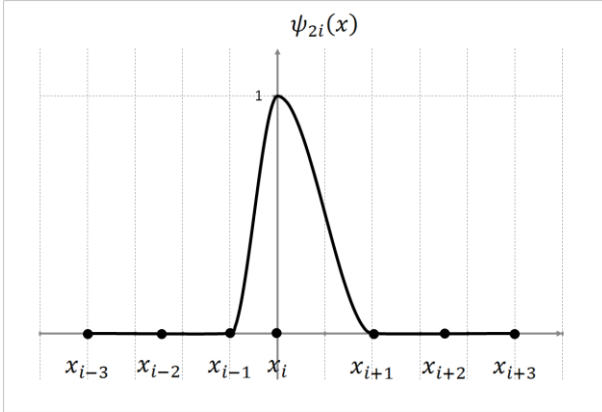


Рис.17. Глобальная эрмитова базисная функция  $\psi_{2i}$ .

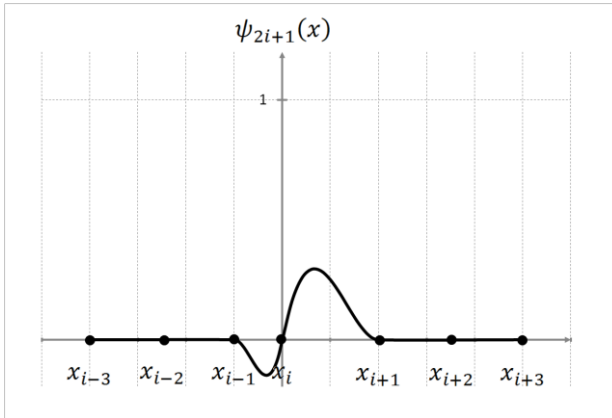


Рис.18. Глобальная эрмитова базисная функция  $\psi_{2i+1}$ .

## КУБИЧЕСКАЯ ИНТЕРПОЛЯЦИЯ

В общем случае, интерполяционный кубический эрмитов сплайн может быть рассчитан по следующей формуле:

$$S(x) = \sum_{i=0}^n f_i \psi_{2i}(x) + \sum_{i=0}^n f'_i \psi_{2i+1}(x),$$

где  $\psi_j(x)$  – это глобальные эрмитовы базисные функции. Очевидно, что расчет значения сплайна в произвольной точке удобно вычислять через локальные базисные функции  $\varphi_i(\xi)$ , как и в случае лагранжевых базисных функций.

Рассмотрим простой и довольно распространенный вариант нахождения значений  $f'_i$  через аппроксимацию квадратичным полиномом Лагранжа по трем точкам.

Для этого на интервале  $[x_{i-1}, x_{i+1}]$  длиной  $(h_{i-1} + h_i)$  строится квадратичный интерполяционный полином Лагранжа  $L_i(x)$  вида:

$$L_i(x) = f_{i-1} \cdot l_i(x) + f_i \cdot c_i(x) + f_{i+1} \cdot r_i(x),$$

причем базисные полиномы Лагранжа  $l_i(x)$ ,  $c_i(x)$  и  $r_i(x)$  имеют вид:

$$l_i(x) = \frac{(x - x_i) \cdot (x - x_{i+1})}{h_{i-1} \cdot (h_{i-1} + h_i)},$$

$$c_i(x) = -\frac{(x - x_{i-1}) \cdot (x - x_{i+1})}{h_{i-1} \cdot h_i},$$

$$r_i(x) = \frac{(x - x_{i-1}) \cdot (x - x_i)}{h_i \cdot (h_{i-1} + h_i)}.$$

В этом случае значения  $f_i'$  вычисляются через производные полинома  $L_i(x)$ :

$$f_i' = \left. \frac{dL_i(x)}{dx} \right|_{x=x_i} = f_{i-1} \cdot l_i'(x) + f_i \cdot c_i'(x) + f_{i+1} \cdot r_i'(x),$$

где:

$$l_i'(x) = \left. \frac{dl(x)}{dx} \right|_{x=x_i} = -\frac{h_i}{h_{i-1} \cdot (h_{i-1} + h_i)},$$

$$c_i'(x) = \left. \frac{dc(x)}{dx} \right|_{x=x_i} = -\frac{h_{i-1} - h_i}{h_{i-1} \cdot h_i},$$

$$r_i'(x) = \left. \frac{dr(x)}{dx} \right|_{x=x_i} = \frac{h_{i-1}}{h_i \cdot (h_{i-1} + h_i)}.$$

Для точек  $x_0$  и  $x_n$  полиномы  $L_0(x)$  и  $L_n(x)$  строятся на интервалах  $[x_0, x_2]$  и  $[x_{n-2}, x_n]$  соответственно, следовательно, значения  $f_0'$  и  $f_n'$  могут быть получены из следующих формул:

$$f_0' = \left. \frac{dL_1(x)}{dx} \right|_{x=x_0} = f_0 \cdot l_1'(x) + f_1 \cdot c_1'(x) + f_2 \cdot r_1'(x),$$

$$f_n' = \left. \frac{dL_{n-1}(x)}{dx} \right|_{x=x_n} = f_{n-2} \cdot l_{n-1}'(x) + f_{n-1} \cdot c_{n-1}'(x) + f_n \cdot r_{n-1}'(x).$$

## КУБИЧЕСКОЕ СГЛАЖИВАНИЕ

Под задачей сглаживания понимается построение достаточно гладкой функции  $S(x)$ , значения которой в точках интерполяции максимально близки к значениям исходной функции. Эту задачу удобно решать на основе метода наименьших квадратов:

$$\sum_{i=0}^m (S(x) - f_i)^2 \rightarrow \min.$$

Разобьем область определения на  $m \ll n$  конечных элементов (пример для  $m = 2$  и  $n = 4$  приведен на рис.19) и выпишем сплайн  $S(x)$  с использованием эрмитовых базисных функций  $\psi_i(x)$  на этих элементах в виде

$$S(x) = \sum_{i=0}^m [\tilde{f}_i \cdot \psi_{2i}(x) + \tilde{f}'_i \cdot \psi_{2i+1}(x)] = \sum_{i=0}^{2m} q_i \cdot \psi_i(x),$$

причем коэффициенты  $q_i$  находятся из условия минимизации функционала

$$\Phi(q) = \sum_{k=0}^n \omega_k \cdot \left( f_k - \sum_{i=0}^{2m} q_i \cdot \psi_i(x_k) \right)^2,$$

где  $\omega_j$  – это заданные веса, определяющие меру близости сплайна  $S(x)$  к значениям функции  $f(x)$  в узлах  $x_i$ .

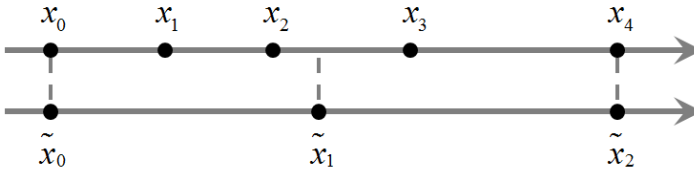


Рис.19. Пример разбиения интервала  $[x_0, x_4]$  на 2 конечных элемента.

В результате подстановки и минимизации, что подробно расписано в [3], можно получить следующую СЛАУ размерности  $2m$ :

$$M \cdot q = b,$$

где компоненты матрицы  $M$  и вектора  $b$  рассчитываются по следующим формулам:

$$m_{i,j} = \sum_{k=1}^n \omega_k \cdot \psi_i(x_k) \cdot \psi_j(x_k),$$

$$f_i = \sum_{k=1}^n \omega_k \cdot f_k \cdot \psi_i(x_k).$$

К сожалению, матрица  $A$  в некоторых случаях может оказаться вырожденной, что означает существование неединственного сплайна с минимальной суммой квадратов отклонений в узлах  $x_k$ . Это может произойти, когда в интервал  $[\tilde{x}_i, \tilde{x}_{i+1}]$  попадает недостаточное количество узлов  $x_k$ . В этом случае можно изменить либо сглаживающую сетку, либо функционал  $\Phi(q)$  путем добавления регуляризующих слагаемых  $\Phi^\alpha(q)$  и  $\Phi^\beta(q)$ :

$$\bar{\Phi}(q) = \sum_{k=0}^n \omega_k \cdot (S(x_k) - f_k)^2 + \Phi^\alpha(q) + \Phi^\beta(q),$$

где:

$$\Phi^\alpha(q) = \int_{\tilde{x}_0}^{\tilde{x}_m} \alpha \cdot \left( \frac{dS(x)}{dx} \right)^2 \cdot dx,$$

$$\Phi^\beta(q) = \int_{\tilde{x}_0}^{\tilde{x}_m} \beta \cdot \left( \frac{d^2S(x)}{dx^2} \right)^2 \cdot dx.$$

Минимизация функционала  $\bar{\Phi}(q)$  при  $\alpha > 0$  и  $\beta > 0$  определяет единственность сплайна  $S(x)$ . При этом увеличение коэффициента  $\alpha$  уменьшает вторую производную сплайна, приближая его к линейной

функции. Аналогично, увеличение коэффициента  $\beta$  уменьшает первую производные сплайна, приближая его к константной функции.

При переходе от  $\Phi(q)$  к  $\bar{\Phi}(q)$  вектор правой части не изменится, в отличие от компонент матрицы СЛАУ, которые перерассчитаются с учетом добавок  $m_{i,j}^\alpha$  и  $m_{i,j}^\beta$ :

$$m_{i,j} = \sum_{k=0}^n \omega_k \cdot \psi_i(x_k) \cdot \psi_j(x_k) + m_{i,j}^\alpha + m_{i,j}^\beta,$$

где:

$$m_{i,j}^\alpha = \int_{\tilde{x}_0}^{\tilde{x}_m} \alpha \cdot \frac{d\psi_i(x)}{dx} \cdot \frac{d\psi_j(x)}{dx} \cdot dx,$$

$$m_{i,j}^\beta = \int_{\tilde{x}_0}^{\tilde{x}_m} \beta \cdot \frac{d^2\psi_i(x)}{dx^2} \cdot \frac{d^2\psi_j(x)}{dx^2} \cdot dx.$$

Расчет коэффициентов  $m_{i,j}$ ,  $m_{i,j}^\alpha$  и  $m_{i,j}^\beta$  удобно организовать через интегрирование локальных эрмитовых базисных функций  $\varphi_i(\xi(x))$  на элементах  $[\tilde{x}_i, \tilde{x}_{i+1}]$  длиной  $h$  (стандартная технология МКЭ):

$$\begin{aligned} \hat{a}_{i,j}^\alpha &= \int_{\tilde{x}_i}^{\tilde{x}_{i+1}} \alpha \cdot \frac{d\varphi_i(x)}{dx} \cdot \frac{d\varphi_j(x)}{dx} \cdot dx = \\ &= \frac{\alpha}{30h} \cdot \begin{bmatrix} 36 & 3 \cdot h & -36 & 3 \cdot h \\ 3 \cdot h & 4 \cdot h^2 & -3 \cdot h & -h^2 \\ -36 & -3 \cdot h & 36 & -3 \cdot h \\ 3 \cdot h & -h^2 & -3 \cdot h & 4 \cdot h^2 \end{bmatrix}, \end{aligned}$$

$$\begin{aligned} \hat{a}_{i,j}^\beta &= \int_{\tilde{x}_i}^{\tilde{x}_{i+1}} \beta \cdot \frac{d^2\varphi_i(x)}{dx^2} \cdot \frac{d^2\varphi_j(x)}{dx^2} \cdot dx = \\ &= \frac{\beta}{h^3} \begin{bmatrix} 60 & 30 \cdot h & -60 & 30 \cdot h \\ 30 \cdot h & 16 \cdot h^2 & -30 \cdot h & 14 \cdot h^2 \\ -60 & -30 \cdot h & 60 & -30 \cdot h \\ 30 \cdot h & 14 \cdot h^2 & -30 \cdot h & 16 \cdot h^2 \end{bmatrix}. \end{aligned}$$

## БАЗИСНЫЕ СПЛАЙНЫ

Появление в середине XX века станков с ЧПУ породило потребность в цифровой модели детали, необходимой для создания управляющей программы станка. По заказу предприятий аэрокосмической и автомобильной промышленности изучением принципов моделирования трехмерных объектов занялись различные исследовательские группы, что и привело в итоге к разработке базисных сплайнов. Кривые Безье, являющиеся частным случаем многочленов Бернштейна, были разработаны независимо друг от друга Пьером Безье (1962 г.) из компании «Рено» и Полем де Кастельжо (1959 г.) из компании «Ситроен», где применялись для проектирования кузовов автомобилей. Поскольку Безье раньше опубликовал свои разработки, то сплайны были названы его именем, а именем де Кастельжо назван рекурсивный способ определения параметрических поверхностей (алгоритм де Кастельжо). Впоследствии это открытие стало одним из важнейших инструментов САПР и программ компьютерной графики, таких, как AutoCAD и Photoshop, вследствие простоты задания и манипуляции.

Оказалось, что сплайн Безье является частным случаем базисных сплайнов, введенных Шенбергом в 1947 г. в качестве обозначения класса полиномиальных сплайнов, использовавшихся до этого лишь в качестве инструмента теоретических исследований при решении различных экстремальных и вариационных задач, в частности, в теории приближений. Термин «В-сплайн» является сокращением от словосочетания «базисный сплайн».

Первой работой с упоминанием неравномерных рациональных В-сплайнов (NURBS, non-uniform rational B-spline), являющихся обобщением сплайна Безье, стала диссертация Кена Версприла (1975 г.). Позже NURBS стали неотъемлемой частью профессиональных приложений для компьютерной графики, таких, как Maya, CorelDraw и 3DS Max. В 2005 году «CAD Society», некоммерческая ассоциация отрасли САПР, присудила Кену Версприлу награду за неоценимый вклад в технологию САПР в виде NURBS.



## СПЛАЙНЫ БЕЗЬЕ

*Кривая Безье* – это параметрическое по  $t \in [0, 1]$  уравнение, являющееся линейной комбинацией базисных функций Безье (представляющих собой базисные полиномы Бернштейна) степени  $n$  и контрольных точек  $P_i = (x_i, y_i)$ :

$$B(t) = \sum_{i=0}^n P_i \cdot b_{i,n}(t).$$

Производная кривой Безье является годографом (рис.2Рис.20) и может быть рассчитана по формуле, получаемой из формулы для производной многочлена Бернштейна:

$$B'(t) = \sum_{i=0}^{n-1} Q_i \cdot b_{i,n-1}(t),$$

где  $Q_i = n \cdot (P_{i+1} - P_i)$  – это, фактически, новые контрольные точки.

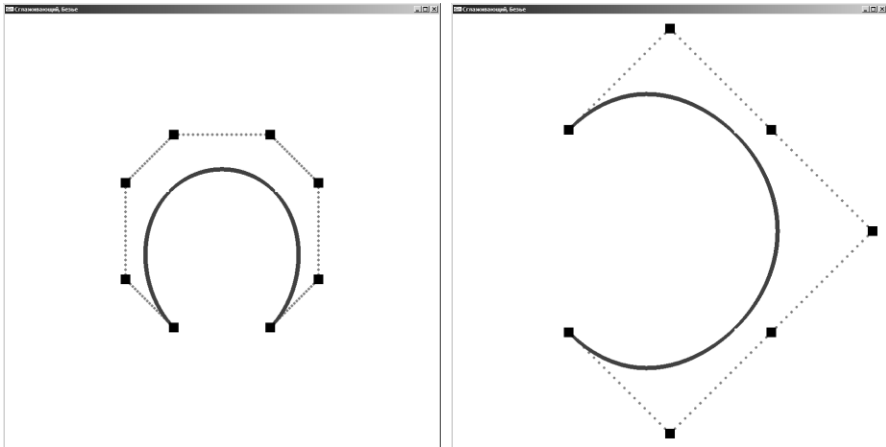


Рис.20. Кривая Безье и ее производная.

## ОСНОВНЫЕ ВИДЫ КРИВЫХ БЕЗЬЕ

В связи с трудоемкостью расчетов в современных графических системах обычно используются лишь линейные, квадратичные и кубические кривые Безье (Таблица 1), из которых и собираются более сложные кривые. В этом случае гладкость в месте соединения двух кривых обеспечивается расположением трех смежных контрольных точек на одной прямой.

Переход от квадратичной формы кривой с контрольными точками  $(P_0, P_1, P_2)$  к кубической с  $(\tilde{P}_0, \tilde{P}_1, \tilde{P}_2, \tilde{P}_3)$  можно осуществить следующим образом:

$$\begin{aligned}\tilde{P}_0 &= P_0, \\ \tilde{P}_1 &= P_0 + \frac{2}{3}(P_1 - P_0), \\ \tilde{P}_2 &= P_1 + \frac{1}{3}(P_2 - P_1), \\ \tilde{P}_3 &= P_2.\end{aligned}$$

В матричной форме кубическую кривую Безье можно представить через базисную матрицу Безье:

$$B(t) = [t^3, t^2, t, 1] \cdot \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}.$$

Таблица 1

Вид кривой Безье для степеней 2-4

$n$	Кривая Безье степени $n$
1	$(1 - t) \cdot P_0 + t \cdot P_1$
2	$(1 - t)^2 \cdot P_0 + 2 \cdot t \cdot (1 - t) \cdot P_1 + t^2 \cdot P_2$
3	$(1 - t)^3 \cdot P_0 + 3 \cdot t \cdot (1 - t)^2 P_1 + 3 \cdot t^2 \cdot (1 - t) P_2 + t^3 \cdot P_3$

## НЕКОТОРЫЕ СВОЙСТВА

1. Кривая сплайна является гладкой и непрерывной, проходящей через начальную и конечную контрольные точки, при этом отрезки  $(P_0, P_1)$  и  $(P_n, P_{n-1})$  являются касательными к кривой в точках  $P_0$  и  $P_n$ .

2. Любой участок кривой сплайна также является кривой Безье.

3. Степень кривой всегда равна  $n$  (на 1 меньше числа контрольных точек, как показано в табл.1), при этом дефект кривой равен 1.

4. Поскольку сплайн является глобальным, изменение координат любой из точек ведет к изменению формы всей кривой, как и добавление новой контрольной точки.

5. Кривая повторяет траекторию своей контрольной ломаной, образуемой контрольными точками, но ее сложность не больше, чем у контрольной ломаной, поскольку количество пересечений кривой Безье с прямой линией не может быть больше, чем количество пересечений этой линии с контрольной ломаной.

6. Кривая целиком лежит в выпуклой оболочке своих контрольных точек. Это значительно облегчает задачу нахождения точек пересечения кривых (если не пересекаются выпуклые оболочки, то не пересекаются и сами кривые) и позволяет визуализировать кривую с помощью ее контрольных точек.

7. Прямая линия образуется при размещении контрольных точек на одной прямой.

8. Конические сечения, в частности, окружность, не могут быть описаны параметрическим уравнением кривой Безье, но могут быть приближены с необходимой точностью.

9. Кривая Безье является аффинно-инвариантной. Таким образом, аффинные преобразования кривой могут быть осуществлены путем применения соответствующих трансформаций к контрольным точкам. В частности, обмен местами между начальной и конечной точками (изменение направления траектории) не влияет на форму кривой.

10. Все базисные функции  $b_{i,n}(t)$  меняются в диапазоне  $[0,1]$ , а их сумма при конкретном значении  $t$  равна 1.

## ОСОБЕННОСТИ РАСЧЕТА БАЗИСНЫХ ФУНКЦИЙ

Поскольку степень кривой всегда равна  $n$  (на 1 меньше числа контрольных точек), то увеличение числа контрольных точек приводит к росту степени сплайна, что, в свою очередь, вызывает вычислительные затруднения при расчете биномиальных коэффициентов базисных функций.

Основная проблема – это даже не рост вычислительных затрат, а рост значений используемых факториалов. В Таблица 2 для различных типов данных языка C++ приведены максимальные значения факториалов, которые могут быть напрямую рассчитаны. Для вещественных типов указана пара максимумов, которые могут быть посчитаны как точно, так и с погрешностью, вызванной размером мантиссы. Использование формулы Стирлинга разрешает эту проблему, однако погрешность при ее использовании улучшается до 9-го знака мантиссы лишь к факториалу 12.

Соответственно, для повышения устойчивости и был разработан рекурсивный алгоритм де Кастельжо, позволяющий достаточно эффективно рассчитывать полиномы Бернштейна.

Таблица 2

Таблица соотношения типа данных и максимального факториала

Тип данных	Число байт	Максимальное значение для типа	Максимальный факториал	
			Точный расчет	Примерный расчет
int	4	4.2e+9	12!	
long long	8	1.8e+19	20!	
float	4	3.4e+38	10!	33!
double	8	1.7e+308	17!	170!
long double	10	1.1e+4932	20!	1754!

## ГЕОМЕТРИЧЕСКАЯ ИНТЕРПРЕТАЦИЯ

Идея рекурсивного алгоритма де Кастельжо построения кривой Безье через параметр  $t \in [0,1]$  заключается в следующем: каждая точка  $B(t)$  кривой находится путем построения сокращающейся последовательности ломаных  $(P_0^{(i)}, P_1^{(i)}, \dots, P_{n-i}^{(i)})$ , последняя из которых (выродившись в точку) и даст требуемую точку кривой:

$$\begin{aligned} & (P_0^{(0)}, P_1^{(0)}, \dots, P_{n-2}^{(0)}, P_{n-1}^{(0)}, P_n^{(0)}), \\ & (P_0^{(1)}, P_1^{(1)}, \dots, P_{n-2}^{(1)}, P_{n-1}^{(1)}), \\ & (P_0^{(2)}, P_1^{(2)}, \dots, P_{n-2}^{(2)}), \\ & \dots \\ & (P_0^{(n-2)}, P_1^{(n-2)}), \\ & (P_0^{(n-1)}), \end{aligned}$$

где:  $P_i^{(0)} = P_i$ ,  $i = \overline{1, n}$ ,

$$P_j^{(k)} = (1-t) \cdot P_j^{(k-1)} + t \cdot P_{j+1}^{(k-1)}, \quad j = \overline{1, n-k}, \quad k = \overline{1, n-1}.$$

На рис.21 приведен пример построения трех точек (для трех значений параметра  $t$ ) кривой Безье по трем контрольным точкам. В данном примере точки  $Q_0(x, y)$  и  $Q_1(x, y)$  делят отрезки  $(P_0, P_1)$  и  $(P_1, P_2)$  в пропорции  $(1-t) \div t$  на две точки:

$$Q_0 = (1-t) \cdot P_0 + t \cdot P_1,$$

$$Q_1 = (1-t) \cdot P_1 + t \cdot P_2.$$

Точка же сплайна  $B(x, y)$  получается из аналогичной пропорции для отрезка  $(Q_0, Q_1)$ :

$$B = (1-t) \cdot Q_0 + t \cdot Q_1.$$

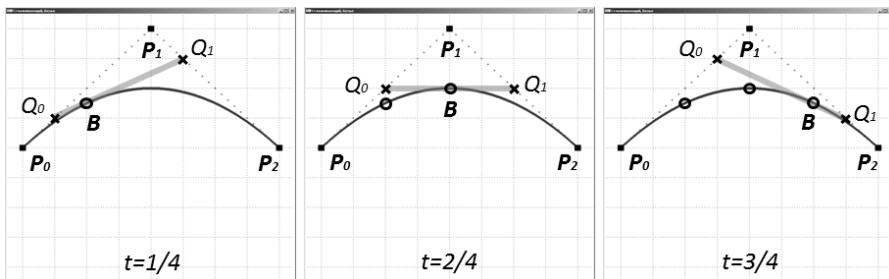


Рис.21. Пример построения кривой Безье по трем точкам.

На рис.22 приведен аналогичный пример построения трех точек кривой Безье уже по четырем контрольным точкам. В данном примере точки  $Q_0$ ,  $Q_1$  и  $Q_2$  пропорционально делят отрезки  $(P_0, P_1)$ ,  $(P_1, P_2)$  и  $(P_2, P_3)$  соответственно, а точки  $R_0$  и  $R_1$  – отрезки  $(Q_0, Q_1)$  и  $(Q_1, Q_2)$ :

$$Q_0 = (1 - t) \cdot P_0 + t \cdot P_1,$$

$$Q_1 = (1 - t) \cdot P_1 + t \cdot P_2.$$

$$Q_2 = (1 - t) \cdot P_2 + t \cdot P_3.$$

$$R_0 = (1 - t) \cdot Q_0 + t \cdot Q_1,$$

$$R_1 = (1 - t) \cdot Q_1 + t \cdot Q_2.$$

Точка же сплайна  $B(x, y)$  получается из аналогичной пропорции для отрезка  $(R_0, R_1)$ :

$$B = (1 - t) \cdot R_0 + t \cdot R_1.$$

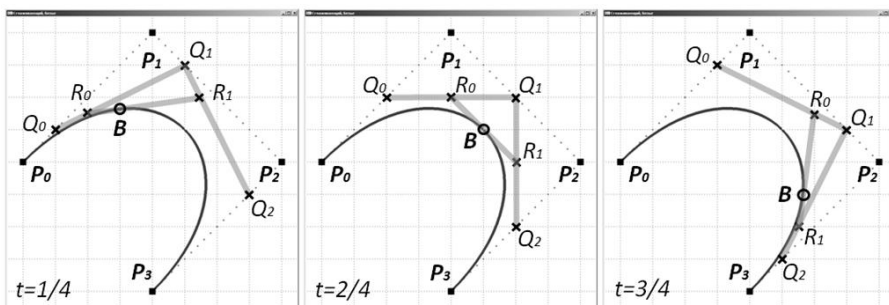


Рис.22. Пример построения кривой Безье по трем точкам.

## ПОВЕРХНОСТЬ БЕЗЬЕ

Поверхность Безье – это параметрическая поверхность, являющаяся тензорным произведением двух кривых Безье с помощью двух независимых параметров  $u$  и  $v$ :

$$B(u, v) = \sum_{i=0}^n \sum_{j=0}^k P_{ij} \cdot b_{i,n}(u) \cdot b_{j,m}(v), \quad u, v \in [0, 1],$$

где  $P_{ij} = P(x_i, y_j)$  – это трехмерные контрольные точки.

Идея *кусочного моделирования* (patch modeling) заключается в том, чтобы создавать объект в виде сетчатых поверхностей, кривизна которых регулируется сплайнами. Данные поверхности состоят из набора "сшитых" друг с другом сегментов (кусков, лоскутов, патчей), вид которых определяется сеткой контрольных точек и соответствующим выбранным сплайном. При необходимости эти сегменты можно и дальше подразделять на элементы для повышения качества получаемой поверхности (увеличение разрешения). На рис.23 приведен пример такого сегмента.

К недостаткам такого рода объектов относится, например, проблематичность поиска их пересечения с прямой, который используется в трассировке лучей. Кроме того, затруднения возникают и при перспективном проецировании. На стадии же визуализации такие объекты все равно разбиваются на плоские треугольники.

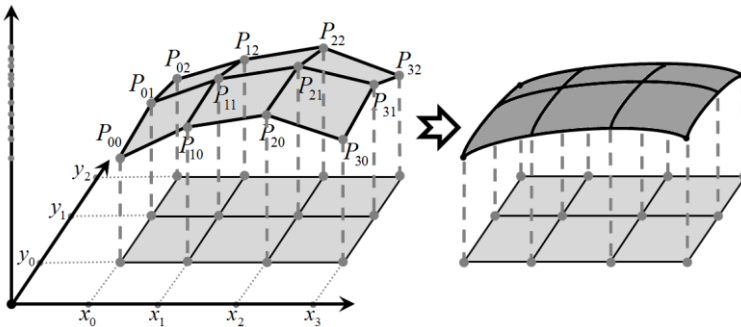


Рис.23. Пример сегмента двумерной поверхности.

## В-СПЛАЙНЫ

*В-сплайн* (базисный сплайн) – это параметрическое уравнение, являющееся линейной комбинацией базисных полиномов Бернштейна степени  $m \leq n$  и контрольных точек  $P_i = (x_i, y_i)$ :

$$B(t) = \sum_{i=0}^n P_i \cdot b_{i,m}(t), \quad t \in T,$$

где вектор  $T$  – это специальный *вектор параметров* (узлов), выбор которого влияет на вид сплайна.

*Поверхность В-сплайна* – это параметрическая поверхность, являющаяся тензорным произведением двух одномерных В-сплайнов степени  $p$  и  $q$  с помощью двух независимых параметров  $u$  и  $v$ :

$$B(u, v) = \sum_{i=0}^n \sum_{j=0}^k P_{ij} \cdot b_{i,p}(u) \cdot b_{j,q}(v), \quad u, v \in [0, 1],$$

где  $P_{ij} = P(x_i, y_j)$  – это трехмерные контрольные точки.

По сути, В-сплайны являются обобщением сплайнов Безье, в результате наследует большинство его свойств.

В-сплайн является сплайном с наименьшим носителем (областью, где функция не равна нулю) для заданных степени полинома, порядка гладкости и разбиения области определения. Фундаментальная теорема устанавливает, что любая сплайн-функция может быть представлена в виде линейной комбинации В-сплайнов той же степени и гладкости на той же области определения.

В-сплайны могут быть вычислены с помощью рекурсивного алгоритма де Бора, обобщающего алгоритм де Кастельжо:

$$b_{i,0}(t) = \begin{cases} 1, & \text{если } t \in [t_i, t_{i+1}) \\ 0, & \text{иначе} \end{cases}$$

$$b_{i,j}(t) = \frac{t - t_i}{t_{i+j} - t_i} b_{i,j-1}(t) + \frac{t_{i+j+1} - t}{t_{i+j+1} - t_{i+1}} b_{i+1,j-1}(t), \quad j = \overline{1, m}$$



## ВЕКТОР ПАРАМЕТРОВ

В отличие от обычной сплайн-интерполяции, сшивка элементарных В-сплайнов производится не в контрольных точках, а в других, координаты которых обычно предлагается определить пользователю.

Формы базисных функций  $b_{k,m}(t)$  определяются относительным расстоянием между компонентами вектора параметров  $T$  размерностью  $n + m + 1$ , что приводит к получению различных видов В-сплайна. Выбор значений  $T$  неоднозначен, но для всех его компонент должно выполняться общее условие:  $t_i \leq t_{i+1}$ .

Если расстояние между всеми компонентами  $T$  одинаково ( $t_{i+1} - t_i = \text{const}$ ), тогда вектор параметров называется *однородным* (uniform), если это условие нарушается, тогда вектор становится *неоднородным* (nonuniform). Вектор  $T$  становится *нормированным*, если его значения пронормированы и меняются от 0 до 1 включительно. Если первые  $m + 1$  узлов равны 0, а последние  $m + 1$  равны 1, тогда такой вектор становится *периодическим* (periodic). При расчете значений сплайна в точке имеет смысл не перебирать первые и последние  $m$  значений из сформированного вектора  $T$  в связи с обнулением базисных функций на этих интервалах.

Обычно выделяют 3 типа В-сплайна: открытый (open B-Spline), закрытый (clamped B-Spline) и замкнутый (closed B-spline).

Особенностью *открытого В-сплайна* является то, что он не доходит до первой и последней контрольных точек (пример на рис.24). Данный сплайн можно задать следующим вектором параметров:

$$T = (0, 1, 2, \dots, n + m).$$

В отличие от открытого сплайна, *закрытый В-сплайн* приходит в контрольные точки  $P_0$  и  $P_n$ . Данный сплайн можно задать следующим вектором параметров:

$$T = T^0 + T^1 + T^2,$$

где:

$$\begin{aligned} T_{1 \times (m)}^0 &= (0, 0, \dots, 0), \\ T_{1 \times (n-m)}^1 &= (1, 2, \dots, n - m), \\ T_{1 \times (m)}^0 &= (n - m + 1, \dots, n - m + 1). \end{aligned}$$

Особенностью *замкнутого В-сплайна* является совпадения его начала и конца, в результате сплайн становится замкнутой кривой (пример на рис.25). Для получения замкнутой кривой необходимо к вектору контрольных точек добавить в конец первые  $m$  точек и задать вектор параметров:

$$T = (0, 1, 2, \dots, n + 2 \cdot m + 2),$$

$$P = (P_0, P_1, \dots, P_n) \cup (P_0, P_1, \dots, P_{m-1}).$$

Если физически не менять вектор контрольных точек, тогда можно просто скорректировать формулу В-сплайна:

$$B(t) = \sum_{i=m}^n P_i \cdot b_{i,m}(t) + \sum_{i=0}^{m-1} P_i \cdot b_{i+n+1,m}(t).$$

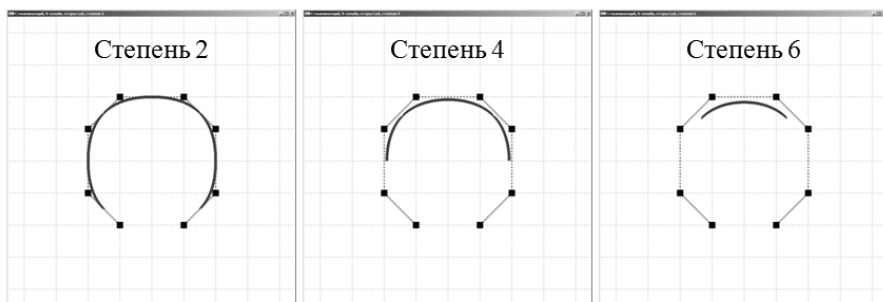


Рис.24. Примеры работы открытого однородного В-сплайна.

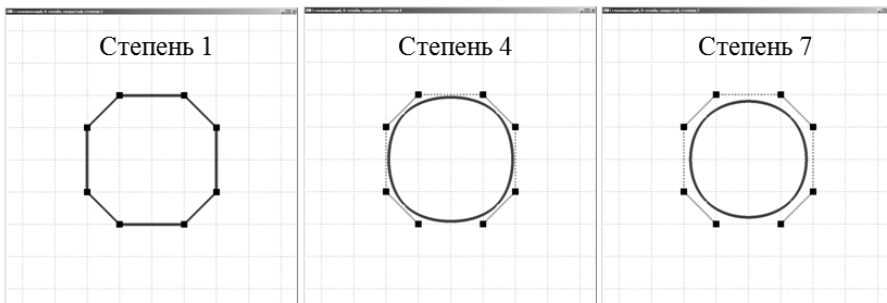


Рис.25. Примеры работы замкнутого однородного В-сплайна.

## НЕКОТОРЫЕ СВОЙСТВА

1. Когда степень В-сплайн становится максимально возможной, сплайн вырождается в кривую Безье, при этом отпадает необходимость в векторе  $T$ , который примет вид:  $(0, 0, \dots, 0, 1, 1, \dots, 1)$ .

2. Вид сплайна определяется вектором его параметров.

3. Поскольку базисный полином  $b_{i,m}(t)$  не обращается в нуль только на промежутке  $[t_i, t_{i+m+1}]$ , т.е. контрольная точка  $P_k$  влияет на кривую лишь в этом диапазоне, то сплайн в общем случае является локальным.

4. Поскольку сплайн является локальным, изменение координат любой из точек ведет к изменению формы кривой лишь в некоторой области, как и при добавлении новой контрольной точки.

5. При необходимости степень сплайна  $m$  можно менять в пределах  $[1, n]$ , получая кривые различной степени на одном и том же наборе контрольных точек. В частности, при  $m = 2$  сплайн совпадает с контрольной ломаной.

6. Кривая повторяет траекторию своей контрольной ломаной, образуемой контрольными точками, и проходит рядом с серединой каждого ее отрезка.

7. Кривая целиком лежит в выпуклой оболочке своих контрольных точек.

8. Конические сечения, в частности, окружность, не могут быть описаны параметрическим уравнением сплайна, но могут быть приближены с необходимой точностью.

9. В-сплайн остается аффинно-инвариантным.

10. В основном применяются В-сплайны степени 2 и 3 (квадратичные и кубические), причем квадратичные применяются при разработке шрифтов, таких, как ClearType.

## NURBS-СПЛАЙНЫ

*NURBS-сплайн* (Non-uniform rational B-spline) – это неоднородный рациональный В-сплайн, т.е. параметрическое уравнение, являющееся взвешенной линейной комбинацией полиномов Бернштейна степени  $m \leq n$  и контрольных точек  $P_i = (x_i, y_i)$ :

$$B(t) = \frac{\sum_{i=0}^n \omega_i \cdot P_i \cdot b_{i,m}(t)}{\sum_{i=0}^n \omega_i \cdot b_{i,m}(t)},$$

где  $\omega_i$  – это веса контрольных точек, регулирующих меру близости сплайна к ним.

*Поверхность NURBS -сплайна* – это параметрическая поверхность, являющаяся тензорным произведением двух одномерных NURBS-сплайнов степени  $p$  и  $q$  с помощью двух независимых параметров  $u$  и  $v$ :

$$B(t) = \frac{\sum_{i=0}^n \sum_{j=0}^k \omega_i \cdot P_{ij} \cdot b_{i,p}(t) \cdot b_{j,q}(t)}{\sum_{i=0}^n \sum_{j=0}^k \omega_i \cdot b_{i,p}(t) \cdot b_{j,q}(t)},$$

где  $P_{ij} = P(x_i, y_j)$  – это трехмерные контрольные точки.

NURBS-сплайны получили широкое распространение из-за своей стандартизованности и относительной простоты, стали неотъемлемой частью таких стандартных пакетов программ для компьютерной графики, как Maya и 3ds Max.

## *НЕКОТОРЫЕ СВОЙСТВА*

1. NURBS-сплайны являются обобщением с использованием однородных координат В-сплайнов и наследуют большинство их свойств.
2. NURBS кривая в трехмерном пространстве является проекций В-сплайновой кривой в четырехмерном пространстве.
3. Когда степень сплайна становится максимально возможной, сплайн вырождается в рациональную кривую Безье.
4. Если все веса  $\omega_i = 1$ , тогда сплайн вырождается в В-сплайн.
5. Сплайн в общем случае является локальным. Вид сплайна определяется вектором его параметров и выбором весов и степени.
6. Конические сечения, в частности, окружность, могут быть точно описаны параметрическим уравнением сплайна.
7. В-сплайн является не просто аффинно-инвариантным, а проективно-инвариантным: не только аффинные, но и проективные преобразования кривой могут быть осуществлены путем применения соответствующих трансформаций к контрольным точкам.

## *ПРИМЕРЫ ВИЗУАЛИЗАЦИИ*

На рис.26 приведен пример визуализации NURBS-поверхности и его контрольных точек, а на рис.27 приведен пример моделирования окружности ([https://ru.qwertyu.wiki/wiki/Non-uniform\\_rational\\_B-spline](https://ru.qwertyu.wiki/wiki/Non-uniform_rational_B-spline)).

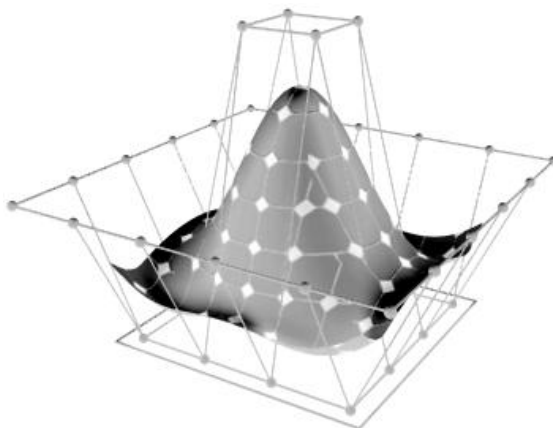


Рис.26. Пример визуализации NURBS-сплайна.

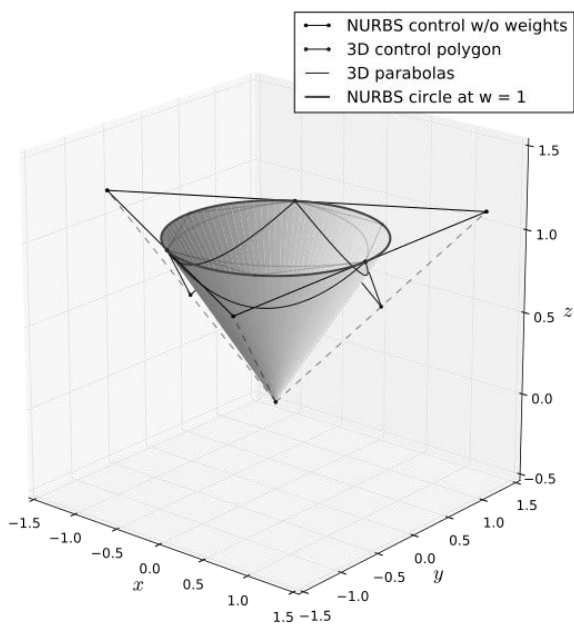


Рис.27. Пример построения окружности NURBS-сплайном.

## РЕАЛИЗАЦИЯ В БИБЛИОТЕКЕ OPENGL

В составе библиотеки OpenGL имеются средства поддержки работы с кривыми и поверхностями Безье, позволяющие вычислять значения полиномов Безье – так называемый *Безье-вычислитель*.

### ФУНКЦИИ ВКЛЮЧЕНИЯ ВЫЧИСЛИТЕЛЯ

По умолчанию расчет сплайна отключен. Включение и выключение расчета кривой или поверхности (одномерного или двумерного сплайна Безье) производится вызовом стандартной пары функций `glEnable` и `glDisable` с параметром:

- 1) `GL_MAP1_VERTEX_3` – для кривой  
в  $(x, y, z)$ -координатах;
- 2) `GL_MAP1_VERTEX_4` – для кривой  
в  $(x, y, z, w)$ -координатах;
- 3) `GL_MAP2_VERTEX_3` – для поверхности  
в  $(x, y, z)$ -координатах;
- 4) `GL_MAP2_VERTEX_4` – для поверхности  
в  $(x, y, z, w)$ -координатах.

Здесь префиксы `MAP1` и `MAP2` указывают на то, является сплайн одномерным или же двумерным. Логично, что именно этот режим и должен быть использован при задании параметров остальных функций.

## *ФУНКЦИИ РАСЧЕТА ПАРАМЕТРОВ СПЛАЙНА*

Функция обработки полинома одной переменной настраивается посредством вызова функции `glMap1`:

```
void glMap1f( GLenum target,
              GLfloat u1, GLfloat u2,
              GLint stride,
              GLint order,
              const GLfloat* points);

void glMap1d( GLenum target,
              GLdouble u1, GLdouble u2,
              GLint stride,
              GLint order,
              const GLdouble * points).
```

Параметр `target` задает тип объекта, который будет представлен полиномом Безье: размерность геометрических точек, режим цвета, нормали и координаты текстур (Таблица 3). Параметры `u1` и `u2` задают значения  $t_0$  и  $t_1$  соответственно. Параметры `stride`, `order` и `points` задают свойства массива данных, параметры которых устанавливаются текущим вызовом: смещение в массиве (размер элемента), количество элементов и сам массив данных.

Функция обработки полинома двух переменных настраивается посредством вызова аналогичной функции `glMap2` с параметрами, которые несут тот же самый смысл, но уже для обеих осей (в этом случае префикс `GL_MAP1` меняется на `GL_MAP2`):

```
void glMap2f( GLenum target,
              GLfloat u1, GLfloat u2,
              GLint ustride, GLint uorder,
              GLfloat v1, GLfloat v2,
              GLint vstride, GLint vorder,
              const GLfloat* points);
```



```
void glMap2d( GLenum target,
              GLdouble u1, GLdouble u2,
              GLint ustride, GLint uorder,
              GLdouble v1, GLdouble v2,
              GLint vstride, GLint vorder,
              const GLdouble* points );
```

Таблица 3

**Значения параметра target**

Значение	Описание
GL_MAP1_VERTEX_3	Режим координат точек: $(x, y, z)$
GL_MAP1_VERTEX_4	Режим координат точек: $(x, y, z, w)$
GL_MAP1_INDEX	Режим цвета: индексированный
GL_MAP1_COLOR_4	Режим цвета: <i>RGBA</i>
GL_MAP1_NORMAL	Режим задания нормалей
GL_MAP1_TEXTURE_COORD_1	Режим координат текстур: $(s)$
GL_MAP1_TEXTURE_COORD_2	Режим координат текстур: $(s, t)$
GL_MAP1_TEXTURE_COORD_3	Режим координат текстур: $(s, t, r)$
GL_MAP1_TEXTURE_COORD_4	Режим координат текстур: $(s, t, r, q)$

## *ФУНКЦИИ ВИЗУАЛИЗАЦИИ СПЛАЙНА*

Функция `glEvalCoord1` визуализирует координаты сплайна в указанной пользователем точке  $t = u$ , заменяя собою обращение к функциям `glVertex`, `glColor` и `glNormal`:

```
void glEvalCoord1f( GLfloat u );  
void glEvalCoord1fv( GLfloat* u );  
void glEvalCoord1d( GLdouble u );  
void glEvalCoord1dv( GLdouble* u ).
```

Если значения параметра  $t$  распределены равномерно, то для автоматического вычисления массива координат сплайна на интервале, заданном значениями параметров  $u1$  и  $u2$ , следует использовать функцию `glMapGrid1`, а для визуализации массива в пределах индексов  $i1$  и  $i2$  – функцию `glEvalMesh`:

```
void glMapGrid1f( GLint un,  
                 GLfloat u1, GLfloat u2 );  
void glMapGrid1d( GLint un,  
                 GLdouble u1, GLdouble u2 );  
void glEvalMesh1( GLenum mode,  
                 GLint i1, GLint i2).
```

В двумерном случае используются аналогичные функции с параметрами, которые несут тот же самый смысл, но уже для обеих осей:

```
void glEvalCoord2f( GLfloat u, GLfloat v );  
void glEvalCoord2fv( GLfloat* u, GLfloat* v );  
void glEvalCoord2d( GLdouble u, GLfloat v );  
void glEvalCoord2dv( GLdouble* u, GLfloat* v ).
```

```

void glMapGrid2f( GLint un,
                  GLfloat u1, GLfloat u2,
                  GLint vn,
                  GLfloat v1, GLfloat v2 );

void glMapGrid2d( GLint un,
                  GLdouble u1, GLdouble u2,
                  GLint vn,
                  GLdouble v1, GLdouble v2 );

void glEvalMesh2( GLenum mode,
                  GLint i1, GLint i2,
                  GLint j1, GLint j2 ).

```

### *ПРИМЕР ЗАДАНИЯ КРИВОЙ БЕЗЪЕ*

```

/* Расчет значений сплайна */
int n = XY.size() - 1;
int parts = 20 * n;
glEnable ( GL_MAP1_VERTEX_3 );
glMap1f ( GL_MAP1_VERTEX_3, 0f, 1f, 3, n, XY.data() );
// glMapGrid1 ( parts, 0f, 1f); // для равномерного
glDisable(GL_MAP1_VERTEX_3);

/* Вывод неравномерного сплайна на экран */
glBegin ( GL_LINE_STRIP );
for ( int i = 0; i <= parts; i++ )
    glEvalCoord1f ( float(i) / parts );
glEnd();

/* Вывод равномерного сплайна на экран */
glEvalMesh1 (GL_LINE_STRIP, 0, parts);

```

## РЕАЛИЗАЦИЯ В БИБЛИОТЕКЕ GLU

В библиотеке GLU реализована поддержка одномерных и двумерных NURBS-сплайнов.

Для создания объекта типа NURBS используется функция `gluNewNurbsRenderer`, которая создает объект и возвращает на него указатель:

```
GLUnurbs* WINAPI gluNewNurbsRenderer ( void ).
```

Удаление же объекта данного типа выполняется командой `gluDeleteNurbsRenderer`:

```
void gluDeleteNurbsRenderer ( GLUnurbs* nurb ).
```

### *ФУНКЦИИ ЗАДАНИЯ ПАРАМЕТРОВ СПЛАЙНА*

Функция `gluNurbsProperty` используется для установки значений различных свойств, например, максимального размера линий (полигонов), используемых для визуализации кривой (поверхности), функция также позволяет активизировать режим, в котором тесселированные геометрические данные могут быть получены через интерфейс `callback-функций`:

```
void gluNurbsProperty ( GLUnurbs* nurb,  
                        GLenum property, GLfloat value ).
```

Для получения уведомлений об ошибках используется функция `gluNurbsCallback`, также она позволяет задать `callback-функции`, которые будут вызываться для извлечения тесселированных геометрических данных:

```
void gluNurbsCallback ( GLUnurbs* nurb,  
                        GLenum which, _GLUfuncptr CallBackFunc ).
```

## ФУНКЦИИ ЗАДАНИЯ СПЛАЙНА

Начало и завершение описания требуемой кривой или поверхности (по аналогии с операторными скобками) выполняется соответствующими функциями `gluBeginCurve`, `gluBeginSurface`, `gluEndCurve` и `gluEndSurface`:

```
void gluBeginCurve ( GLUnurbs* nurb ),  
void gluBeginSurface ( GLUnurbs* nurb ),  
void gluEndCurve ( GLUnurbs* nurb ),  
void gluEndSurface ( GLUnurbs* nurb ).
```

Для задания контрольных точек и нормалей к кривой или поверхности используются функции `gluNurbsCurve` или `gluNurbsSurface`:

```
void gluNurbsCurve( GLUnurbs* nurb,  
                    GLint knotCount, GLfloat* knots,  
                    GLint stride,  
                    GLfloat* control,  
                    GLint order,  
                    GLenum type).  
  
void gluNurbsSurface( GLUnurbs* nurb,  
                     GLint sKnotCount, GLfloat* sKnots,  
                     GLint tKnotCount, GLfloat* tKnots,  
                     GLint sStride, GLint tStride,  
                     GLfloat* control,  
                     sOrder,  
                     GLint tOrder,  
                     GLenum type);
```

Параметры `knots` и `knotCount` устанавливают вектор узлов  $T$  и его размерность, а параметры `sKnots` (`tKnots`) и `sKnotCount` (`tKnotCount`) выполняют аналогичные действия для векторов узлов  $T^s$  и  $T^t$  по каждой из осей.

### *ПРИМЕР ЗАДАНИЯ NURBS-КРИВОЙ*

```
GLUnurbsObj *obj = gluNewNurbsRenderer();

gluBeginCurve(obj);

gluNurbsCurve (
    obj,
    Knots.size(), Knots.data(),
    3,
    XY.data(),
    degree_m + 1,
    GL_MAP1_VERTEX_3);

gluEndCurve (obj);

gluDeleteNurbsRenderer(obj);
```

### *ПРИМЕР ЗАДАНИЯ NURBS-ПОВЕРХНОСТИ*

```
GLUnurbsObj *obj = gluNewNurbsRenderer();

gluBeginSurface(obj);

gluNurbsSurface
    (obj,
     8, Knots, 8, Knots,
     4*3, 3,
     &XYZ[0][0][0],
     degree_p, degree_q,
     GL_MAP2_VERTEX_3);

gluEndSurface(theNurb);

gluDeleteNurbsRenderer(obj);
```

## ВИЗУАЛИЗАЦИЯ СПЛАЙНОВ

Существуют разные подходы к отрисовке сплайнов в двумерной или трехмерной области, но суть у всех одна: требуется получить множество пар (в двумерном случае) или троек (в трехмерном случае), являющимися координатами графика сплайна, которые затем непрерывно соединяются. Очевидно, что расчет координат имеет смысл выполнять заранее, дабы при отрисовке не тратилось время на расчет самого сплайна.

В общем случае, для получения координат графика сплайна нужно пробежаться по всей области определения с некоторым мелким шагом и запомнить значение сплайна в каждой из полученных точек. Проблемой может стать накопление ошибок и непрохождение сплайна через точки интерполяции из-за дискретности шага. Кроме того, выбор мелкости шага (уровня дискретизации) влияет и на гладкость графика. Соответствующие примеры приведены на рис.28.

Вообще говоря, выбор уровня дискретизации зависит от текущего масштаба, скорости изменения функции и от размеров окна в пикселях.

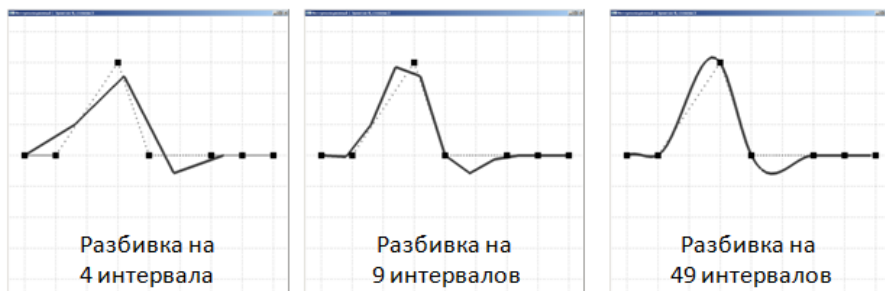


Рис.28. Влияние уровня дискретизации на качество графика.

## ВЫВОД ГРАФИКА НА ЭКРАН

Для одномерного сплайна, заданного на области  $[x_0, x_n]$ , визуализацию удобно проводить с помощью примитива «замкнутая ломаная»:

```
glBegin ( GL_LINE_STRIP );
for( int i = 0; i < SplineGraphN; i++ )
    glVertex2f ( SplineGraphX[i], SplineGraphY[i] );
glEnd();
```

Для двумерного сплайна, заданного на области  $[x_0, x_n] \times [y_0, y_k]$ , визуализацию удобно проводить с помощью двумерного примитива «четырёхугольники с общей границей» или его треугольного аналога:

```
for( int j = 0; j < SplineGraphNy - 1; j++ )
{
    float x0 = SplineGraphX[i], x1 = SplineGraphX[i+1];
    glBegin ( GL_QUAD_STRIP );
    for( int i = 0; i < SplineGraphNx - 1; i++ )
    {
        float y0 = SplineGraphY[j], y1 = SplineGraphY[j+1];
        glVertex3f ( x0, y0, SplineGraphZ[i+0][j+0] );
        glVertex3f ( x0, y1, SplineGraphZ[i+0][j+1] );
        glVertex3f ( x1, y1, SplineGraphZ[i+1][j+1] );
        glVertex3f ( x1, y0, SplineGraphZ[i+1][j+0] );
    }
    glEnd();
}
```

При визуализации двумерного сплайна в целях удобства восприятия имеет смысл задать в узлах четырёхугольника вектор нормали и включить освещение.



## РАСЧЕТ КООРДИНАТ ГРАФИКА

Самый простой способ получения координат сплайна – это пробег по всей области определения  $[x_0, x_n]$  с мелким шагом  $h$ , вычисление  $y = S(x)$  и запоминание пары  $(x, y)$  командой SplineGraphAdd:

```
int SplineSplit = 1000; // количество разбиений области
double h = ( OX[N] - OX[0] ) / SplineSplit;
for( double x = OX[0]; x <= OX[N]; x += h )
{
    double y = GetSplineValue ( x );
    SplineGraphAdd ( x, y );
}
```

В случае параметрического сплайна областью определения становится интервал  $[t_0, t_n]$ , соответственно, нужно аналогично пробежать весь этот интервал с некоторым шагом. Например, для сплайна Безье код может выглядеть следующим образом:

```
for( double t = 0; t <= 1.00001; t += h )
{
    double x = 0;
    double y = 0;
    for( int i = 0; i <= N; i++ )
    {
        double basis = BasisBezier(t, i, n);
        x += OX[i] * basis;
        y += OY[i] * basis;
    }
    SplineGraphAdd (x, y );
}
```

Чтобы гарантированно попасть в точки интерполяции, разбиение лучше осуществлять на каждом из интервалов:

```
int SplineSplit = 100; // количество разбиений интервала
for( int i = 0; i < N; i++ )
{
    double x0 = OX[i], x1 = OX[i+1];
    double dh = (x1 - x0 ) / SplineSplit;
    for( int j = 0; j <= SplineSplit; j++ )
    {
        double x = x0 + j * dh;
        double y = GetSplineValue ( x );
        SplineGraphAdd ( x, y );
    }
}
```

Но в данном случае не учитываются длина интервала и скорость изменения функции. По идее, переменная `SplineSplit` должна быть задана для каждого интервала независимо.

Кроме того, уровень дискретизации должен зависеть и от масштаба: при мелком масштабе нет необходимости в подробном шаге дискретизации (лишние расчеты). Но вот при укрупнении масштаба начиная с какого-то момента гладкий график станет кусочно-линейным.

## ПРИМЕРЫ ВИЗУАЛИЗАЦИИ СПЛАЙНОВ

### ПРИМЕР №1

На рис.29-36 приведены результаты построения некоторых рассмотренных ранее сплайнов на "проблемной" функции  $f(x) = |x|$  с разным количеством узлов интерполяции.

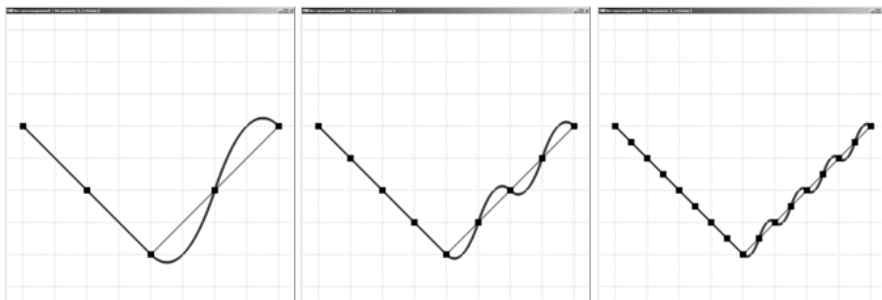


Рис.29. Двухточечный квадратичный интерполяционный сплайн.

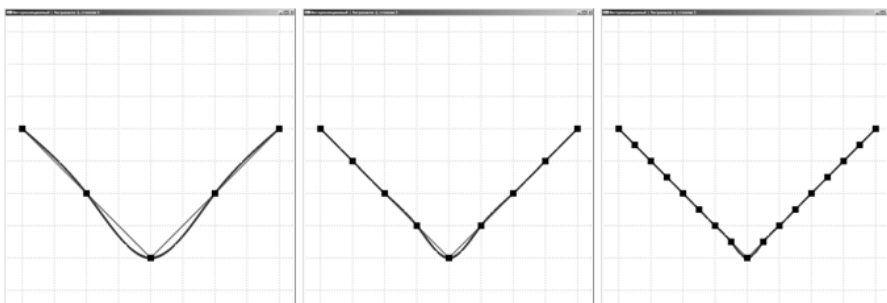


Рис.30. Двухточечный кубический интерполяционный сплайн.

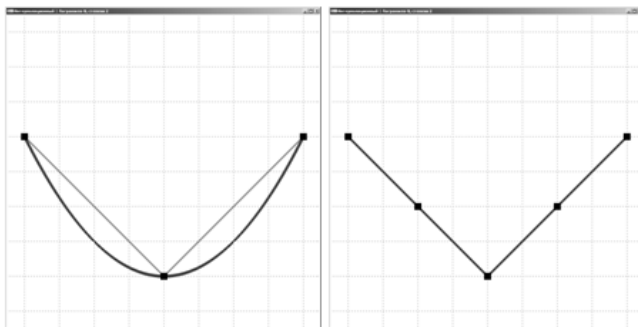


Рис.31. Многоточечный квадратичный интерполяционный сплайн.

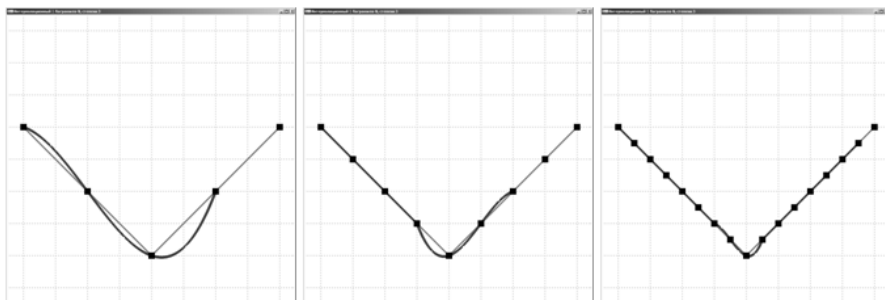


Рис.32. Многоточечный кубический интерполяционный сплайн.

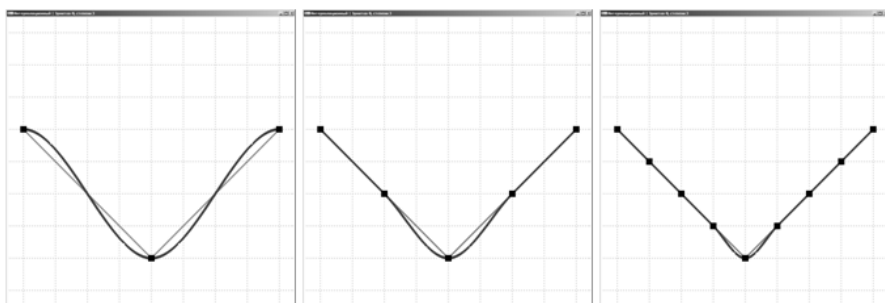


Рис.33. Эрмитов кубический интерполяционный сплайн.

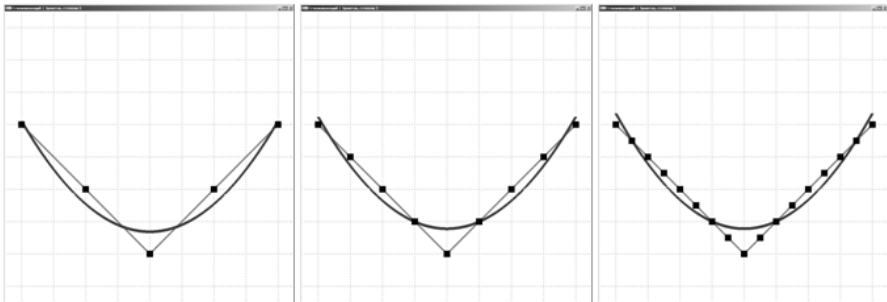


Рис.34. Эрмитов кубический сглаживающий сплайн на 1 интервале.

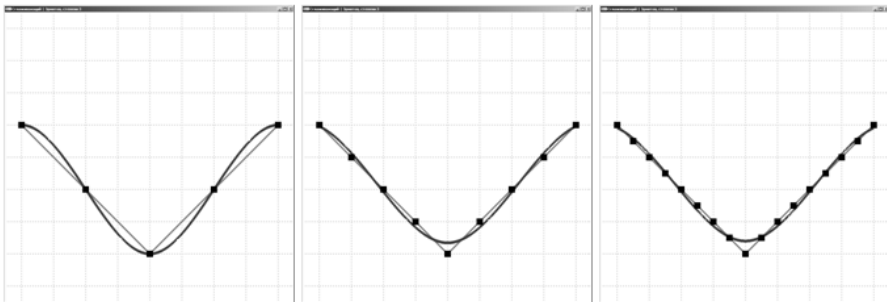


Рис.35. Эрмитов кубический сглаживающий сплайн на 2 интервалах.

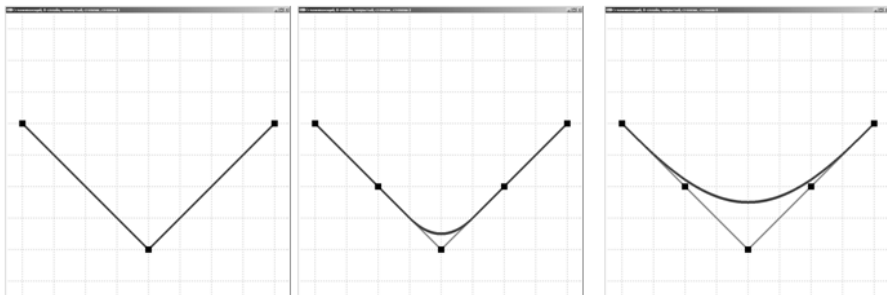


Рис.36. В-сплайн со степенями 1, 2 и 4.

## ПРИМЕР №2

На рис.37-44 приведены результаты построения некоторых рассмотренных ранее сплайнов на следующей "проблемной" функции с разным количеством узлов интерполяции:

$$f(x) = \frac{1}{1 + 25 \cdot x^2}.$$

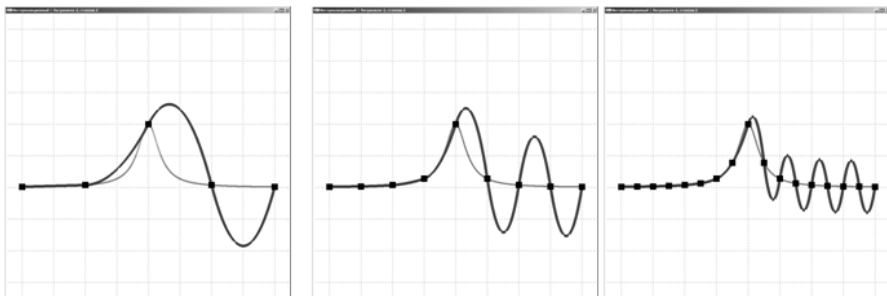


Рис.37. Двухточечный квадратичный интерполяционный сплайн.

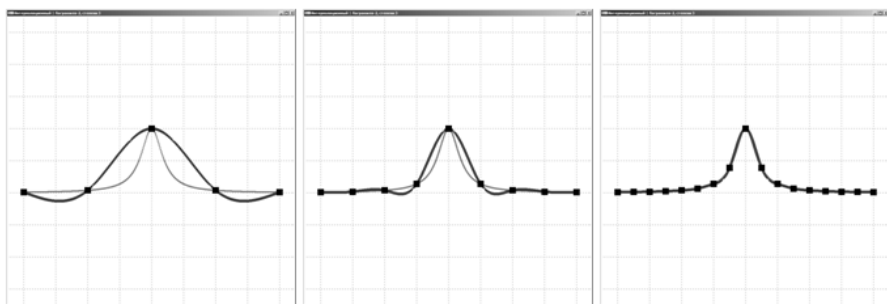


Рис.38. Двухточечный кубический интерполяционный сплайн.

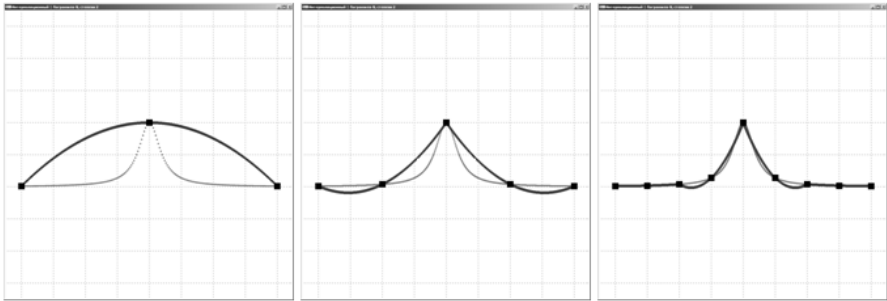


Рис.39. Многоточечный квадратичный интерполяционный сплайн.

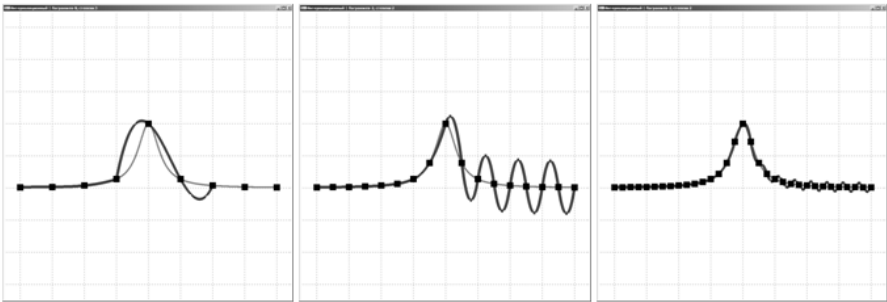


Рис.40. Многоточечный кубический интерполяционный сплайн.

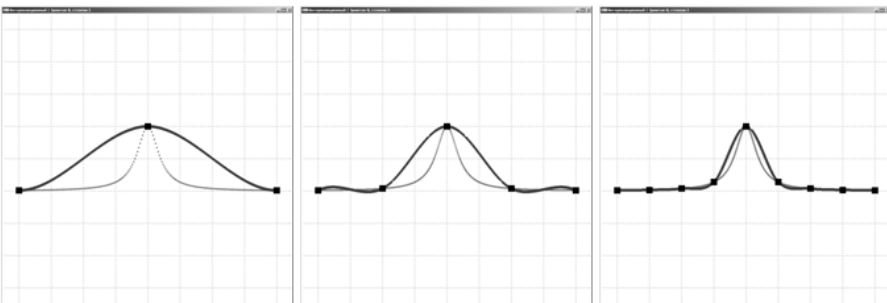


Рис.41. Эрмитов кубический интерполяционный сплайн.

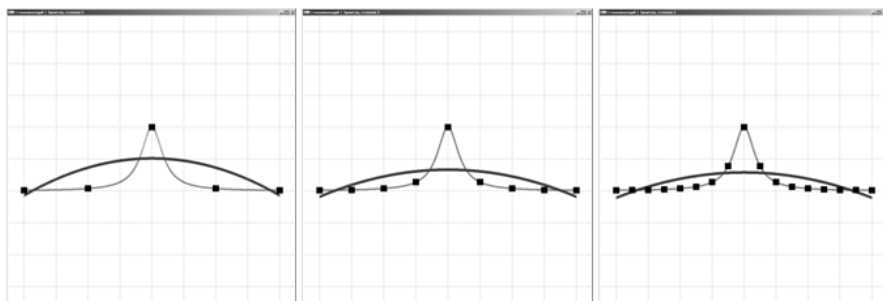


Рис.42. Эрмитов кубический сглаживающий сплайн на 1 интервале.

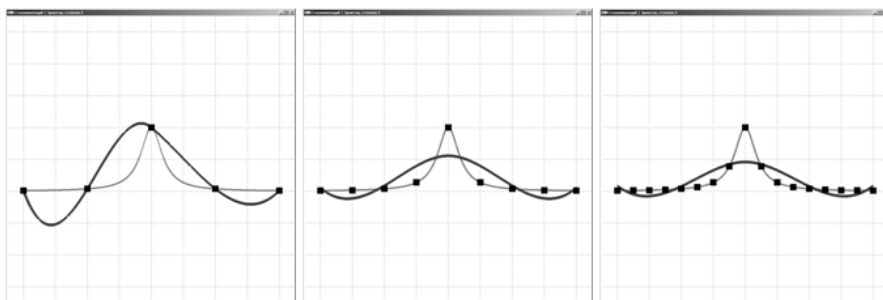


Рис.43. Эрмитов кубический сглаживающий сплайн на 2 интервалах.

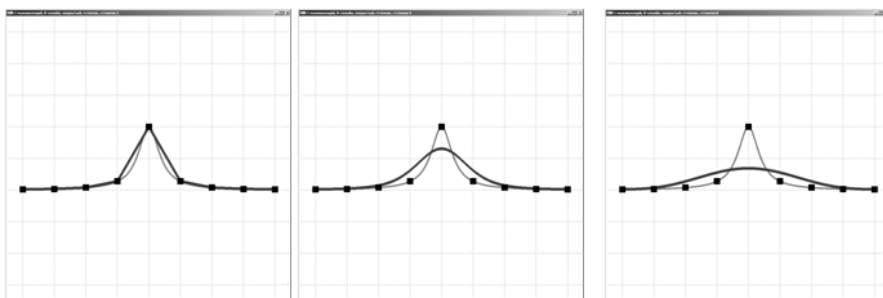


Рис.44. В-сплайн со степенями 1,4 и 8.



### ПРИМЕР №3

На рис.45-50 приведены результаты построения некоторых рассмотренных ранее сплайнов на функции, заданной лишь семью контрольными точками, из которых одна ( $P_2$ ) выделяется.

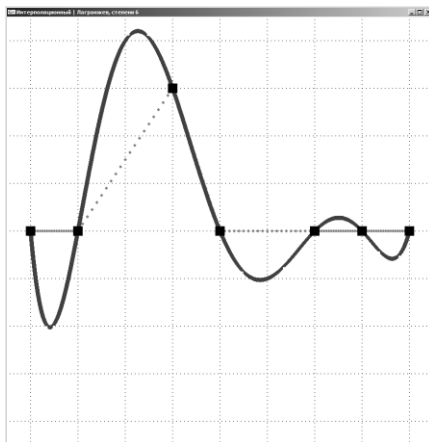


Рис.45. Сплайн Лагранжа.

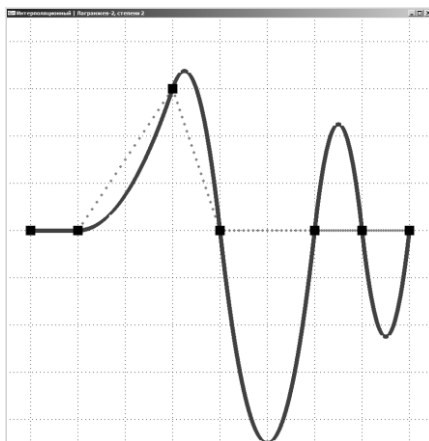


Рис.46. Двухточечный квадратичный интерполяционный сплайн.

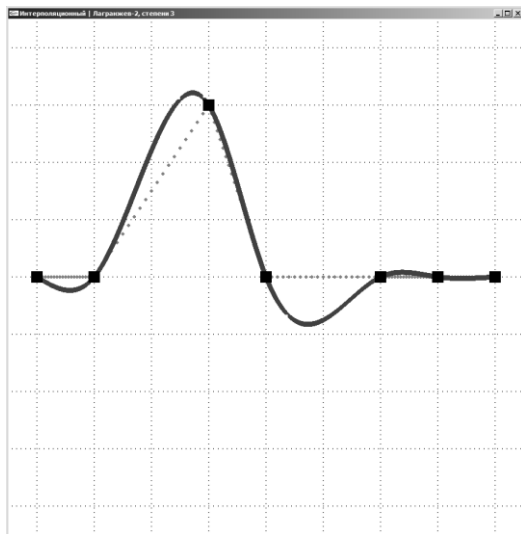


Рис.47. Двухточечный кубический интерполяционный сплайн.

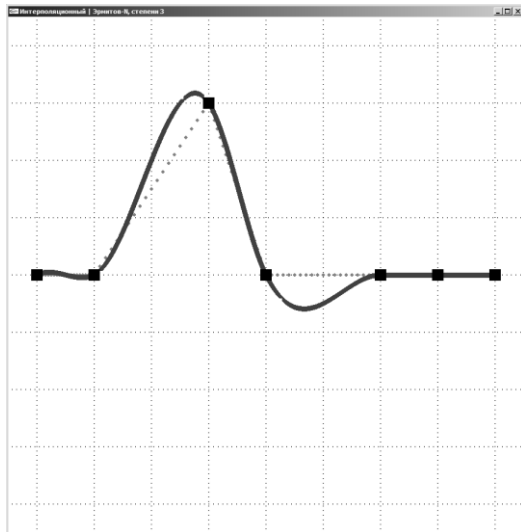


Рис.48. Эрмитов кубический интерполяционный сплайн.

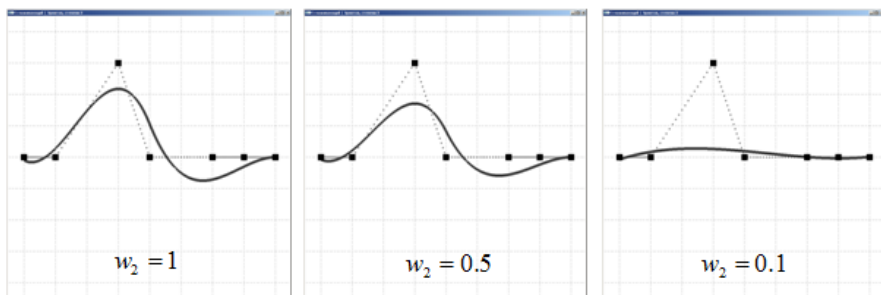


Рис.49. Эрмитов кубический сглаживающий сплайн с различными весами.

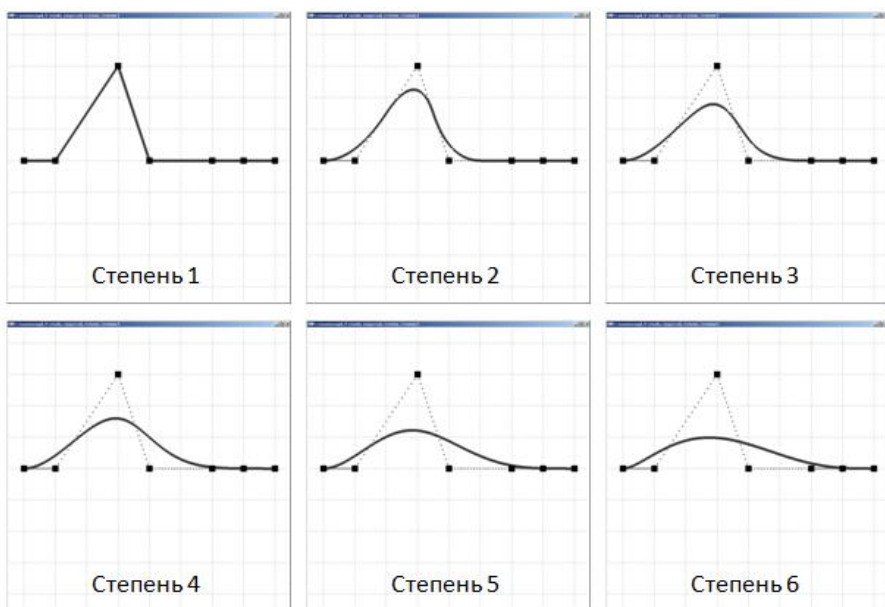


Рис.50. В-сплайн степени 1-6.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Задорожный А.Г.* Введение в двумерную компьютерную графику с использованием библиотеки OpenGL: учебное пособие / А. Г. Задорожный, Д. В. Вагин, Ю. И. Кошкина. – Новосибирск: Изд-во НГТУ, 2018. – 102 с. – Режим доступа: <https://elibrary.nstu.ru/source?id=73094>.
2. *Задорожный А.Г.* Введение в трехмерную компьютерную графику с использованием библиотеки OpenGL: учебное пособие / А. Г. Задорожный, М. Г. Персова, Ю. И. Кошкина. – Новосибирск: Изд-во НГТУ, 2018. – 99 с. – Режим доступа: <https://elibrary.nstu.ru/source?id=75699>.
3. *Соловейчик Ю. Г.* Метод конечных элементов для решения скалярных и векторных задач: учебное пособие / Ю. Г. Соловейчик, М. Э. Рояк, М. Г. Персова. – Новосибирск: Сер. «Учебники НГТУ», 2007. – 896 с.
4. *Роджерс Д.* Математические основы машинной графики / Д. Роджерс, Дж. Адамс. – М.: Мир, 2001. – 604 с.
5. *Шикин Е.В.* Кривые и поверхности на экране компьютера. Руководство по сплайнам для пользователя / Е.В. Шикин, А.И. Плис. – М.: Диалог-МИФИ, 1996. – 237с.
6. *Алберг, Дж.* Теория сплайнов и ее приложения / Дж. Алберг, Э Нильсон, Дж. Уолш. – М.: Мир, 1972. – 319 с.
7. *Стечкин С. Б.* Сплайны в вычислительной математике / С.Б. Стечкин, Ю.Н. Субботин. –М.: Наука, 1976. – 247с.
8. *Завьялов Ю.С.* Сплайны в инженерной геометрии / Ю.С. Завьялов, В.А. Леус, В.А. Скороспелов. – М.: Машиностроение, 1980. – 224 с.
9. *К. де Бор.* Практическое руководство по сплайнам / К. де Бор. – М.: Радио и связь, 1985. – 303 с.
10. *Мысовских И. П.* Лекции по численным методам. Изд. второе. / И.П. Мысовских. – СПб.: Издво СПбГУ, 1998. 472 с.
11. *Залогова Л.А.* Компьютерная графика / Л. Залогова. – М., 2005. – 319 с.
12. *Дегтярев В.М.* Инженерная и компьютерная графика: учебник / В.М. Дегтярев, В.П. Затыльников. – М., 2010. – 238 с.

## СОДЕРЖАНИЕ

ОБЩИЕ СВЕДЕНИЯ .....	3
Многочлены .....	5
Бином Ньютона .....	6
Алгебраические полиномы .....	7
Многочлены Чебышева .....	8
Многочлен Тейлора .....	9
Многочлены Бернштейна .....	10
Понятия сплайнов .....	11
Задачи сплайнов .....	11
Основные определения .....	12
Классификация сплайнов .....	13
По мерности пространства .....	13
По назначению .....	14
По локальности .....	15
По типу краевых условий .....	16
По ширине фрагментов .....	17
По количеству фрагментов .....	17
По виду фрагментов .....	18
По состыковке фрагментов .....	19
По параметризации .....	19
Интерполяционный многочлен .....	20
Интерполяционные формулы Ньютона .....	21
О естественных кубических сплайнах .....	22
Феномен Рунге .....	23
Параметризация сплайна .....	24

Функции, заданные параметрически .....	24
Построение параметрического сплайна .....	25
Пример параметризации для линейного сплайна .....	26
ЛАГРАНЖЕВЫ СПЛАЙНЫ .....	27
Интерполяционный полином Лагранжа .....	28
Двухточечная интерполяция .....	29
Кусочно-линейная .....	30
Кусочно-квадратичная .....	31
Кусочно-кубическая .....	32
Многоточечная интерполяция .....	34
Кусочно-линейная .....	35
Кусочно-квадратичная .....	37
Кусочно-кубическая .....	38
ЭРМИТОВЫ СПЛАЙНЫ .....	39
Эрмитовы базисные функции .....	41
Кубическая интерполяция .....	43
Кубическое сглаживание .....	45
БАЗИСНЫЕ СПЛАЙНЫ .....	48
Сплайны Безье .....	49
Основные виды кривых Безье .....	50
Некоторые свойства .....	51
Особенности расчета базисных функций .....	52
Геометрическая интерпретация .....	53
Поверхность Безье .....	55
В-сплайны .....	56
Вектор параметров .....	57
Некоторые свойства .....	59

NURBS-сплайны .....	60
Некоторые свойства.....	61
Примеры визуализации .....	61
Реализация в библиотеке OpenGL .....	63
Функции включения вычислителя.....	63
Функции расчета параметров сплайна .....	64
Функции визуализации сплайна.....	66
Пример задания кривой Безье .....	67
Реализация в библиотеке GLU .....	68
Функции задания параметров сплайна.....	68
Функции задания сплайна.....	69
Пример задания NURBS-кривой.....	70
Пример задания NURBS-поверхности .....	70
ВИЗУАЛИЗАЦИЯ СПЛАЙНОВ.....	71
Вывод графика на экран.....	72
Расчет координат графика .....	73
Примеры визуализации сплайнов .....	75
Пример №1 .....	75
Пример №2 .....	78
Пример №3 .....	81
БИБЛИОГРАФИЧЕСКИЙ СПИСОК .....	84

**Задорожный Александр Геннадьевич  
Киселев Дмитрий Сергеевич**

## **ПОСТРОЕНИЕ СПЛАЙНОВ С ИСПОЛЬЗОВАНИЕМ БИБЛИОТЕКИ OPENGL**

**Учебное пособие**

*В авторской редакции*

Выпускающий редактор И.П. Брованова  
Дизайн обложки А.В. Ладыжская

Налоговая льгота – Общероссийский классификатор продукции  
Издание соответствует коду 95 3000 ОК 005-93 (ОКП)

---

Подписано в печать 13.12.2019. Формат 60 × 84 1/16. Бумага офсетная.  
Тираж 100 экз. Уч.-изд. л. 5,11. Печ. л. 5,5. Изд. № 358/18. Заказ № 163  
Цена договорная

---

Отпечатано в типографии  
Новосибирского государственного технического университета  
630073, г. Новосибирск, пр. К. Маркса, 20