

УДК 519.6

URL: <https://ptsj.ru/catalog/math/compmath/964.html>

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМОВ ПОСТРОЕНИЯ И ЛОКАЛЬНОГО УТОЧНЕНИЯ Т-СПЛАЙНОВ

Д.Ю. Федотов

fedotov_clever_91@mail.ru

МГТУ им. Н.Э. Баумана, Москва, Российская Федерация

Представлены результаты разработки и последующей реализации алгоритмов построения и локального уточнения Т-сплайнов, основанные на теории Т-сплайнов. Проведен обзор NURBS и обобщающих их Т-сплайнов, рассмотрен описывающий их математический аппарат и приведено математическое обоснование алгоритмов их построения и локального уточнения путем вставки новых точек с целью улучшения локальной области без потери точности. Разработана программная реализация на языке программирования C++ с визуализацией результатов средствами OpenGL, приведены примеры работы рассмотренных алгоритмов: на заданной сетке успешно построен Т-сплайн и выполнена вставка нескольких точек в разные области без ухудшения точности построенной изначально криволинейной поверхности.

EDN: XXRBVHY**Ключевые слова:** сплайн, NURBS, Т-сплайн, алгоритм, геометрическое моделирование, компьютерная геометрия

Введение. Постепенное внедрение компьютерных технологий во все сферы нашей жизни существенно расширило возможности путем частичной или полной автоматизации труда, способствующей сокращению затрат и увеличению производительности. По заказу предприятий аэрокосмической и автомобильной промышленности изучением принципов моделирования трехмерных объектов занялись различные исследовательские группы, что и привело в итоге к созданию сплайнов NURBS (Non-Uniform Rational B-Splines — неравномерные рациональные В-сплайны), используемых по всему миру в коммерческих САД-системах и большинстве пакетов программ для компьютерной графики, например, Maya и Autodesk 3ds Max.

Т-сплайны — логическое развитие NURBS-сплайнов. Являясь обобщением В-сплайн поверхностей, сплайн данного вида обладает гораздо большей гибкостью и решает одну из базовых проблем «взрывной» генерации данных (появления большого объема излишних данных) при слиянии В-сплайновых поверхностей [1–3].

В-сплайны и NURBS. В-сплайны представляют собой обобщение кривых и поверхностей Безье: они также позволяют конструктивно задавать

форму кривой с помощью контрольных точек, однако их алгебраическая степень не зависит от числа заданных точек. Уравнение сплайна имеет вид, аналогичный кривой Безье, но сопрягающие функции не являются многочленами Бернштейна, а определяются рекурсивно в зависимости от параметра. Область задания параметра разбита на узлы, соответствующие точкам сопряжения алгебраических кривых заданной степени.

Общая формула кривой имеет вид

$$r(t) = \sum_{i=0}^n N_i^k(t) P_i, \quad t_{\min} \leq t \leq t_{\max}, \quad 0 \leq k \leq n,$$

где P_i — координаты i -й контрольной вершины; $N_i^k(t)$ — базисные функции, определяемые с помощью рекурсивной формулы Кокса — де Бура:

$$N_i^0(t) = \begin{cases} 1, & t_i \leq t < t_{i+1}, \\ 0 & \text{иначе;} \end{cases}$$

$$N_i^k(t) = \frac{t - t_i}{t_{i+k} - t_i} N_i^{k-1}(t) + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} N_{i+1}^{k-1}(t).$$

Узлы t_i — элементы узлового вектора, удовлетворяющие условию $t_i \leq t_{i+1}$, а сам параметр изменяется в пределах $[t_{\min}; t_{\max}]$.

Рассмотрим понятие рациональных В-сплайнов, общий вид которых может быть записан следующим образом:

$$r(t) = \frac{\sum_{i=0}^n w_i P_i N_i^k(t)}{\sum_{j=0}^n w_j N_j^k(t)} = \sum_{i=0}^n P_i R_i^k(t),$$

где w_i — весовые коэффициенты; $R_i^k(t) = \frac{w_i N_i^k(t)}{\sum_{j=0}^n w_j N_j^k(t)}$ — базисные функции

рационального В-сплайна.

Рациональные В-сплайны являются обобщением нерациональных, когда не все весовые коэффициенты равны единице. При $w_i \geq 0$ они наследуют почти полный набор геометрических и аналитических свойств В-сплайнов. При использовании в определении неравномерного узлового вектора мы получим так называемые NURBS-сплайны.

NURBS (Non-Uniform Rational B-Splines) — рациональные В-сплайны, допускающие наличие неравномерного узлового вектора (рис. 1). Они получили широкое распространение благодаря своей стандартизованности и относительной простоте и служат неотъемлемой частью большинства пакетов программ для компьютерной графики.

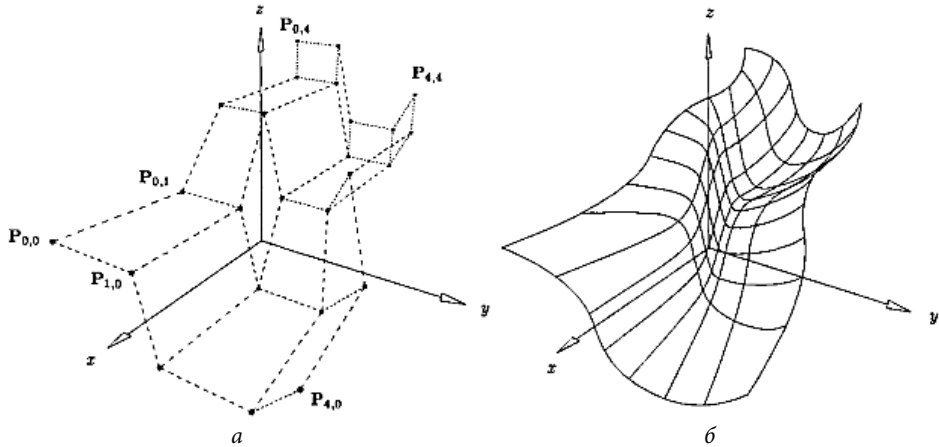


Рис. 1. Пример NURBS-поверхности:

а — контрольная сетка; б — биквадратичная NURBS-поверхность

NURBS-поверхность — это параметрическая поверхность, являющаяся тензорным произведением двух одномерных NURBS-сплайнов степени p и q :

$$r(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m w_{ij} P_{ij} N_i^p(u) N_j^q(v)}{\sum_{k=0}^n \sum_{l=0}^m w_{kl} N_k^p(u) N_l^q(v)}, \quad 0 \leq u, v \leq 1,$$

где $P_{ij} = \{x_{ij}, y_{ij}, z_{ij}\}$ — узел контрольной сетки; w_{ij} — соответствующий весовой коэффициент; $U = \{0, \dots, 0, u_{p+1}, \dots, u_{r-p-1}, 1, \dots, 1\}$, $r = n + p + 1$ — узловой вектор параметра u ; $V = \{0, \dots, 0, v_{q+1}, \dots, v_{s-q-1}, 1, \dots, 1\}$, $s = m + q + 1$ — узловой вектор параметра v .

Базисной функцией поверхности будет являться

$$R_{ij}^{pq}(u, v) = \frac{w_{ij} N_i^p(u) N_j^q(v)}{\sum_{k=0}^n \sum_{l=0}^m w_{kl} N_k^p(u) N_l^q(v)}. \tag{1}$$

С учетом (1) уравнение NURBS-поверхности может быть переписано в следующем виде [4–6]

$$r(u, v) = \sum_{i=0}^n \sum_{j=0}^m R_{ij}^{pq}(u, v) P_{ij}.$$

Т-сплайны. Т-сплайны были впервые широко представлены Томасом Седербергом в [7]. Они являются видом поверхности свободной формы, по аналогии с NURBS-поверхностями, сетка которой допускает наличие Т-соединений — узлов, соседствующих только с тремя точками, а не четырьмя, как в привычной нам прямоугольной сетке. Хотя понятие Т-сплайнов определено для любой степени, для наглядности в данной работе ограничимся случаем базисных функций третьей степени, подробнее данный аспект разобран в [8].

Цель при построении Т-сплайна — сохранение большей части свободы РВ-сплайнов (представленных в [7]) при восстановлении структуры, подобной контрольной сетке NURBS, называемой Т-сеткой. Т-сетка предлагает более удобное взаимодействие с контрольными точками путем операций вставки и удаления, а также именно из нее выводятся узловые векторы \mathbf{s}_i и \mathbf{t}_i , узлы которых представляют собой точки в параметрическом пространстве (s, t) . При вырождении Т-сетки в прямоугольную Т-сплайн преобразуется в обычный В-сплайн.

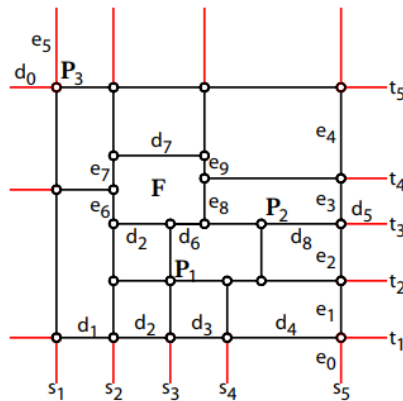


Рис. 2. Т-сетка сплайна

Рассмотрим обозначения (рис. 2), используемые при выводе узловых интервалов из Т-сетки в параметрическом пространстве (s, t) ; (s_i, t_j) — параметрические координаты точки; (e_i, d_j) — длины интервалов вдоль соответствующих переменных. К примеру, $s_4 = s_3 + d_3$, $t_5 = t_4 + e_4$. Красными на рис. 2

обозначены ребра, содержащие интервалы с граничными условиями (за них принимают длины внешних интервалов), при их отсутствии граничные условия принимают нулевыми.

Ребрам Т-сетки соответствуют интервалы, на которые наложены следующие ограничения.

Правило 1. Суммы узловых интервалов на противоположных ребрах любого участка поверхности должны быть равны. К примеру, на рис. 2 для F должно выполняться: $d_2 + d_6 = d_7$ и $e_6 + e_7 = e_8 + e_9$.

Правило 2. Если Т-узлы, расположенные на противоположных гранях, могут быть соединены (тем самым разбивая участок на две части) без нарушения 1-го правила — новое ребро должно быть встроено в сетку.

Каждой точке P_i соответствует базисная функция $B_i(s, t)$, построенная на узловых векторах $\mathbf{s}_i = \{s_{i0}, s_{i1}, s_{i2}, s_{i3}, s_{i4}\}$ и $\mathbf{t}_i = \{t_{i0}, t_{i1}, t_{i2}, t_{i3}, t_{i4}\}$. Вывод данных векторов производится следующим образом. Параметрические координаты точки в данных обозначениях равны (s_{i2}, t_{i2}) . Поиск оставшихся значений происходит путем построения пересечений с ближайшими ребрами лучей $S(\alpha) = (s_{i2} + \alpha, t_{i2})$ и $T(\beta) = (s_{i2}, t_{i2} + \beta)$. К примеру, для точки $P_1(s_3, t_2)$ на рис. 2 узловые вектора будут иметь вид

$$\begin{aligned} \mathbf{s}_1 &= \{s_1, s_2, s_3, s_4, s_5 - d_8\}; \\ \mathbf{t}_1 &= \{t_1 - e_0, t_1, t_2, t_3, t_4 + e_9\}. \end{aligned}$$

В общем случае каждая точка обладает определенным весом $w_i \in R$ и, следовательно, формула стыковочной функции Т-сплайна, определенной в точке P_i имеет вид

$$R_i(s, t) = \frac{w_i B_i(s, t)}{\sum_{j=0}^n w_j B_j(s, t)}, \tag{2}$$

где $B_i(s, t) = N_{i0}^3(s)N_{i0}^3(t)$ — стыковочная функция РВ-сплайна (2), построенная на соответствующих P_i узловых векторах \mathbf{s}_i и \mathbf{t}_i ; $\{P_0, \dots, P_n\}$ — полный набор контрольных точек сплайна.

Итоговая формула взвешенного Т-сплайна выглядит следующим образом:

$$r(s, t) = \sum_{i=0}^n P_i R_i(s, t) = \frac{\sum_{i=0}^n P_i w_i B_i(s, t)}{\sum_{j=0}^n w_j B_j(s, t)}. \tag{3}$$

Уточнение стыковочных функций. Процедура уточнения стыковочных функций (Blending Function Refinement) играет важную роль при вставке и удалении контрольных точек.

Пусть дан узловой вектор $\mathbf{s} = \{s_0, s_1, s_2, s_3, s_4\}$ и вектор $\tilde{\mathbf{s}}$ размерности m , содержащий \mathbf{s} в качестве подвектора. В таком случае базисная функция $N(s) = N_{\{s_0, s_1, s_2, s_3, s_4\}}(s)$ может быть представлена линейной комбинацией $m-4$ В-сплайновых базисных функций, заданных на подвекторах $\tilde{\mathbf{s}}$.

Для простоты будем рассматривать случай вставки одного узла k , при котором размерность финального вектора $\tilde{\mathbf{s}}$ равняется $m=6$. Уравнения для случаев $m > 6$ выводятся по аналогии.

Если $\tilde{\mathbf{s}} = \{s_0, k, s_1, s_2, s_3, s_4\}$, то

$$N(s) = c_0 N_{\{s_0, k, s_1, s_2, s_3\}}(s) + d_0 N_{\{k, s_1, s_2, s_3, s_4\}}(s), \quad (4)$$

где $c_0 = \frac{k-s_0}{s_3-s_0}$, $d_0 = 1$.

Если $\tilde{\mathbf{s}} = \{s_0, s_1, k, s_2, s_3, s_4\}$, то

$$N(s) = c_1 N_{\{s_0, s_1, k, s_2, s_3\}}(s) + d_1 N_{\{s_1, k, s_2, s_3, s_4\}}(s), \quad (5)$$

где $c_1 = \frac{k-s_0}{s_3-s_0}$, $d_1 = \frac{s_4-k}{s_4-s_1}$.

Если $\tilde{\mathbf{s}} = \{s_0, s_1, s_2, k, s_3, s_4\}$, то

$$N(s) = c_2 N_{\{s_0, s_1, s_2, k, s_3\}}(s) + d_2 N_{\{s_1, s_2, k, s_3, s_4\}}(s), \quad (6)$$

где $c_2 = \frac{k-s_0}{s_3-s_0}$, $d_2 = \frac{s_4-k}{s_4-s_1}$.

Если $\tilde{\mathbf{s}} = \{s_0, s_1, s_2, s_3, k, s_4\}$, то

$$N(s) = c_3 N_{\{s_0, s_1, s_2, s_3, k\}}(s) + d_3 N_{\{s_1, s_2, s_3, k, s_4\}}(s), \quad (7)$$

где $c_3 = 1$, $d_3 = \frac{s_4-k}{s_4-s_1}$.

Если же $k \leq s_0$ или $k \geq s_4$, то $N(s)$ остается без изменений.

Стыковочная функция $B_i(s, t)$ может быть подвергнута вставке узла по обоим направлениям, тем самым расщепляясь на две новые функции. Последующая вставка узлов дает целый набор функций, в сумме дающих изначальную (рис. 3):

$$B_1(s, t) = c_1^1 \tilde{B}_1(s, t) + c_1^2 \tilde{B}_2(s, t) + c_1^3 \tilde{B}_3(s, t) + c_1^4 \tilde{B}_4(s, t). \quad (8)$$

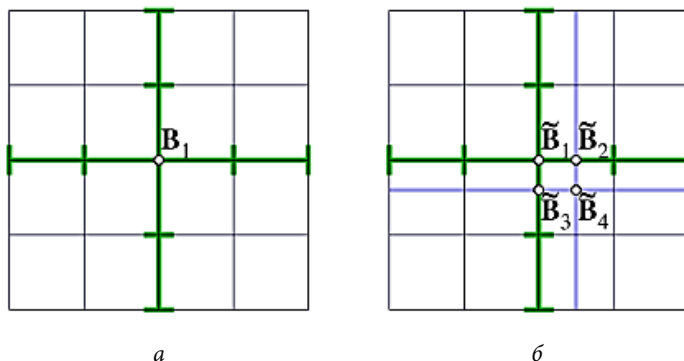


Рис. 3. Пример уточнения функции $B_1(s, t)$:

a — начальная стыковочная функция; *б* — уточняющие стыковочные функции

Алгоритм локального улучшения. Процесс локального улучшения (Local Refinement Algorithm) состоит в произведении вставки некоторого количества контрольных точек в Т-сетку без изменения формы самой поверхности. Его также называют локальной вставкой узла, поскольку при вставке дополнительных узлов в Т-сетку исходные контрольные точки влияют на построение стыковочных функций.

Алгоритм состоит из двух этапов:

- 1) топологического, на котором определяется необходимый набор дополнительных точек для вставки;
- 2) геометрического, на котором вычисляются координаты и веса обновленной Т-сетки.

В Т-сплайне существует тесная связь между стыковочными функциями и Т-сеткой: каждой контрольной точке соответствует индивидуальная стыковочная функция, для которой определен узловой вектор, удовлетворяющий правилу 1.

Предположим, что могут существовать стыковочные функции, нарушающие данное правило. Возможны три варианта развития событий:

Нарушение 1. В соответствующем функции узловом векторе отсутствует узел, удовлетворяющий правилу 1 для данной Т-сетки.

Нарушение 2. В соответствующем функции узловом векторе присутствует узел, не удовлетворяющий правилу 1 для данной Т-сетки.

Нарушение 3. Контрольная точка не имеет стыковочных функций.

Т-сплайн построен корректно, если ни одно из нарушений не обнаружено. В противном случае алгоритм будет последовательно исправлять нарушения, пока они не закончатся.

Топологический этап алгоритма локального улучшения состоит из следующего набора инструкций.

1. Вставить все нужные контрольные точки в T-сетку.
2. Если существуют стыковочные функции, удовлетворяющие условиям нарушения 1, произвести необходимые вставки узлов в данные функции согласно (4)–(8).
3. Если существуют стыковочные функции, удовлетворяющие условиям нарушения 2, добавить соответствующую контрольную точку в T-сетку.
4. Повторять шаги 2 и 3, пока нарушений не останется.

Заметим, что решение всех случаев нарушений 1 и 2 автоматически решает все случаи нарушения 3 [9, 10].

Реализация и результаты. Рассмотрим базовые структуры данных и классы, необходимые для программной реализации T-сплайнов. Затем рассмотрим примеры некоторых поверхностей, полученных с помощью описанного алгоритма.

Согласно определению T-сплайна [7, 8], его основой является T-сетка. Она представляет собой массив вершин и ребер, их соединяющих (рис. 4). Для корректного задания на ребра сетки накладываются определенные ограничения (правила 1 и 2).

В качестве базовых структур данных рассмотрим вершину, ребро и саму T-сплайновую поверхность.

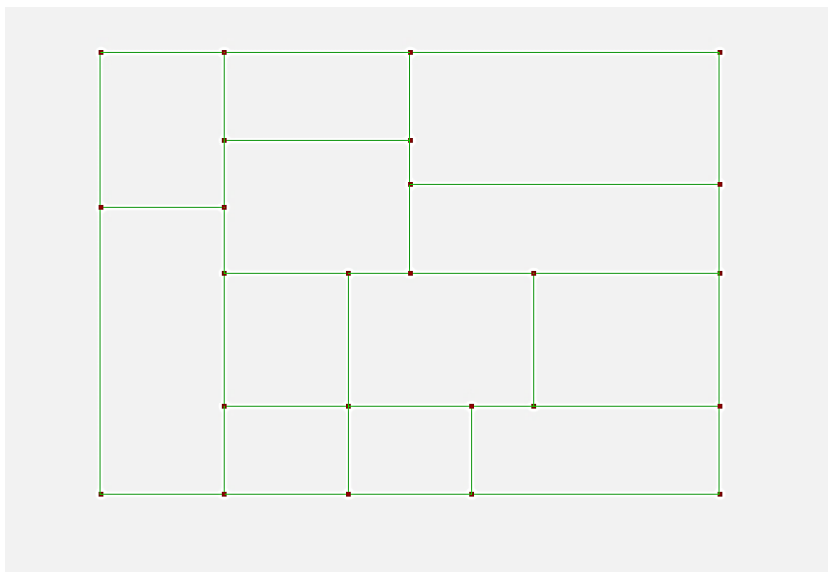


Рис. 4. Тестовая T-сетка

Вершина помимо базовых координат в трехмерном пространстве должна содержать в себе узловые векторы s_i и t_i по каждому параметру. В рассматриваемом случае это реализуется путем хранения в вершине информации о соседних ребрах. Ребро содержит в себе лишь информацию о начальной и конечной вершинах. Т-сплайновая поверхность состоит из массивов вершин и ребер, на базе которых она строится.

Расчет значений Т-сплайна проводится по формуле (3). Учтем, что на текущий момент все веса контрольных точек равны единице (рис. 5).

Выполним теперь непосредственно локальное улучшение данного сплайна путем вставки некоторого числа новых вершин и ребер согласно обозначенному ранее алгоритму. Улучшение (т. е. непосредственная вставка новых узлов, ребер и замена стыковочных функций) должно осуществляться таким образом, чтобы сам сплайн не менялся.

Рассмотрим новую Т-сетку, полученную после вставки всех дополнительных узлов (рис. 6). Заметим, что данная сетка изображена именно в параметрическом пространстве и в декартовой системе координат точки могут быть смещены из-за процедуры уточнения стыковочных функций (рис. 7).

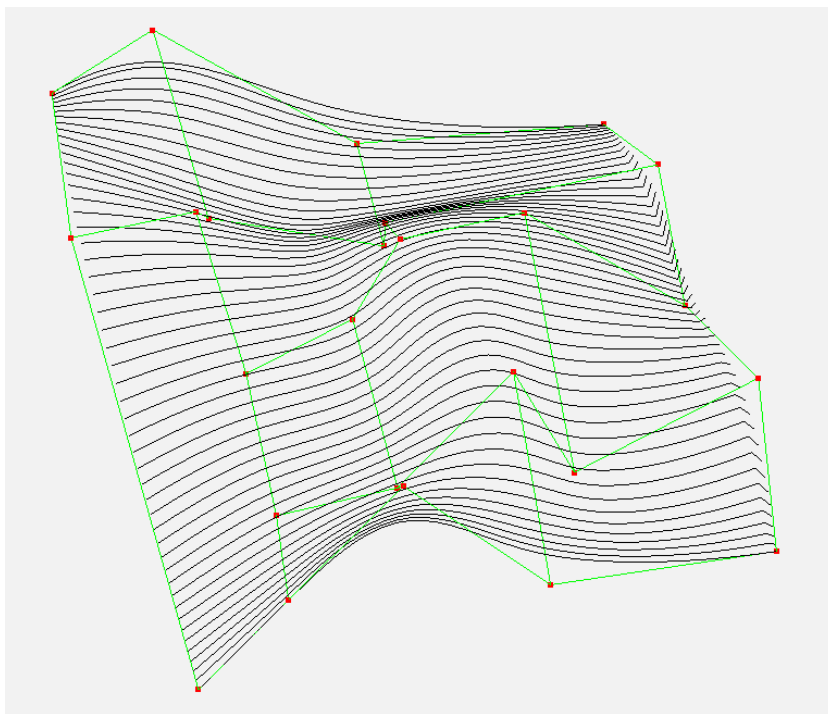


Рис. 5. Т-сплайн, построенный на тестовой сетке

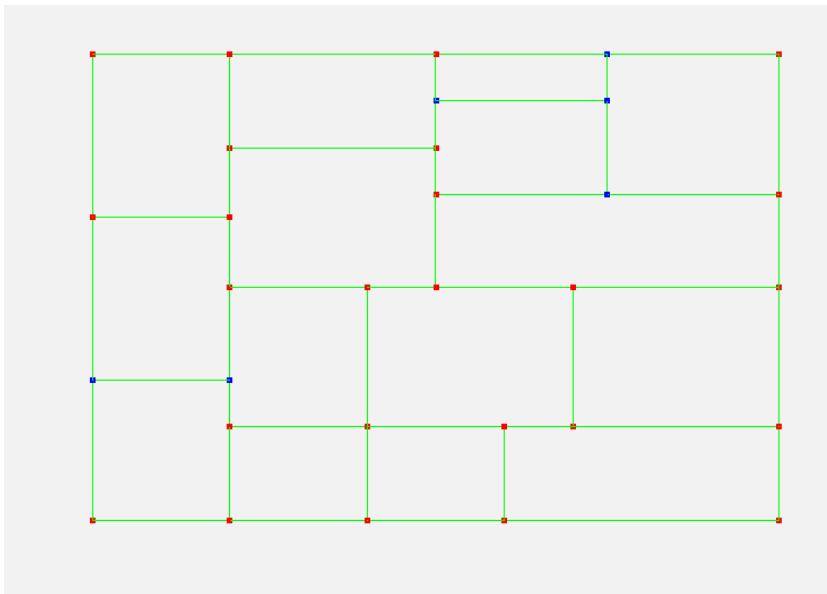


Рис. 6. Дополненная Т-сетка (новые узлы обозначены синим цветом)

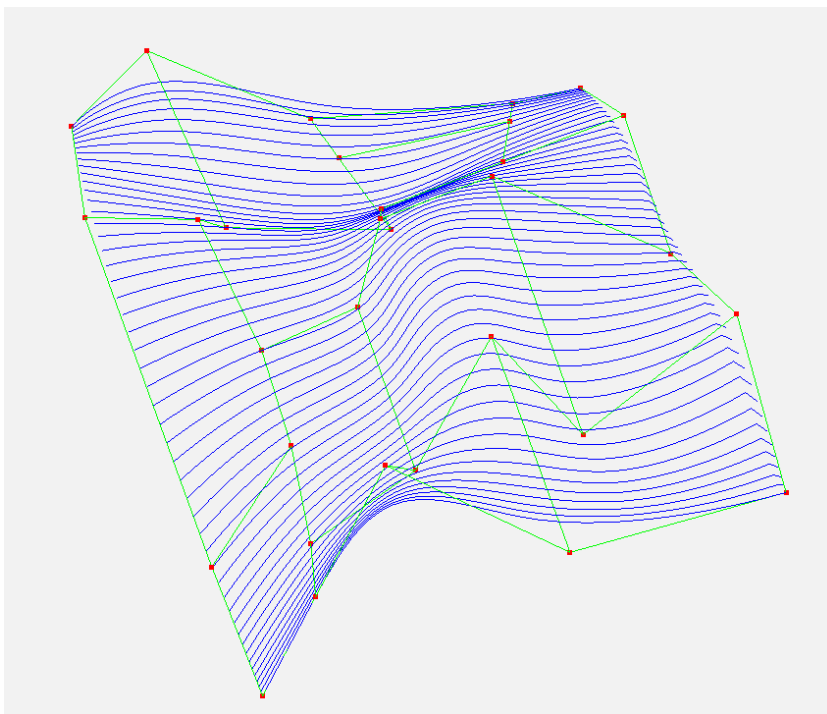


Рис. 7. Уточненный Т-сплайн

Заключение. В данной работе рассмотрены поверхности свободной формы Т-сплайн, их определение и алгоритм построения. Помимо этого также представлены его формула расчета и алгоритм локального уточнения сетки, в дальнейшем реализованные средствами языка программирования С++, продемонстрированы результаты работы написанных алгоритмов (см. рис. 5, 7).

В качестве дальнейших целей исследования предлагается ряд задач: рассмотрение алгоритма упрощения Т-сплайна путем вставки/удаления узлов с последующим локальным уточнением, применение метода наименьших квадратов для приближения Т-сплайном заданной поверхности, разработка методов парсинга форматов файлов, использующихся для хранения данных о Т-сплайне.

Литература

- [1] Задорожный А.Г., Киселев Д.С. *Построение сплайнов с использованием библиотеки OpenGL*. Новосибирск, НГТУ, 2019, 88 с.
- [2] Завьялов Ю.С., Квасов Б.И., Мирошниченко В.Л. *Методы сплайн-функций*. Москва, Наука, 1980, 352 с.
- [3] Бакенов А.С. Т-сплайны: геометрическая гибкость и возможность локального изменения представления поверхности. *7-я Междунар. конф. по компьютерной графике и машинному зрению: сб. тр.*, Пермь, Пермский государственный национальный исследовательский университет, 2017, с. 328–331. EDN: OUROEF
- [4] Роджерс Д., Адамс Дж. *Математические основы машинной графики*. Москва, Мир, 2001, 604 с.
- [5] Les Piegl, Tiller W. *The NURBS Book*. New York, Springer, 1996, 646 p.
- [6] Голованов Н.Н. *Геометрическое моделирование*. Москва, ИНФРА-М. Курс, 2016, 400 с.
- [7] Sederberg T.W., Zheng J., Bakenov A., Nasri A. T-Splines and T-NURCCs. *ACM Transactions on Graphics*, 2003, vol. 22, pp. 477–484.
- [8] Bazilevs Y., Calo V.M., Cottrell J.A. et al. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering*, 2010, vol. 99, pp. 229–263. <http://doi.org/10.1016/j.cma.2009.02.036>
- [9] Sederberg T.W., Cardon D.L., Finnigan G.T. et al. T-spline Simplification and Local Refinement. *ACM Transactions on Graphics*, 2004, vol. 23, pp. 276–283.
- [10] Cardon D.L. *T-Spline Simplification*. Th. Diss. Brigham Young University, 2007, 114 p.

Поступила в редакцию 30.12.2023

Федотов Даниил Юрьевич — студент кафедры «Вычислительная математика и математическая физика», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Научный руководитель — Захаров Андрей Алексеевич, кандидат физико-математических наук, доцент кафедры «Вычислительная математика и математическая физика», МГТУ им. Н.Э. Баумана, Москва, Российская Федерация.

Ссылку на эту статью просим оформлять следующим образом:

Федотов Д.Ю. Программная реализация алгоритмов построения и локального уточнения T-сплайнов. *Политехнический молодежный журнал*, 2024, № 01 (90). URL: <https://ptsj.ru/catalog/math/compmath/964.html>

SOFTWARE IMPLEMENTATION OF ALGORITHMS FOR THE T-SPLINE CONSTRUCTION AND LOCAL REFINEMENT

D.Y. Fedotov

fedotov_clever_91@mail.ru

Bauman Moscow State Technical University, Moscow, Russian Federation

The paper presents results of development and subsequent implementation of algorithms for the T-splines construction and local refinement based on the T-spline theory. NURBS and T-splines generalizing them were reviewed, mathematical apparatus describing them was analyzed, and mathematical justification was provided for their construction and local refinement algorithms by inserting the new points to improve the local area without losing accuracy. Software implementation was developed in the C++ programming language with results visualization using the OpenGL package; examples of the considered algorithms operation were provided. A T-spline was successfully constructed on the given mesh, and several points were inserted into different areas without compromising accuracy of the initially constructed curved surface.

EDN: XXRBHY

Keywords: Spline, NURBS, T-spline, algorithm, geometric simulation, computer geometry

References

- [1] Zadorozhnyy A.G., Kiselev D.S. *Postroenie splaynov c ispol'zovaniem biblioteki OpenGL*. Novosibirsk, NGTU, 2019, 88 p. (In Russ.).
- [2] Zav'yalov Yu.S., Kvasov B.I., Miroshnichenko V.L. *Metody splayn-funktsiy*. Moscow, Nauka, 1980, 352 p. (In Russ.).
- [3] Bakenov A.S. T-splines: geometrical flexibility and local modification. *Geometric Modeling. Computer Graphics in Education. GraphiCon*, 2017, pp. 328–331. (In Russ.). EDN: OUROEF
- [4] Rodzhers D., Adams Dzh. *Matematicheskie osnovy mashinnoy grafiki*. Moscow, Mir, 2001, 604 p. (In Russ.).
- [5] Les Piegl, Tiller W. *The NURBS Book*. New York, Springer, 1996, 646 p.
- [6] Golovanov N.N. *Geometricheskoe modelirovanie*. Moscow, INFRA-M. Kurs, 2016, 400 p. (In Russ.).
- [7] Sederberg T.W., Zheng J., Bakenov A., Nasri A. T-Splines and T-NURCCs. *ACM Transactions on Graphics*, 2003, vol. 22, pp. 477–484.
- [8] Bazilevs Y., Calo V.M., Cottrell J.A. et al. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering*, 2010, vol. 99, pp. 229–263. <http://doi.org/10.1016/j.cma.2009.02.036>

- [9] Sederberg T.W., Cardon D.L., Finnigan G.T. et al. T-spline Simplification and Local Refinement. *ACM Transactions on Graphics*, 2004, vol. 23, pp. 276–283.
- [10] Cardon D.L. *T-Spline Simplification*. Th. Diss. Brigham Young University, 2007, 114 p.

Received 30.12.2023

Fedotov D.Yu. — Student, Department of Computational Mathematics and Mathematical Physics, Bauman Moscow State Technical University, Moscow, Russian Federation.

Scientific advisor — Zakharov A.A., Ph. D. (Phy. & Math.), Associate Professor, Department of Computational Mathematics and Mathematical Physics, Bauman Moscow State Technical University, Moscow, Russian Federation.

Please cite this article in English as:

Fedotov D.Yu. Software implementation of algorithms for the t-spline construction and local refinement. *Politekhicheskiy molodezhnyy zhurnal*, 2024, no. 01 (90). (In Russ.). URL: <https://ptsj.ru/catalog/math/compmath/964.html>