МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«**ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ**»

Инженерно-технологическая академия

**Учебно-методическое пособие**

**Построение сплайновых кривых и поверхностей на основе дельта-преобразований второго порядка**

по дисциплине

СПЕЦИАЛИЗИРОВАННЫЕ МЕТОДЫ И АЛГОРИТМЫ ОБРАБОТКИ ЦИФРОВОЙ ИНФОРМАЦИИ

Издательство Южного федерального университета

**Таганрог 2016**

УДК 004.92(075)

ББК 32.973я73

П-637

Составители: Кравченко П.П., Жиглатый А.А.

Учебно-методическое пособие к лабораторной работе «Построение сплайновых кривых и поверхностей на основе дельта-преобразований второго порядка» по курсу «Специализированные методы и алгоритмы обработки цифровой информации». - Таганрог: Изд-во ЮФУ, 2016. - 82с.

Предназначено для студентов магистерской программы 09.04.04, направлений 02.03.04, 09.03.04, изучающих курс "Специализированные методы и алгоритмы обработки цифровой информации". Содержит описание лабораторной работы, целью которой является изучение математических основ построения сплайнов на основе дельта-преобразований второго порядка и закрепление навыков программирования.

Ил. 4. Библиогр.: 6 назв.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 5](#_Toc455367064)

[1. Математические основы построения сплайнов на основе дельта-преобразований второго порядка 6](#_Toc455367065)

[2. Алгоритмизация и программная реализация построения одномерных сплайновых кривых на основе дельта-преобразований второго порядка (Лабораторная работа №1) 8](#_Toc455367066)

[2.1. Общие вопросы алгоритмизации построения одномерных сплайновых кривых на основе дельта-преобразований второго порядка 8](#_Toc455367067)

[2.2. Алгоритмизация построения сплайн-функций на основе дельта-преобразований второго порядка с возможностью использования изменяющегося шага дискретизации между узлами интерполяции (в функции от *t*) 11](#_Toc455367068)

[2.3. Алгоритмизация построения сплайн-функций на основе дельта-преобразований второго порядка с постоянным шагом дискретизации  12](#_Toc455367069)

[2.4. Алгоритмизация построения сплайн-функций на основе дельта-преобразований второго порядка с постоянным шагом дискретизации  без использования операций умножения 13](#_Toc455367070)

[2.5. Варианты заданий к лабораторной работе №1 14](#_Toc455367071)

[2.6. Контрольные вопросы 14](#_Toc455367072)

[3. Алгоритмизация и программная реализация построения параметрических дельта-сплайнов (Лабораторная работа №2) 16](#_Toc455367073)

[3.1. Алгоритмизация построения параметрических дельта-сплайнов 16](#_Toc455367074)

[3.2. Варианты заданий к лабораторной работе №2 18](#_Toc455367075)

[3.3. Контрольные вопросы 18](#_Toc455367076)

[4. Алгоритмизация и программная реализация построения сплайновых поверхностей и трехмерных кривых на основе дельта-преобразований второго порядка (Лабораторная работа №3) 20](#_Toc455367077)

[4.1. Алгоритмизация построения сплайновых поверхностей 20](#_Toc455367078)

[4.2. Алгоритмизация построения трехмерных кривых с использованием параметрических дельта-сплайнов 23](#_Toc455367079)

[4.3. Варианты заданий по лабораторной работе №3 23](#_Toc455367080)

[4.4. Контрольные вопросы 24](#_Toc455367081)

[5. Требования к выполнению и оформлению лабораторных работ 26](#_Toc455367082)

[5.1. Требования к выполнению лабораторных работ 26](#_Toc455367083)

[5.2. Требования к отчетам по лабораторным работам 26](#_Toc455367084)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 27](#_Toc455367085)

[ПРИЛОЖЕНИЕ . 28](#_Toc455367086)

[Лабораторная работа №1. 28](#_Toc455367087)

[Лабораторная работа №2. 38](#_Toc455367088)

[Лабораторная работа №3. 51](#_Toc455367090)

**ВВЕДЕНИЕ**

Интенсивное развитие современных технологий всё в большей мере базируется на использовании средств моделирования и высокопроизводительной компьютерной графики. Данные средства представляют возможности формирования и отображения в реальном времени сложных по конфигурациям динамических двух- и трёхмерных имеющих сложную форму объектов: самолётов, ракет, судов и т.п. Возникает также необходимость моделирования в масштабе реального и ускоренного времени различного рода траекторий со специфичными требованиями к гладкости линии траекторного пути (траектории полета летательных аппаратов, снарядов, движения судов и других объектов), интерполирования сложных функций и пр.

Разработка алгоритмов и программных средств построения графических объектов на основе сплайн-функций представляет собой актуальную, но и достаточно сложную задачу [1-4].

Решение этой задачи связано с необходимостью знания известных методов построения сплайн-функций, с умением выполнять сравнительный анализ характеристик этих методов. Практическая сторона включает разработку математической модели и ее алгоритмизацию, разработку непосредственно программных средств, реализующих построение сплайнов и сплайновых поверхностей.

В процессе решения данной задачи возникает необходимость использования для построения сплайн-функций простого с достаточными функциональными возможностями математического аппарата, позволяющего снизить вычислительные затраты, повысить скорость формирования изображений.

Одними из важнейших показателей алгоритмов, реализующих сплайн-функции, являются: вычислительная трудоемкость, характеризующая быстродействие алгоритмов; объем информации, необходимый для хранения параметров кривой; возможности определения значений производных в узлах интерполяции и некоторые другие [5-6].

Целью данных лабораторных работ является ознакомление с новыми алгоритмами построения сплайновых кривых и поверхностей, базирующихся на дельта-преобразованиях второго порядка. Предлагается также выполнить разработку демонстрационных программ, подтверждающих работоспособность и эффективность по быстродействию новой методики.

**1. Математические основы построения сплайнов на основе дельта-преобразований второго порядка**

В основе дельта-преобразований лежит выполняющееся с достаточно малым шагом дискретизации построение непрерывной или дискретной функции, у которой производные определенного (первого, второго) порядка на интервале преобразования одинаковы по модулю и различаются по знаку. Использование дельта-преобразования позволяет заменить операции, выполняющиеся над многоразрядными кодами представления ординат функций, более простыми операциями над квантами, которые могут принимать значения, например, +1 и –1. Благодаря этому при проектировании различных программных средств, цифровых систем и устройств оказывается возможным сократить объемы пересылаемой и хранимой информации, повысить производительность, точность обработки информации, уменьшить затраты на проектирование и производство, повысить пропускную способность каналов связи [1].

В качестве сплайна на основе дельта-преобразований второго порядка (дельта-сплайна) будем понимать локально независимую одномерную функцию, проходящую через узлы интерполяции, которые задают общую форму кривой, обладающую гладкостью первого порядка. Локальная независимость означает, что значения отсчетов фрагмента сплайна между соседними базовыми отсчетами можно определять независимо от других фрагментов сплайна. Эта функция характеризуется следующими краевыми условиями и особенностями (рисунок 1.1):

* фрагменты соседних сплайнов пересекаются (конечная точка одного фрагмента совпадает с начальной точкой следующего фрагмента);
* первые производные двух последовательно идущих фрагментов сплайна равны в точке (узле) их соединения, что гарантирует гладкость перехода от одного фрагмента к другому;
* модуль второй производной интерполирующей функции (кванта преобразования) между узлами интерполяции является постоянной величиной;
* на интервале между узлами имеется два участка знакопостоянства (с противоположными знаками) кванта преобразования.





























Рисунок 1.1 – Интерполяционная кривая, включающая траектории *А* и *В*

В общем случае участок сплайна между соседними отсчетами включает две траектории (рисунок 1.1), которые реализуют участки «разгона» (траектория *B*) и «торможения» (траектория *A*). Исходные, соответствующие этим участкам, дифференциальные уравнения можно записать в виде:



.

Здесь   интерполирующая функция;   значение второй производной (кванта преобразования); – точка переключения знака кванта преобразования;  – краевые условия в узлах интерполяции; *T* – интервал между узлами интерполяции; *D* – расположение точки  переключения знака кванта преобразования.

В качестве исходных данных для расчёта сплайна используются значения отсчётов в узлах интерполяции и интервалы между узлами. Возможно также априорное задание значений производных в узлах, а также значения шага дискретизации Δ*t* формируемой интерполируемой кривой на интервале между узлами.

Алгоритмически решение данной задачи можно представить в виде следующей последовательности действий:

1. Определение (при необходимости)  и , шага ;

2. Нахождение параметров *P* (модуль кванта преобразования) и *D*;

3. Построение собственно сплайна на основе найденных параметров.

Подробное освещение алгоритмизации и возможностей программной реализации построения сплайновых одномерных и многомерных кривых, а также поверхностей на основе дельта-преобразований второго порядка приведены в приложениях 1, 2 и 3 данного учебно-методического пособия.

**2. Алгоритмизация и программная реализация построения одномерных сплайновых кривых на основе дельта-преобразований второго порядка (Лабораторная работа №1)**

**2.1. Общие вопросы алгоритмизации построения одномерных сплайновых кривых на основе дельта-преобразований второго порядка**

Сущность построения сплайна (рисунок 2.1) как интерполирующей функции одной переменной состоит в следующем. Задаются значения отсчетов и значения первых производных в узлах интерполяции, причем производные в точке пересечения двух последовательно идущих фрагментов сплайна должны быть равными. Рассчитываются параметры *P* и *D* для каждого участка и выполняется построение сплайна. Сплайн на рисунке 2.1 представляет соединенные интерполяционными кривыми последовательности отсчётов в узлах (см. также рисунок 1.1).



Рисунок 2.1 – Пример сплайновой кривой

Построение участка сплайна между двумя соседними базовыми отсчетами осуществляется в три этапа:

1. Задано количество базовых отсчетов  (на рисунке 2.1 ). Расстояния между всеми базовыми отсчетами (длины участков сплайна) равны *T* (для рисунка 2.1 принимаем эту длину за единицу). Возможны два варианта задания шага :

* принимаем количество шагов дискретизации для всех участков сплайна, равным *K*; тогда величина шага дискретизации для каждого участка сплайна определяется в виде:

;

* принимаем значение шага дискретизации  в качестве постоянной величины для всех участков сплайнов; тогда количество шагов дискретизации для каждого участка сплайна определится в виде

.

1. Формируется массив базовых отсчетов . Вычисляются производные для базовых отсчетов по одной из трех формул (см. ниже в данном разделе), и формируется соответствующий массив производных Производные в крайних точках задаются априорно (например,0). Выполняется расчет параметров сплайна и формируются массивы для этих параметров.
2. Производится построение сплайновой кривой: строится массив для интерполируемых отсчетов длиной *M\*K*, вычисляются значения отсчетов на каждом интервале сплайна для каждого дискретного отсчета по одной из трех формул интерполяции (см. разделы 2.2, 2.3, 2.4 в соответствии с вариантом задания).

Для упрощения и более обобщенного представления используемых формул будем использовать обозначение пары координат узловых точек в виде , где *n*  номер узла интерполяции, при этом расположение (обозначение) значений независимой переменной между соответствующей парой узлов (на участке интерполяции) должно соответствовать используемой формуле интерполяции.

Производные в узлах могут быть исходно заданы или определены с использованием следующего алгоритма для *M* узлов:

1. Задать значения производных для первого узла и последнего , где ()  количество узлов; например,

и .

2. В случае, если для промежуточного узла выполняются соотношения

или

то принимать

В противном случае значения производных в узлах определять как среднее арифметическое двух производных для узлов слева и справа от узла в соответствии с выражением



***Примечание.*** *При выполнении всех лабораторных работ, выполняемых в рамках данного учебно-методического пособия, не допускать такое размещение пар узлов  и ,  и , при котором  *

Выражения для вычисления значения кванта преобразования ,  и расстояния от узла *n* до точки переключения знака кванта преобразования  вычислять по следующим формулам:

;

,

где

;



В случае  или  принимается .

Перед построением сплайна задается порядок дискретного изменения независимой переменной *t* между узлами интерполяции; в частном случае независимая переменная может изменяться с некоторым постоянным, по крайней мере между парой узлов, шагом . При этом следует учитывать ограничения изменения независимой переменной в пределах соответствующих траекторий *А* и *B* (в пределах  для траектории *А* и  для траектории *B*).

**2.2. Алгоритмизация построения сплайн-функций на основе дельта-преобразований второго порядка с возможностью использования изменяющегося шага дискретизации между узлами интерполяции (**в функции от *t***)**

При выполнении лабораторных работ 1 и 2 перед построением сплайна задаются значения дискретного изменения независимой переменной *t* между узлами интерполяции (например, в виде массивов). Для упрощения количество отсчетов между соседними узлами может быть выбрано одинаковым (достаточно большим для обеспечения графического качества непрерывных линий). Процесс формирования сплайна между каждой парой узлов может реализовываться в соответствии с приводимой ниже последовательностью. Возможно также формирование значений дискретного изменения независимой переменной *t* между узлами интерполяции на основе определенных постоянных значений в пределах участков сплайнов, как это описано в п. 2.1.

При выполнении лабораторной работы 3 значения дискретного изменения независимой переменной *t* формировать с учетом рассмотренных в разделе 4 особенностей построения поверхностей и трехмерных кривых.

Из узла по траектории *А* строится кривая с движением влево на расстояние *Dn* до точки переключения кванта преобразования. Расчет выполняется по формуле:

; 

Далее из узла *n* по траектории *В* строится кривая при движении вправо с шагом на расстояние () до точки  переключения кванта преобразования. Расчет выполняется по формуле:

; 

**2.3. Алгоритмизация построения сплайн-функций на основе дельта-преобразований второго порядка с постоянным шагом дискретизации** 

Перед построением сплайна задается значение дискретного изменения независимой переменной *t* между парами узлов интерполяции; в частном случае независимая переменная может изменяться с некоторым одинаковым постоянным шагом между всеми узлами интерполяции.

Пусть  – постоянный шаг дискретизации независимой переменной,  – количество дискретных шагов по траектории *А*,  – количество дискретных шагов по траектории *В*.

Из узла  по траектории *А* строится кривая с движением влево с шагом  на расстояние *Dn* до точки переключения кванта преобразования  шагов. Расчет выполняется по формулам:



где .

Далее из узла *n* по траектории *В* строится кривая при движении вправо с шагом  на расстояние () до точки переключения кванта преобразования  шагов. Расчет выполняется по формулам:



где .

**2.4. Алгоритмизация построения сплайн-функций на основе дельта-преобразований второго порядка с постоянным шагом дискретизации**  **без использования операций умножения**

Можно показать, что интерполяция может быть реализована без использования операций умножения многоразрядных чисел; при этом на один интерполируемый отсчет (шаг) требуется три операции сложения, что характеризует, вероятно, предельно достижимую по нижнему уровню вычислительную трудоемкость выполнения данных действий для сплайнов.

Предположим, что на подготовительном этапе помимо отмеченных выше значений,  и  определены также , , . Теперь формулы по соответствующим траекториям *A* и *B* можно представить в следующем виде (порядок движения по траекториям *А* и *В*, как и в приведенных выше описаниях):



где .



где .

Применение данного алгоритма позволяет получить наибольший эффект по вычислительной производительности в случае необходимости многократных формирований одних и тех же сплайнов и использовании предварительно рассчитанных и постоянно хранящихся указанных выше параметров (например, при формировании изображений в компьютерных играх).

**2.5. Варианты заданий к лабораторной работе №1**

Разработать и реализовать на компьютере программу построения и графического отображения интерполирующих одномерных сплайновых кривых на основе дельта-преобразований второго порядка в соответствии с заданным вариантом.

|  |  |
| --- | --- |
| **Вариант** | **Задание** |
| 1 | Дельта-сплайны с изменением шага дискретизации в функции от *t* между узлами интерполяции |
| 2 | Дельта-сплайны с постоянным шагом дискретизации между узлами интерполяции и с использованием операций умножения многоразрядных чисел |
| 3 | Дельта-сплайны с постоянным шагом дискретизации между узлами интерполяции и без использования операций умножения многоразрядных чисел |

В приложении приведены примеры детализированного представления алгоритмов построения одномерных сплайновых кривых на основе дельта-преобразований второго порядка в виде псевдокодов по вариантам лабораторной работы №1.

**2.6. Контрольные вопросы**

1. В чем состоит практический интерес использования сплайнов в компьютерной графике?
2. Поясните алгоритмическую последовательность действий в формирования одномерного сплайна?
3. Поясните сущность построения двумерного параметрического сплайна?
4. В чем состоит специфика требований к алгоритмам построения сплайнов для компьютерной графики?
5. В чем состоит сущность метода построения сплайна на основе принципов дельта-преобразований второго порядка?
6. Объяснить смысл процедуры расчета производных при построении сплайнов на основе дельта-преобразований второго порядка.
7. Какой минимальный набор исходных данных необходим для реализации расчета и построения сплайнов на основе дельта-преобразований второго порядка?
8. Поясните сущность параметрических компонент  и алгоритма для построения сплайна.
9. Оцените трудоемкость формирования интерполяционных отсчетов при движении с дискретным шагом  на основе известных (заблаговременно рассчитанных) значений  и .
10. Поясните возможность формирования интерполяционных отсчетов при движении с дискретным шагом без использования операций умножения многоразрядных чисел..
11. В чем состоят достоинства и недостатки рассматриваемых алгоритмов на основе дельта-преобразований второго порядка?
12. Предложите наиболее перспективные с точки зрения эффективности по вычислительной трудоемкости варианты практического использования рассматриваемых алгоритмов построения сплайнов в компьютерной графике при разделении процессов расчета параметров ,  и выполнения собственно интерполяции.

**3. Алгоритмизация и программная реализация построения параметрических дельта-сплайнов (Лабораторная работа №2)**

**3.1. Алгоритмизация построения параметрических дельта-сплайнов**

Выше рассматривались массивы узлов, точки которых были занумерованы так, что абсциссы и их ординаты образовывали строго возрастающие последовательности; это обстоятельство определяло и способ их построения.

В случаях, когда эти условия не могут быть выполнены, построение базируется на параметрическом представлении кривой.

Пусть задан массив узлов  (рисунок 3.1,а, узлы обозначены жирными точками). Формируем для каждой точки  и , массивы в порядке возрастания *n* (независимая переменная представлена ). Соответствующее расположение узлов по  (рисунок 3.1,б) и по(рисунок 3.1,в) принимают вид при возрастающих значениях независимой переменной.

а)

б)

в)

Рисунок 3.1 – Параметрическое представление двумерной кривой

На основе рассмотренной в разделе 2 данного пособия методологии построения одномерных сплайнов строятся две интерполяционные кривые:

-  с использованием узлов 

-  с использованием узлов 

Для вырисовывания кривой на экране компьютера может быть достаточным задание базовых узлов только в экстремальных точках перегиба (жирные точки обозначения этих узлов выделены окружностью). Рекомендуется выбирать количество шагов K на всех шагах интерполяции (между узлами) постоянным. Необходимое количество уточняется в процессе моделирования путем визуальной качественной оценки.

**3.2. Варианты заданий к лабораторной работе №2**

Разработать на основе изложенных в данном разделе и в разделе 2 методологии и алгоритмов программу, реализующую построение и графическое представление двумерной кривой на основе дельта-преобразований в соответствии с вариантом.

|  |  |
| --- | --- |
| **Вариант** | **Задание** |
| 1 | Параметрические дельта-сплайны с изменением шага дискретизации в функции от *t* между узлами интерполяции |
| 2 | Параметрические дельта-сплайны с постоянным шагом дискретизации между узлами интерполяции и с использованием операций умножения многоразрядных чисел |
| 3 | Параметрические дельта-сплайны с постоянным шагом дискретизации между узлами интерполяции и без использования операций умножения многоразрядных чисел |

В приложении приведены примеры детализированного представления алгоритмов построения одномерных сплайновых кривых на основе дельта-преобразований второго порядка в виде псевдокодов по вариантам лабораторной работы №2.

**3.3. Контрольные вопросы**

1. Поясните сущность построения двумерного параметрического сплайна.
2. В чем состоит специфика требований к алгоритмам построения сплайнов для компьютерной графики.
3. Объяснить смысл процедуры расчета производных при построении сплайнов на основе дельта-преобразований второго порядка при построении параметрического сплайна.
4. Какой минимальный набор исходных данных необходим для реализации расчета и построения сплайнов на основе дельта-преобразований второго порядка при построении параметрического сплайна.
5. Поясните сущность параметрических компонент  и  алгоритма для построения параметрического сплайна.
6. Оцените трудоемкость формирования интерполяционных отсчетов при движении с дискретным шагом  на основе известных (заблаговременно рассчитанных) значений  и  при построении параметрического сплайна.
7. Поясните возможность формирования интерполяционных отсчетов при движении с дискретным шагом  без использования операций умножения многоразрядных чисел при формировании параметрического сплайна.
8. Предложите наиболее перспективные с точки зрения эффективности по вычислительной трудоемкости варианты практического использования рассматриваемых алгоритмов построения сплайнов в компьютерной графике при разделении процессов расчета параметров ,  и выполнения собственно интерполяции при формировании параметрического сплайна.

**4. Алгоритмизация и программная реализация построения сплайновых поверхностей и трехмерных кривых на основе дельта-преобразований второго порядка (Лабораторная работа №3)**

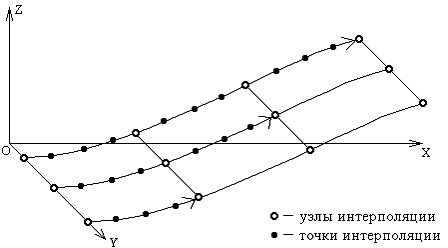
**4.1. Алгоритмизация построения сплайновых поверхностей**

Воспроизведение поверхностей трехмерного объекта играет важную роль при использовании систем обработки графической информации для решения таких прикладных задач, в которых пользователю необходимо взаимодействовать с различными видами твердых тел, а также тогда, когда объекты задаются множеством расположенных в пространстве точек [1-4].

Поверхности могут быть слишком сложны для того, чтобы описать их одним уравнением во всей области их определения, и, следовательно, возникает потребность в кусочной аппроксимации/интерполяции поверхностей. Простейшими являются кусочно-линейные интерполяции поверхностей, в частности, построенные посредством использования многогранника с треугольными гранями (триангуляции). К сожалению, во многих случаях число треугольников, необходимое для получения приемлемого результата, оказывается слишком большим, и поэтому приходится использовать поверхности более высокого порядка (т.е. не кусочно-*линейные*). Вследствие этого возникают проблемы представления таких кусочных поверхностей.

Одним из вариантов решения указанной проблемы являются методы, в которых предусматривается задание на поверхности множества базовых кривых (сплайнов), интерполирующих участки поверхностей.

В рамках лабораторной работы поверхность задается в виде совокупности узлов, располагаемых в точках пересечения данной поверхности с плоскостями, параллельными осям OXZ и OYZ (рисунок 4.1).



а)



б)



в)

Рисунок 4.1 – Построение поверхности на основе дельта-преобразований второго порядка

Для построения поверхности выполняются следующие действия:

1. В плоскостях, параллельных оси OXZ (рисунок 4.1,а) выполняется построение одномерных кривых (разд. 2), которые интерполируют участки между соседними узлами. Ось OX рассматривается как независимая переменная .

2. В плоскостях, параллельных оси OYZ (рисунок 4.1,б) выполняется построение одномерных кривых (разд.2), которые интерполируют участки между соседними узлами. Ось OY рассматривается как независимая переменная .

3. Каждый из полученных четырехугольников разбивается на два треугольника, после чего производится их построение (рисунок 4.1,в.).

Количество шагов на всех интервалах интерполяции принимается постоянным, необходимое количество уточняется в процессе моделирования путем визуальной качественной оценки.

**4.2. Алгоритмизация построения трехмерных кривых с использованием параметрических дельта-сплайнов**

Построение трехмерной кривой сводится обычно к построению кривой по ее двум проекциям. Действительно, зная координаты точек на плоскостях *XOZ* и *YOZ*, можно получить трехмерные координаты каждой точки пространственной кривой.

Лабораторная работа №3 может быть выполнена в следующем порядке. Проекция пространственной кривой на плоскость *XOY* строится по алгоритму построения параметрических дельта-сплайнов, а ось *Z* используется как ось аргумента (перемещение по ней подчиняется правилу , , где *N* — количество интерполирующих точек, находящихся между *i*-м и (*i-*1)-м узлами интерполяции).

**4.3. Варианты заданий по лабораторной работе №3**

Разработать на основе изложенных в данном разделе, в разделах 2 и 3 методологии и алгоритмов программу, реализующую построение и графическое представление поверхности или трехмерной кривой на основе дельта-преобразований второго в соответствии с вариантом.

|  |  |
| --- | --- |
| **Вариант** | **Задание** |
| 1 | Создать программу, позволяющую строить интерполирующую *трехмерную кривую* на основе дельта-сплайнов в функции от *t* между узлами интерполяции |
| 2 | Создать программу, позволяющую строить интерполирующую *трехмерную кривую* на основе дельта-сплайнов с постоянным шагом дискретизации  между узлами интерполяции с использованием операций умножения многоразрядных чисел |
| 3 | Создать программу, позволяющую строить интерполирующую *трехмерную кривую* на основе дельта-сплайнов с постоянным шагом дискретизации  между узлами интерполяции без использования операций умножения многоразрядных чисел |
| 4 | Создать программу, позволяющую строить интерполирующую *поверхность* на основе дельта-сплайнов в функции от *t* между узлами интерполяции |
| 5 | Создать программу, позволяющую строить интерполирующую *поверхность* на основе дельта-сплайнов с постоянным шагом дискретизации  между узлами интерполяции с использованием операций умножения многоразрядных чисел |
| 6 | Создать программу, позволяющую строить интерполирующую *поверхность* на основе дельта-сплайнов с постоянным шагом дискретизации  между узлами интерполяции без использования операций умножения многоразрядных чисел |

В приложении приведены примеры детализированного представления алгоритмов построения одномерных сплайновых кривых на основе дельта-преобразований второго порядка в виде псевдокодов по вариантам лабораторной работы №3.

**4.4. Контрольные вопросы**

1. В чем состоит практический интерес использования сплайнов для построения поверхностей в компьютерной графике.
2. Поясните алгоритмическую последовательность действий в формирования сплайновых поверхностей.
3. Поясните алгоритмическую последовательность действий в формирования сплайновых трехмерных кривых.
4. В чем состоит специфика требований к алгоритмам построения сплайнов для компьютерной графики при построении поверхностей и трехмерных кривых.
5. В чем состоит сущность метода построения сплайна на основе принципов дельта-преобразований второго порядка при построении поверхностей и трехмерных кривых.
6. Объяснить смысл процедуры расчета производных при построении сплайнов на основе дельта-преобразований второго порядка.
7. Какой минимальный набор исходных данных необходим для реализации расчета и построения сплайнов на основе дельта-преобразований второго порядка при построении поверхностей и трехмерных кривых.
8. Поясните сущность компонент  и  алгоритма для построения параметрического сплайна.
9. Оцените трудоемкость формирования интерполяционных отсчетов при движении с дискретным шагом  на основе известных (заблаговременно рассчитанных) значений  и  при формировании поверхностей и трехмерных кривых.
10. Поясните возможность формирования интерполяционных отсчетов при движении с дискретным шагом без использования операций умножения многоразрядных чисел при формировании поверхностей и трехмерных кривых.
11. В чем состоят достоинства и недостатки рассматриваемых алгоритмов на основе дельта-преобразований второго порядка?
12. Предложите наиболее перспективные с точки зрения эффективности по вычислительной трудоемкости варианты практического использования рассматриваемых алгоритмов построения сплайнов в компьютерной графике при разделении процессов расчета параметров ,  и выполнения собственно интерполяции при формировании поверхностей и трехмерных кривых.

**5. Требования к выполнению и оформлению лабораторных работ**

**5.1. Требования к выполнению лабораторных работ**

При выполнении лабораторных работ №1, №2 и №3 следует учесть, что программы, реализующие заданный вариант, должны удовлетворять следующим требованиям:

1. работоспособность;
2. наличие функционального интерфейса, позволяющего обеспечивать загрузку, редактирование и (по возможности) сохранение данных, необходимых для демонстрации работоспособности программы;
3. наличие возможности производить наглядное графическое отображение результатов работы программы;
4. возможность обзора трехмерных объектов (в лабораторной работе №3) из разных точек пространства, для демонстрации *трехмерности* данного объекта (выполнение пункта по согласованию с преподавателем.

Кроме указанного выше, *рекомендуется* проявить смекалку и изобретательность в разработке программ и оформлении интерфейса с целью повышения функциональности данных программ, а также увеличения эффективности взаимодействия пользователя и ПО.

**5.2. Требования к отчетам по лабораторным работам**

Отчет по проделанным лабораторным работам должен содержать:

1. Номер варианта.
2. Задание и цель работы.
3. Теоретический материал, на основе которого должна проводиться практическая работа.
4. Выводы о проделанной работе.
5. Рисунки, отображающие результаты работы программы.
6. Листинг программы.

**СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Методы сплайн-функций. Завьялов Ю.С., Квасов Б.И., Мирошниченко В.Л.- М: Наука, Главная редакция физико-математической науки, 1980. - 355 с.
2. Шикин Е.В., Плис А.И. Кривые и поверхности на экране компьютера. Руководство по сплайнам для пользователей. - М.: ДИАЛОГ-МИФИ, 1996. - 240 с.
3. Скворцов А.В., Мирза Н.С. Алгоритмы построения и анализа триангуляции. – Томск: Изд-во Том. Ун-та, 2006. -168 с.
4. Роджерс Д., Адамс Дж. Математические основы машинной графики: Пер. с англ. - М.: Мир, 2001. - 604 с.
5. Кравченко П.П. Обработка информации и цифровое управление на основе оптимизированных дельта-преобразований второго порядка: Учебное пособие. – Таганрог: Изд-во ЮФУ, 2013. – 98 с.
6. Кравченко П.П. Оптимизированные дельта-преобразований второго порядка. Теория и применение: Монография. - М.: Радиотехника, 2010. — 288 с.

**ПРИЛОЖЕНИЕ**

**Лабораторная работа №1.**

Алгоритмизация и программная реализация построения одномерных сплайновых кривых на основе дельта-преобразований второго порядка.

**Вариант 1.** Дельта-сплайны с изменением шага дискретизации в функции от *t* между узлами интерполяции. Псевдокод.

**использовать** **Файлы**

**цел** *N*

*N***:=2000** **|максимальное число элементов массивов**

**вещтаб** *MASX*[**1**:*N*+**1**] **|выходной массив Х, по которому строим точки в конце**

**вещтаб** *y*[**1**:*N*+**1**] **|выходной массив Y, по которому строим точки в конце**

**цел** *M*,*i*,*j*,*vvod*

**вещтаб** *xNode*[**1**:*N*] **|массив базовых отсчетов х**

**вещтаб** *yNode*[**1**:*N*] **|массив базовых отсчетов y**

**вещтаб** *yder*[**1**:*N*] **|массив производных по y**

**вещ** *yA*, *yB* **|текущие рассматриваемые две точки по Y**

**вещ** *yAder*,*yBder* **|их производные**

**вещ** *T*,*L*,*P*,*D*,*Dx* **|параметры Сплайна**

**вещ** *curx*,*curx2*,*xCheck* **|координаты х между базовыми отсчетами, точка смены знака**

**цел** *K* **|число интервалов**

**цел** флаг **|для проверки входных данных**

**алг**

**нач**

**файл** ф

ф**:=открыть на запись**(**"1.txt"**)

**вывод** **"1-ввести число интервалов, 2-ввести значение шага интерполяции"**

**ввод** *vvod*

**если** *vvod* = **1** **то**

**ввод** *K*

**иначе**

**если** *vvod* = **2** **то**

**ввод** *Dx*

**иначе** **вывод** **"неверный ввод"**

**все**

**все**

**|ЗАДАЕМ МАССИВЫ БАЗОВЫХ ОТСЧЕТОВ**

*M***:=5** **|число точек**

*xNode*[**1**]**:=** **1**; *xNode*[**2**]**:=** **1.5**; *xNode*[**3**]**:=** **3**; *xNode*[**4**]**:=** **7**; *xNode*[**5**]**:=** **8**

*yNode*[**1**]**:=** **2**; *yNode*[**2**]**:=** **7**; *yNode*[**3**]**:=** **4**; *yNode*[**4**]**:=** **3**; *yNode*[**5**]**:=** **7**

*MASX*[**1**]**:=***xNode*[**1**];

*yder*[**1**]**:=0** **|первая производная по У**

*yder*[*M*]**:=0** **|последняя производная по У**

*j***:=1**

**|проверка на то,чтоб точки шли по возрастанию, и с разными Х :**

флаг**:=1**

**нц** **для** *i* **от** **1** **до** (*M*-**1**)

**если** *xNode* [*i*] >= *xNode* [*i*+**1**] **то**

флаг**:=0**

**все**

**кц**

**если** флаг=**1** **то**

**нц** **для** *i* **от** *M*-**1** **до** (**1**)

**если** *xNode* [*i*] >= *xNode* [*i*+**1**] **то**

флаг**:=0**

**все**

**кц**

**все**

**если** флаг=**0** **то** **вывод** **"неправильные входные данные"**

**иначе**

**нц** **для** *i* **от** **2** **до** (*M*-**1**) **|вычисление остальных производных Y**

*yder*[*i*]**:=** **0.5**\*((*yNode*[*i*]-*yNode*[*i*-**1**])/(*xNode*[*i*]-*xNode*[*i*-**1**])+(*yNode*[*i*+**1**]-*yNode*[*i*])/(*xNode*[*i*+**1**]-*xNode*[*i*]));

**кц**

**нц** **для** *i* **от** **1** **до** (*M*-**1**)

*yA***:=***yNode*[*i*]; **|начальный узел**

*yB***:=***yNode*[*i*+**1**]; **|конечный узел**

*yBder***:=***yder*[*i*+**1**]; **|их производные**

*yAder***:=***yder*[*i*];

*T***:=***xNode*[*i*+**1**]-*xNode*[*i*];**|интервал между узлами интерполяции (Tn = tn+1 - tn)**

**если** (*T*=**0**) **то**

*P***:=0**;

*D***:=0**;

**иначе**

*L***:=***yB*-*yA*-**0.5**\**T*\*(*yBder*+*yAder*);**|расстояние от начального узла до точки переключения знака**

*P***:=** (-**1**\**L*-***sign***(*L*)\****sqrt***(*L*\**L*+**0.25**\**T*\**T*\*(*yBder*-*yAder*)\*(*yBder*-*yAder*)))/(**0.5**\**T*\**T*); **|модуль кванта преобразования**

**если** (*P*=**0**) **или** (*L*=**0**) **то** D**:=0**

**иначе** *D***:=**(*yAder*-*yBder*+*T*\**P*)/(**2**\**P*); **| D - расстояние до переключения знака**

**все**

**все**;

*curx***:=***xNode*[*i*];

**если** (*D*<>**0**) **то** *xCheck***:=** *xNode*[*i*]+*D* **|координата переключения знака**

**вывод** **"точка смены знака="**, *xCheck*, **нс**

**иначе** *xCheck***:=** *xNode*[*i*]+ (*xNode*[*i*+**1**]-*xNode*[*i*])/**2**;

**все**

**если** (*vvod* = **1**) **то** *Dx* **:=** (*xNode*[*i*+**1**]-*curx*)/*K***|; если нам задали число интервалов,считаем шаг интерполяции**

**все**

**нц** **пока** *curx*<*xNode*[*i*+**1**]

*curx2***:=***curx*-*xNode*[*i*];

**если** *curx* <= *xCheck* **то** *y*[*j*]**:=** *yA*+*yAder*\**curx2*-*curx2*\**curx2*\**P*/**2**

**иначе** *y*[*j*]**:=***yB*-*yBder*\*(*T*-*curx2*)+(*T*-*curx2*)\*(*T*-*curx2*)\**P*/**2**;

**все**

*MASX*[*j*]**:=***curx*; **|запись Х в массив результатов**

**если** (*curx*+*Dx* >*xNode*[*i*+**1**] ) **то** *curx***:=** *xNode*[*i*+**1**]

**иначе** *curx***:=***curx*+*Dx*;

**все**

*j***:=***j*+**1**;

**кц**

**кц**

*MASX*[*j*]**:=***xNode*[*M*];

*y*[*j*]**:=***yNode*[*M*];

**нц** **для** *i* **от** **1** **до** *j*

**вывод** ф, **нс**, *MASX*[*i*], **" "**, *y*[*i*], **нс**

**кц**

**все**

**закрыть**(ф)

**кон**

**Вариант 2**. Дельта-сплайны с постоянным шагом дискретизации  между узлами интерполяции и с использованием операций умножения многоразрядных чисел. Псевдокод.

**использовать** **Файлы**

**цел** *N* **|максимальное число элементов массивов**

*N***:=2000**

**цел** *Nv* **|входной параметр - количество базовых отсчетов**

*Nv***:=5**

**вещтаб** *x*[**1**:*Nv*] **|базовые отсчеты х**

**вещтаб** *y*[**1**:*Nv*] **|базовые отсчеты у**

**вещтаб** *dy*[**1**:*Nv*] **|производные по у**

**вещтаб** *Rdy*[**1**:*N*] **|параметр Сплайна**

**вещтаб** *Rx*[**1**:*N* ] **|выходной массив х**

**вещтаб** *Ry*[**1**:*N*] **|выходной массив у**

**цел** *i*,*j*,*k*,*error*,*dxtype*,*Ndx*,*Na*,*Nb*,*delta*,*ktemp*

**вещ** *dx*,*T*,*P*,*L*,*D*,*c*

**цел** флаг

**файл** *f*

**алг**

**нач**

**вывод** **"1 - ввести число интервалов, 2 - ввести значение шага интерполяции"**

**ввод** *dxtype*

**если** *dxtype*=**1** **то**

**вывод** **"введите число интервалов"**

**ввод** *Ndx*

**иначе**

**вывод** **"введите значение шага интерполяции"**

**ввод** *dx*

**все**

*x*[**1**]**:=1** *y*[**1**]**:=2**

*x*[**2**]**:=2**  *y*[**2**]**:=7**

*x*[**3**]**:=4**  *y*[**3**]**:=4**

*x*[**4**]**:=7**  *y*[**4**]**:=3**

*x*[**5**]**:=8**  *y*[**5**]**:=7**

**|проверка на то,чтоб точки шли по возрастанию, и с разными Х :**

флаг**:=1**

**нц** **для** *i* **от** **1** **до** (*Nv*-**1**)

**если** *x* [*i*] >= *x* [*i*+**1**] **то**

флаг**:=0**

**все**

**кц**

**если** флаг=**1** **то**

**нц** **для** *i* **от** *Nv*-**1** **до** (**1**)

**если** *x* [*i*] >= *x* [*i*+**1**] **то**

флаг**:=0**

**все**

**кц**

**все**

**если** флаг=**0** **то** **вывод** **"неправильные входные данные"**

**иначе**

*dy*[**1**]**:=0**

*dy*[*Nv*]**:=0**

**нц** **для** *i* **от** **2** **до** *Nv*-**1** **|считаем производные у**

*dy*[*i*]**:=0.5**\*((*y*[*i*]-*y*[*i*-**1**])/(*x*[*i*]-*x*[*i*-**1**])+(*y*[*i*+**1**]-*y*[*i*])/(*x*[*i*+**1**]-*x*[*i*]))

**кц**

*k***:=1**

*error***:=0**

**нц** **для** *i* **от** **1** **до** *Nv*-**1**

*T***:=***x*[*i*+**1**]-*x*[*i*] **|считаем интервал между отсчетами по х**

**если** *T*>**0** **то** **|если он больше нуля то строим дальше**

**если** *dxtype*=**1** **то** *dx***:=***T*/*Ndx* **все**

*L***:=***y*[*i*+**1**]-*y*[*i*]-**0.5**\**T*\*(*dy*[*i*+**1**]+*dy*[*i*])

*P***:=**(-*L*-***sign***(*L*)\****sqrt***(*L*\**L*+**0.25**\**T*\**T*\*(*dy*[*i*+**1**]-*dy*[*i*])\*(*dy*[*i*+**1**]-*dy*[*i*])))/(**0.5**\**T*\**T*)

**если** (*P*=**0**) **или** (*L*=**0**) **то** *D***:=0**

**иначе**

*D***:=**(*dy*[*i*]-*dy*[*i*+**1**]+*T*\**P*)/(**2**\**P*)**|расстояние до точки смены знака**

**все**

*c***:=*abs***(*P*)\**dx*\**dx*

*Na***:=*int***(*D*/*dx* )**|количество точек до точки смены знака**

*Nb***:=*int***((*T*-*D*)/*dx*)**|количество точек после точки смены знака**

**|слева направо**

*delta***:=**-***sign***(*P*)

*Ry*[*k*]**:=***y*[*i*]

*Rdy*[*k*]**:=***dy*[*i*]\**dx*

*Rx*[*k*]**:=***x*[*i*]

**нц** **для** *j* **от** **1** **до** *Na*

*k***:=***k*+**1**

**вывод** *k*, **нс**

*Rdy*[*k*]**:=***Rdy*[*k*-**1**]+*delta*\**c*

*Ry*[*k*]**:=***Ry*[*k*-**1**]+*Rdy*[*k*]-**0.5**\**delta*\**c*

*Rx*[*k*]**:=***Rx*[*k*-**1**]+*dx*

**кц**

**|справа налево**

*k***:=***k*+*Nb*+**1**

**вывод** *k*, **нс**

*ktemp***:=***k*

*delta***:=*sign***(*P*)

*Ry*[*k*]**:=***y*[*i*+**1**]

*Rdy*[*k*]**:=***dy*[*i*+**1**]\**dx*

*Rx*[*k*]**:=***x*[*i*+**1**]

**нц** **для** *j* **от** **1** **до** *Nb*

*k***:=***k*-**1**

**вывод** *k*, **нс**

*Rdy*[*k*]**:=***Rdy*[*k*+**1**]-*delta*\**c*

*Ry*[*k*]**:=***Ry*[*k*+**1**]-*Rdy*[*k*]-**0.5**\**delta*\**c*

*Rx*[*k*]**:=***Rx*[*k*+**1**]-*dx*

**кц**

*k***:=***ktemp*

**иначе** *error***:=1**

**все**

**кц**

**если** *error*=**0** **то** **|если расстояние между соседними Х больше нуля**

*f***:=открыть на запись**(**"1.txt"**)

**нц** **для** *i* **от** **1** **до** *k*

**вывод** *f*, **нс**, *Rx*[*i*],**" "**,*Ry*[*i*], **нс**

**кц**

**закрыть**(*f*)

**все**

**все**

**кон**

**Вариант 3.** Дельта-сплайны с постоянным шагом дискретизации  между узлами интерполяции без использования операций умножения многоразрядных чисел. Псевдокод.

**использовать** **Файлы**

**цел** *N* **|максимальное число элементов массивов**

*N***:=2000**

**цел** *Nv* **|входной параметр - количество базовых отсчетов**

*Nv***:=5**

**вещтаб** *x*[**1**:*Nv*] **|базовые отсчеты х**

**вещтаб** *y*[**1**:*Nv*] **|базовые отсчеты у**

**вещтаб** *dy*[**1**:*Nv*] **|производные по у**

**вещтаб** *S*[**1**:*N*] **|параметр Сплайна**

**вещтаб** *Rx*[**1**:*N*] **|выходной массив х**

**вещтаб** *Ry*[**1**:*N*] **|выходной массив у**

**цел** *i*,*j*,*k*,*error*,*dxtype*,*Ndx*,*Na*,*Nb*,*ktemp*

**вещ** *dx*,*T*,*P*,*L*,*D*,*c*

**цел** флаг

**файл** *f*

**алг**

**нач**

**вывод** **"1 - ввести число интервалов, 2 - ввести значение шага интерполяции"**

**ввод** *dxtype*

**если** *dxtype*=**1** **то**

**вывод** **"введите число интервалов"**

**ввод** *Ndx*

**иначе**

**вывод** **"введите значение шага интерполяции"**

**ввод** *dx*

**все**

*x*[**1**]**:=1** *y*[**1**]**:=2**

*x*[**2**]**:=2**  *y*[**2**]**:=7**

*x*[**3**]**:=4** *y*[**3**]**:=4**

*x*[**4**]**:=7**  *y*[**4**]**:=3**

*x*[**5**]**:=8**  *y*[**5**]**:=7**

**|проверка на то,чтоб точки шли по возрастанию, и с разными Х :**

флаг**:=1**

**нц** **для** *i* **от** **1** **до** (*Nv*-**1**)

**если** *x* [*i*] >= *x* [*i*+**1**] **то**

флаг**:=0**

**все**

**кц**

**если** флаг=**1** **то**

**нц** **для** *i* **от** *Nv*-**1** **до** (**1**)

**если** *x* [*i*] >= *x* [*i*+**1**] **то**

флаг**:=0**

**все**

**кц**

**все**

**если** флаг=**0** **то** **вывод** **"неправильные входные данные"**

**иначе**

*dy*[**1**]**:=0**

*dy*[*Nv*]**:=0**

**нц** **для** *i* **от** **2** **до** *Nv*-**1** **|считаем производные у**

*dy*[*i*]**:=0.5**\*((*y*[*i*]-*y*[*i*-**1**])/(*x*[*i*]-*x*[*i*-**1**])+(*y*[*i*+**1**]-*y*[*i*])/(*x*[*i*+**1**]-*x*[*i*]))

**кц**

*k***:=1**

*error***:=0**

**нц** **для** *i* **от** **1** **до** *Nv*-**1**

*T***:=***x*[*i*+**1**]-*x*[*i*] **|считаем интервал между отсчетами по х**

**если** *T*>**0** **то** **|если он больше нуля то строим дальше**

**если** *dxtype*=**1** **то** *dx***:=***T*/*Ndx* **все**

*L***:=***y*[*i*+**1**]-*y*[*i*]-**0.5**\**T*\*(*dy*[*i*+**1**]+*dy*[*i*])

*P***:=**(-*L*-***sign***(*L*)\****sqrt***(*L*\**L*+**0.25**\**T*\**T*\*(*dy*[*i*+**1**]-*dy*[*i*])\*(*dy*[*i*+**1**]-*dy*[*i*])))/(**0.5**\**T*\**T*)

**если** (*P*=**0**) **или** (*L*=**0**) **то** *D***:=0**

**иначе**

*D***:=**(*dy*[*i*]-*dy*[*i*+**1**]+*T*\**P*)/(**2**\**P*) **|расстояние до точки смены знака**

**все**

*c***:=***P*\**dx*\**dx*

*Na***:=*int***(*D*/*dx*) **|количество точек до точки смены знака**

*Nb***:=*int***((*T*-*D*)/*dx*) **|количество точек после точки смены знака**

**|слева направо**

**вывод** *k*, **нс**

*Ry*[*k*]**:=***y*[*i*]

*S*[*k*]**:=***dy*[*i*]\**dx*

*Rx*[*k*]**:=***x*[*i*]

**нц** **для** *j* **от** **1** **до** *Na*

*k***:=***k*+**1**

**вывод** *k*, **нс**

*S*[*k*]**:=***S*[*k*-**1**]-*c*

*Ry*[*k*]**:=***Ry*[*k*-**1**]+*S*[*k*]+**0.5**\**c*

*Rx*[*k*]**:=***Rx*[*k*-**1**]+*dx*

**кц**

**|справа налево**

*k***:=***k*+*Nb*+**1**

**вывод** *k*, **нс**

*ktemp***:=***k*

*Ry*[*k*]**:=***y*[*i*+**1**]

*S*[*k*]**:=***dy*[*i*+**1**]\**dx*

*Rx*[*k*]**:=***x*[*i*+**1**]

**нц** **для** *j* **от** **1** **до** *Nb*

*k***:=***k*-**1**

**вывод** *k*, **нс**

*S*[*k*]**:=***S*[*k*+**1**]-*c*

*Ry*[*k*]**:=***Ry*[*k*+**1**]-*S*[*k*]-**0.5**\**c*

*Rx*[*k*]**:=***Rx*[*k*+**1**]-*dx*

**кц**

*k***:=***ktemp*

**иначе** *error***:=1**

**все**

**кц**

**если** *error*=**0** **то** **|если расстояние между соседними Х больше нуля**

*f***:=открыть на запись**(**"1.txt"**)

**нц** **для** *i* **от** **1** **до** *k*

**вывод** *f*, **нс**, *Rx*[*i*],**" "**,*Ry*[*i*], **нс**

**кц**

**закрыть**(*f*)

**все**

**все**

**кон**

**Лабораторная работа №2.**

Алгоритмизация и программная реализация построения параметрических дельта-сплайнов.

**Вариант №1.** Параметрические дельта-сплайны с изменением шага дискретизации в функции от *t* между узлами интерполяции. Псевдокод.

**использовать** **Файлы**

**цел** *N*

*N***:=2000** **|максимальное число элементов массивов**

**вещтаб** *MASX*[**1**:*N*+**1**] **|выходной массив Х, по которому строим точки в конце**

**вещтаб** *y*[**1**:*N*+**1**] **|выходной массив Y, по которому строим точки в конце**

**цел** *M*,*i*,*j*,*vvod*

**вещтаб** *xNode*[**1**:*N*] **|массив базовых отсчетов х**

**вещтаб** *yNode*[**1**:*N*] **|массив базовых отсчетов y**

**вещтаб** *tNode*[**1**:*N*] **|массив параметра t**

**вещтаб** *yder*[**1**:*N*] **|массив производных по y**

**вещтаб** *xder*[**1**:*N*] **|массив производных по x**

**вещ** *xA*, *xB* **|текущие рассматриваемые две точки по X**

**вещ** *xAder*,*xBder* **|их производные**

**вещ** *yA*, *yB* **|текущие рассматриваемые две точки по Y**

**вещ** *yAder*,*yBder* **|их производные**

**вещ** *T*,*L*,*P*,*D*,*step*,*Dt* **|параметры Сплайна**

**вещ** *curt*,*curt2*,*xCheck*,*yCheck* **|координаты х между базовыми отсчетами, точка смены знака**

**цел** *K* **|число интервалов**

**вещтаб** *qx*[**1**:*N*+**1**],*qy*[**1**:*N*+**1**],*qty*[**1**:*N*+**1**],*qtx*[**1**:*N*+**1**]

**алг**

**нач**

**файл** ф

ф**:=открыть на запись**(**"1.txt"**)

**вывод** ( **'1- ввести число интервалов, 2- ввести значение шага интерполяции'**);

**ввод** (*vvod*);

**если** (*vvod* = **1**) **то**

**ввод** (*K*);

**иначе** **если** (*vvod* = **2**) **то**

**ввод** (*step*);

**иначе** **вывод** ( **'неверный ввод'**);

**все**

**все**

**|ЗАДАЕМ МАССИВЫ БАЗОВЫХ ОТСЧЕТОВ**

*M***:=10**;**|число точек**

*xNode*[**1**]**:=** **1**; *xNode*[**2**]**:=** **2**; *xNode*[**3**]**:=** **3**; *xNode*[**4**]**:=** **4**; *xNode*[**5**]**:=** **5**; *xNode*[**6**]**:=** **5**; *xNode*[**7**]**:=** **4**; *xNode*[**8**]**:=** **3**; *xNode*[**9**]**:=** **2**; *xNode*[**10**]**:=** **1**;

*yNode*[**1**]**:=** **1**; *yNode*[**2**]**:=** **1**; *yNode*[**3**]**:=** **1**; *yNode*[**4**]**:=** **2**; *yNode*[**5**]**:=** **3**; *yNode*[**6**]**:=** **5**; *yNode*[**7**]**:=** **6**; *yNode*[**8**]**:=** **6**; *yNode*[**9**]**:=** **6**; *yNode*[**10**]**:=** **4**;

**|формируем массив t:**

*tNode*[**1**]**:=0**;

**нц** **для** *i* **от** **2** **до** *M* **|вычисление остальных t**

*tNode*[*i*]**:=** *i*;

**кц**

*qx*[**1**]**:=***xNode*[**1**]; **|записываем Х в выходной массив**

**|вычисляем производные:**

*yder*[**1**]**:=0**; **|первая производная по У**

*xder*[**1**]**:=0**; **|первая производная по X**

*yder*[*M*]**:=0**; **|последняя производная по У**

*xder*[*M*]**:=0**; **|последняя производная по X**

*j***:=1**;

**нц** **для** *i* **от** **2** **до** (*M*-**1**) **|вычисление остальных производных Y**

*yder*[*i*]**:=** **0.5**\*((*yNode*[*i*]-*yNode*[*i*-**1**])/(*tNode*[*i*]-*tNode*[*i*-**1**])+(*yNode*[*i*+**1**]-*yNode*[*i*])/(*tNode*[*i*+**1**]-*tNode*[*i*]));

*xder*[*i*]**:=** **0.5**\*((*xNode*[*i*]-*xNode*[*i*-**1**])/(*tNode*[*i*]-*tNode*[*i*-**1**])+(*xNode*[*i*+**1**]-*xNode*[*i*])/(*tNode*[*i*+**1**]-*tNode*[*i*]));

**кц**

**|считаем сплайн для X:**

**нц** **для** *i* **от** **1** **до** (*M*-**1**)

*xA***:=***xNode*[*i*]; **|начальный узел**

*xB***:=***xNode*[*i*+**1**]; **|конечный узел**

*xBder***:=***xder*[*i*+**1**];**|их производные**

*xAder***:=***xder*[*i*];

*T***:=***tNode*[*i*+**1**]-*tNode*[*i*];**|интервал между узлами интерполяции (Tn = tn+1 - tn)**

**если** (*T*=**0**) **то** *P***:=0**; *D***:=0**;

**иначе**

*L***:=***xB*-*xA*-**0.5**\**T*\*(*xBder*+*xAder*);**|расстояние от начального узла до точки переключения знака**

*P***:=** (-**1**\**L*-***sign***(*L*)\****sqrt***(*L*\**L*+**0.25**\**T*\**T*\*(*xBder*-*xAder*)\*(*xBder*-*xAder*)))/(**0.5**\**T*\**T*); **|модуль кванта преобразования**

**если** (*P*=**0**) **или** (*L*=**0**) **то** D**:=0**

**иначе** *D***:=**(*xAder*-*xBder*+*T*\**P*)/(**2**\**P*); **| D - расстояние до переключения знака**

**все**

**все**

*curt***:=***tNode*[*i*];

**если** (*D*<>**0**) **то** *xCheck***:=** *tNode*[*i*]+*D* **|координата переключения знака**

**иначе** *xCheck***:=** *tNode*[*i*]+ (*tNode*[*i*+**1**]-*tNode*[*i*])/**2**;

**все**

**если** (*vvod* = **1**) **то** *step* **:=** (*tNode*[*i*+**1**]-*curt*)/*K*; **|если нам задали число интервалов,считаем шаг интерполяции**

**все**

**нц** **пока** *curt*<*tNode*[*i*+**1**]

*curt2***:=***curt*-*tNode*[*i*];

**если** *curt* <= *xCheck* **то** *qx*[*j*]**:=** *xA*+*xAder*\**curt2*-*curt2*\**curt2*\**P*/**2**

**иначе** *qx*[*j*]**:=***xB*-*xBder*\*(*T*-*curt2*)+(*T*-*curt2*)\*(*T*-*curt2*)\**P*/**2**;

**все**

*qtx*[*j*]**:=***curt*; **|запись Х в массив результатов**

**если** (*curt*+*step* >*tNode*[*i*+**1**] ) **то** *curt***:=** *tNode*[*i*+**1**]

**иначе** *curt***:=***curt*+*step*;

**все**

*j***:=***j*+**1**;

**кц**

**кц**

**|считаем сплайн для У:**

*j***:=1**;

**нц** **для** *i* **от** **1** **до** (*M*-**1**)

*yA***:=***yNode*[*i*]; **|начальный узел**

*yB***:=***yNode*[*i*+**1**]; **|конечный узел**

*yBder***:=***yder*[*i*+**1**];**|их производные**

*yAder***:=***yder*[*i*];

*T***:=***tNode*[*i*+**1**]-*tNode*[*i*];**|интервал между узлами интерполяции (Tn = tn+1 - tn)**

**если** (*T*=**0**) **то** *P***:=0**; *D***:=0**;

**иначе**

*L***:=***yB*-*yA*-**0.5**\**T*\*(*yBder*+*yAder*);**|расстояние от начального узла до точки переключения знака**

*P***:=** (-**1**\**L*-***sign***(*L*)\****sqrt***(*L*\**L*+**0.25**\**T*\**T*\*(*yBder*-*yAder*)\*(*yBder*-*yAder*)))/(**0.5**\**T*\**T*); **|модуль кванта преобразования**

**все**

**если** (*P*=**0**) **или** (*L*=**0**) **то** D**:=0**

**иначе** *D***:=**(*yAder*-*yBder*+*T*\**P*)/(**2**\**P*); **| D - расстояние до переключения знака**

**все**

*curt***:=***tNode*[*i*];

**если** (*D*<>**0**) **то** *yCheck***:=** *tNode*[*i*]+*D* **|координата переключения знака**

**иначе** *yCheck***:=** *tNode*[*i*]+ (*tNode*[*i*+**1**]-*tNode*[*i*])/**2**;

**все**

**если** (*vvod* = **1**) **то** *step* **:=** (*tNode*[*i*+**1**]-*curt*)/*K*; **все** **|если нам задали число интервалов,считаем шаг интерполяции**

**нц** **пока** *curt*<*tNode*[*i*+**1**]

*curt2***:=***curt*-*tNode*[*i*];

**если** *curt* <= *yCheck* **то** *qy*[*j*]**:=** *yA*+*yAder*\**curt2*-*curt2*\**curt2*\**P*/**2**

**иначе** *qy*[*j*]**:=***yB*-*yBder*\*(*T*-*curt2*)+(*T*-*curt2*)\*(*T*-*curt2*)\**P*/**2**;

**все**

*qty*[*j*]**:=***curt*; **|запись Х в массив результатов**

**если** (*curt*+*step* >*tNode*[*i*+**1**] ) **то** *curt***:=** *tNode*[*i*+**1**]

**иначе** *curt***:=***curt*+*step*;

**все**

*j***:=***j*+**1**;

**кц**

**кц**

**|ВЫВОД КОНЕЧНОГО РЕЗУЛЬТАТА В ФАЙЛ**

**нц** **для** *i* **от** **1** **до** *j*-**1**

**вывод** ф,**нс**, *qx*[*i*], **" "**, *qy*[*i*], **нс**

**кц**

**закрыть**(ф)

**кон**

**Вариант №2**. Параметрические дельта-сплайны с постоянным шагом дискретизации  между узлами интерполяции с использованием операций умножения многоразрядных чисел. Псевдокод.

**использовать** **Файлы**

**цел** *N*

*N***:=2000**

**цел** *Nv*

*Nv***:=10**

**вещтаб** *x*[**1**:*Nv*]

**вещтаб** *y*[**1**:*Nv*]

**вещтаб** *t*[**1**:*Nv*]

**вещтаб** *dx*[**1**:*N*]

**вещтаб** *Rdx*[**1**:*N*]

**вещтаб** *dy*[**1**:*N*]

**вещтаб** *Rdy*[**1**:*N*]

**вещтаб** *Rx*[**1**:*N*]

**вещтаб** *Ry*[**1**:*N*]

**цел** *i*,*j*,*k*,*error*,*dttype*,*Ndt*,*Na*,*Nb*,*delta*,*ktemp*

**вещ** *dt*,*T*,*P*,*L*,*D*,*c*

**файл** *f*

**алг**

**нач**

**вывод** **"1 - ввести число интервалов, 2 - ввести значение шага интерполяции"**

**ввод** *dttype*

**если** *dttype*=**1** **то**

**вывод** **"введите число интервалов"**

**ввод** *Ndt*

**иначе**

**вывод** **"введите значение шага интерполяции"**

**ввод** *dt*

**все**

*x*[**1**] **:=2**; *y*[**1**] **:=3**;

*x*[**2**] **:=4**; *y*[**2**] **:=4**;

*x*[**3**] **:=9**; *y*[**3**] **:=2**;

*x*[**4**] **:=11**; *y*[**4**] **:=5**;

*x*[**5**] **:=8**; *y*[**5**] **:=7**;

*x*[**6**] **:=7**; *y*[**6**] **:=5**;

*x*[**7**] **:=9**; *y*[**7**] **:=4**;

*x*[**8**] **:=10**; *y*[**8**] **:=5**;

*x*[**9**] **:=8**; *y*[**9**] **:=6**;

*x*[**10**]**:=8.5**; *y*[**10**]**:=5**;

**нц** **для** *i* **от** **1** **до** *Nv*

*t*[*i*]**:=***i*

**кц**

*dy*[**1**]**:=0**

*dy*[*Nv*]**:=0**

**нц** **для** *i* **от** **2** **до** *Nv*-**1**

*dy*[*i*]**:=0.5**\*((*y*[*i*]-*y*[*i*-**1**])/(*t*[*i*]-*t*[*i*-**1**])+(*y*[*i*+**1**]-*y*[*i*])/(*t*[*i*+**1**]-*t*[*i*]))

**кц**

*k***:=1**

*error***:=0**

**нц** **для** *i* **от** **1** **до** *Nv*-**1**

*T***:=***t*[*i*+**1**]-*t*[*i*]

**если** *T*>**0** **то**

**если** *dttype*=**1** **то** *dt***:=***T*/*Ndt* **все**

*L***:=***y*[*i*+**1**]-*y*[*i*]-**0.5**\**T*\*(*dy*[*i*+**1**]+*dy*[*i*])

*P***:=**(-*L*-***sign***(*L*)\****sqrt***(*L*\**L*+**0.25**\**T*\**T*\*(*dy*[*i*+**1**]-*dy*[*i*])\*(*dy*[*i*+**1**]-*dy*[*i*])))/(**0.5**\**T*\**T*)

**если** (*P*=**0**) **или** (*L*=**0**) **то** *D***:=0**

**иначе**

*D***:=**(*dy*[*i*]-*dy*[*i*+**1**]+*T*\**P*)/(**2**\**P*)

**все**

*c***:=*abs***(*P*)\**dt*\**dt*

*Na***:=*int***(*D*/*dt*)

*Nb***:=*int***((*T*-*D*)/*dt*)

**|слева направо**

*delta***:=**-***sign***(*P*)

**вывод** *k*, **нс**

*Ry*[*k*]**:=***y*[*i*]

*Rdy*[*k*]**:=***dy*[*i*]\**dt*

**нц** **для** *j* **от** **1** **до** *Na*

*k***:=***k*+**1**

**вывод** *k*, **нс**

*Rdy*[*k*]**:=***Rdy*[*k*-**1**]+*delta*\**c*

*Ry*[*k*]**:=***Ry*[*k*-**1**]+*Rdy*[*k*]-**0.5**\**delta*\**c*

**кц**

**|справа налево**

*k***:=***k*+*Nb*+**1**

**вывод** *k*, **нс**

*ktemp***:=***k*

*delta***:=*sign***(*P*)

*Ry*[*k*]**:=***y*[*i*+**1**]

*Rdy*[*k*]**:=***dy*[*i*+**1**]\**dt*

**нц** **для** *j* **от** **1** **до** *Nb*

*k***:=***k*-**1**

**вывод** *k*, **нс**

*Rdy*[*k*]**:=***Rdy*[*k*+**1**]-*delta*\**c*

*Ry*[*k*]**:=***Ry*[*k*+**1**]-*Rdy*[*k*]-**0.5**\**delta*\**c*

**кц**

*k***:=***ktemp*

**иначе** *error***:=1**

**все**

**кц**

**|иксы**

*dx*[**1**]**:=0**

*dx*[*Nv*]**:=0**

**нц** **для** *i* **от** **2** **до** *Nv*-**1**

*dx*[*i*]**:=0.5**\*((*x*[*i*]-*x*[*i*-**1**])/(*t*[*i*]-*t*[*i*-**1**])+(*x*[*i*+**1**]-*x*[*i*])/(*t*[*i*+**1**]-*t*[*i*]))

**кц**

*k***:=1**

*error***:=0**

**нц** **для** *i* **от** **1** **до** *Nv*-**1**

*T***:=***t*[*i*+**1**]-*t*[*i*]

**если** *T*>**0** **то**

**если** *dttype*=**1** **то** *dt***:=***T*/*Ndt* **все**

*L***:=***x*[*i*+**1**]-*x*[*i*]-**0.5**\**T*\*(*dx*[*i*+**1**]+*dx*[*i*])

*P***:=**(-*L*-***sign***(*L*)\****sqrt***(*L*\**L*+**0.25**\**T*\**T*\*(*dx*[*i*+**1**]-*dx*[*i*])\*(*dx*[*i*+**1**]-*dx*[*i*])))/(**0.5**\**T*\**T*)

**если** (*P*=**0**) **или** (*L*=**0**) **то** *D***:=0**

**иначе**

*D***:=**(*dx*[*i*]-*dx*[*i*+**1**]+*T*\**P*)/(**2**\**P*)

**все**

*c***:=*abs***(*P*)\**dt*\**dt*

*Na***:=*int***(*D*/*dt*)

*Nb***:=*int***((*T*-*D*)/*dt*)

**|слева направо**

*delta***:=**-***sign***(*P*)

**вывод** *k*, **нс**

*Rx*[*k*]**:=***x*[*i*]

*Rdx*[*k*]**:=***dx*[*i*]\**dt*

**нц** **для** *j* **от** **1** **до** *Na*

*k***:=***k*+**1**

**вывод** *k*, **нс**

*Rdx*[*k*]**:=***Rdx*[*k*-**1**]+*delta*\**c*

*Rx*[*k*]**:=***Rx*[*k*-**1**]+*Rdx*[*k*]-**0.5**\**delta*\**c*

**кц**

**|справа налево**

*k***:=***k*+*Nb*+**1**

**вывод** *k*, **нс**

*ktemp***:=***k*

*delta***:=*sign***(*P*)

*Rx*[*k*]**:=***x*[*i*+**1**]

*Rdx*[*k*]**:=***dx*[*i*+**1**]\**dt*

**нц** **для** *j* **от** **1** **до** *Nb*

*k***:=***k*-**1**

**вывод** *k*, **нс**

*Rdx*[*k*]**:=***Rdx*[*k*+**1**]-*delta*\**c*

*Rx*[*k*]**:=***Rx*[*k*+**1**]-*Rdx*[*k*]-**0.5**\**delta*\**c*

**кц**

*k***:=***ktemp*

**иначе** *error***:=1**

**все**

**кц**

**если** *error*=**0** **то**

*f***:=открыть на запись**(**"2.txt"**)

**нц** **для** *i* **от** **1** **до** *k*

**вывод** *f*, **нс**, *Rx*[*i*],**" "**,*Ry*[*i*], **нс**

**кц**

**закрыть**(*f*)

**все**

**кон**

**Вариант №3**. Параметрические дельта-сплайны с постоянным шагом дискретизации  между узлами интерполяции без использования операций многоразрядного умножения. Псевдокод.

**использовать** **Файлы**

**цел** *N*

*N***:=2000**

**цел** *Nv*

*Nv***:=10**

**вещтаб** *x*[**1**:*Nv*]

**вещтаб** *y*[**1**:*Nv*]

**вещтаб** *t*[**1**:*Nv*]

**вещтаб** *dx*[**1**:*N*]

**вещтаб** *Rdx*[**1**:*N*]

**вещтаб** *dy*[**1**:*N*]

**вещтаб** *Rdy*[**1**:*N*]

**вещтаб** *Rx*[**1**:*N*]

**вещтаб** *Ry*[**1**:*N*]

**цел** *i*,*j*,*k*,*error*,*dttype*,*Ndt*,*Na*,*Nb*,*ktemp*

**вещ** *dt*,*T*,*P*,*L*,*D*,*c*

**файл** *f*

**алг**

**нач**

**вывод** **"1 - ввести число интервалов, 2 - ввести значение шага интерполяции"**

**ввод** *dttype*

**если** *dttype*=**1** **то**

**вывод** **"введите число интервалов"**

**ввод** *Ndt*

**иначе**

**вывод** **"введите значение шага интерполяции"**

**ввод** *dt*

**все**

*x*[**1**] **:=2**; *y*[**1**] **:=3**;

*x*[**2**] **:=4**; *y*[**2**] **:=4**;

*x*[**3**] **:=9**; *y*[**3**] **:=2**;

*x*[**4**] **:=11**; *y*[**4**] **:=5**;

*x*[**5**] **:=8**; *y*[**5**] **:=7**;

*x*[**6**] **:=7**; *y*[**6**] **:=5**;

*x*[**7**] **:=9**; *y*[**7**] **:=4**;

*x*[**8**] **:=10**; *y*[**8**] **:=5**;

*x*[**9**] **:=8**; *y*[**9**] **:=6**;

*x*[**10**]**:=8.5**; *y*[**10**]**:=5**;

**нц** **для** *i* **от** **1** **до** *Nv*

*t*[*i*]**:=***i*

**кц**

*dy*[**1**]**:=0**

*dy*[*Nv*]**:=0**

**нц** **для** *i* **от** **2** **до** *Nv*-**1**

*dy*[*i*]**:=0.5**\*((*y*[*i*]-*y*[*i*-**1**])/(*t*[*i*]-*t*[*i*-**1**])+(*y*[*i*+**1**]-*y*[*i*])/(*t*[*i*+**1**]-*t*[*i*]))

**кц**

*k***:=1**

*error***:=0**

**нц** **для** *i* **от** **1** **до** *Nv*-**1**

*T***:=***t*[*i*+**1**]-*t*[*i*]

**если** *T*>**0** **то**

**если** *dttype*=**1** **то** *dt***:=***T*/*Ndt* **все**

*L***:=***y*[*i*+**1**]-*y*[*i*]-**0.5**\**T*\*(*dy*[*i*+**1**]+*dy*[*i*])

*P***:=**(-*L*-***sign***(*L*)\****sqrt***(*L*\**L*+**0.25**\**T*\**T*\*(*dy*[*i*+**1**]-*dy*[*i*])\*(*dy*[*i*+**1**]-*dy*[*i*])))/(**0.5**\**T*\**T*)

**если** (*P*=**0**) **или** (*L*=**0**) **то** *D***:=0**

**иначе**

*D***:=**(*dy*[*i*]-*dy*[*i*+**1**]+*T*\**P*)/(**2**\**P*)

**все**

*c***:=***P*\**dt*\**dt*

*Na***:=*int***(*D*/*dt*)

*Nb***:=*int***((*T*-*D*)/*dt*)

**|слева направо**

**вывод** *k*, **нс**

*Ry*[*k*]**:=***y*[*i*]

*Rdy*[*k*]**:=***dy*[*i*]\**dt*

**нц** **для** *j* **от** **1** **до** *Na*

*k***:=***k*+**1**

**вывод** *k*, **нс**

*Rdy*[*k*]**:=***Rdy*[*k*-**1**]-*c*

*Ry*[*k*]**:=***Ry*[*k*-**1**]+*Rdy*[*k*]+**0.5**\**c*

**кц**

**|справа налево**

*k***:=***k*+*Nb*+**1**

**вывод** *k*, **нс**

*ktemp***:=***k*

*Ry*[*k*]**:=***y*[*i*+**1**]

*Rdy*[*k*]**:=***dy*[*i*+**1**]\**dt*

**нц** **для** *j* **от** **1** **до** *Nb*

*k***:=***k*-**1**

**вывод** *k*, **нс**

*Rdy*[*k*]**:=***Rdy*[*k*+**1**]-*c*

*Ry*[*k*]**:=***Ry*[*k*+**1**]-*Rdy*[*k*]-**0.5**\**c*

**кц**

*k***:=***ktemp*

**иначе** *error***:=1**

**все**

**кц**

**|иксы**

*dx*[**1**]**:=0**

*dx*[*Nv*]**:=0**

**нц** **для** *i* **от** **2** **до** *Nv*-**1**

*dx*[*i*]**:=0.5**\*((*x*[*i*]-*x*[*i*-**1**])/(*t*[*i*]-*t*[*i*-**1**])+(*x*[*i*+**1**]-*x*[*i*])/(*t*[*i*+**1**]-*t*[*i*]))

**кц**

*k***:=1**

*error***:=0**

**нц** **для** *i* **от** **1** **до** *Nv*-**1**

*T***:=***t*[*i*+**1**]-*t*[*i*]

**если** *T*>**0** **то**

**если** *dttype*=**1** **то** *dt***:=***T*/*Ndt* **все**

*L***:=***x*[*i*+**1**]-*x*[*i*]-**0.5**\**T*\*(*dx*[*i*+**1**]+*dx*[*i*])

*P***:=**(-*L*-***sign***(*L*)\****sqrt***(*L*\**L*+**0.25**\**T*\**T*\*(*dx*[*i*+**1**]-*dx*[*i*])\*(*dx*[*i*+**1**]-*dx*[*i*])))/(**0.5**\**T*\**T*)

**если** (*P*=**0**) **или** (*L*=**0**) **то** *D***:=0**

**иначе**

*D***:=**(*dx*[*i*]-*dx*[*i*+**1**]+*T*\**P*)/(**2**\**P*)

**все**

*c***:=***P*\**dt*\**dt*

*Na***:=*int***(*D*/*dt*)

*Nb***:=*int***((*T*-*D*)/*dt*)

**|слева направо**

**вывод** *k*, **нс**

*Rx*[*k*]**:=***x*[*i*]

*Rdx*[*k*]**:=***dx*[*i*]\**dt*

**нц** **для** *j* **от** **1** **до** *Na*

*k***:=***k*+**1**

**вывод** *k*, **нс**

*Rdx*[*k*]**:=***Rdx*[*k*-**1**]-*c*

*Rx*[*k*]**:=***Rx*[*k*-**1**]+*Rdx*[*k*]+**0.5**\**c*

**кц**

**|справа налево**

*k***:=***k*+*Nb*+**1**

**вывод** *k*, **нс**

*ktemp***:=***k*

*Rx*[*k*]**:=***x*[*i*+**1**]

*Rdx*[*k*]**:=***dx*[*i*+**1**]\**dt*

**нц** **для** *j* **от** **1** **до** *Nb*

*k***:=***k*-**1**

**вывод** *k*, **нс**

*Rdx*[*k*]**:=***Rdx*[*k*+**1**]-*c*

*Rx*[*k*]**:=***Rx*[*k*+**1**]-*Rdx*[*k*]-**0.5**\**c*

**кц**

*k***:=***ktemp*

**иначе** *error***:=1**

**все**

**кц**

**если** *error*=**0** **то**

*f***:=открыть на запись**(**"2.txt"**)

**нц** **для** *i* **от** **1** **до** *k*

**вывод** *f*, **нс**, *Rx*[*i*],**" "**,*Ry*[*i*], **нс**

**кц**

**закрыть**(*f*)

**все**

**кон**

**Лабораторная работа №3.**

Алгоритмизация и программная реализация построения сплайновых поверхностей и трехмерных кривых на основе дельта-преобразований второго порядка.

**Вариант №1.** Интерполирующая *трехмерная кривая* на основе дельта-сплайнов с изменением шага дискретизации в функции от *t* между узлами интерполяции.Псевдокод.

**использовать** **Файлы**

**файл** *outf*

**цел** *i*, *j*, *k*

**вещ** *current*, *swsign*, *t*

**вещ** *T*, *L*, *P*, *D*, *step*

**цел** *count*, *elecount*

**алг**

**нач**

**вывод** **"Введите число интервалов: "**

**ввод** *k*

**вывод** **"Введите число точек: "**

**ввод** *count*

**вещтаб** *valX*[**1**:*count*]

**вещтаб** *valY*[**1**:*count*]

**вещтаб** *valZ*[**1**:*count*]

**вещтаб** *valT*[**1**:*count*]

**нц** **для** *i* **от** **1** **до** *count*

*valT*[*i*] **:=** *i*

**вывод** **"Введите координату X точки "**, *i*, **": "**

**ввод** *valX*[*i*]

**вывод** **"Введите координату Y точки "**, *i*, **": "**

**ввод** *valY*[*i*]

**вывод** **"Введите координату Z точки "**, *i*, **": "**

**ввод** *valZ*[*i*]

**кц**

*elecount* **:=** (*count* - **1**)\**k* + **1**;

**вещтаб** *derX*[**1**:*count*]

**вещтаб** *derY*[**1**:*count*]

**вещтаб** *derZ*[**1**:*count*]

**вещтаб** *resX*[**1**:*elecount*]

**вещтаб** *resY*[**1**:*elecount*]

**вещтаб** *resZ*[**1**:*elecount*]

**| Зададим значения производных в крайних точках**

*derX*[**1**] **:=** **0**

*derY*[**1**] **:=** **0**

*derZ*[**1**] **:=** **0**

*derX*[*count*] **:=** **0**

*derY*[*count*] **:=** **0**

*derZ*[*count*] **:=** **0**

**| Найдём значения производных в остальных точках**

**нц** **для** *i* **от** **2** **до** (*count*-**1**)

**если** (((*valX*[*i*-**1**] <= *valX*[*i*]) **и** (*valX*[*i*+**1**] <= *valX*[*i*])) **или** ((*valX*[*i*-**1**] >= *valX*[*i*]) **и** (*valX*[*i*+**1**] >= *valX*[*i*])))

**то** *derX*[*i*] **:=** **0**

**иначе** *derX*[*i*] **:=** **0.5** \* ((*valX*[*i*] - *valX*[*i*-**1**])/(*valT*[*i*] - *valT*[*i*-**1**]) + (*valX*[*i*+**1**] - *valY*[*i*])/(*valT*[*i*+**1**] - *valT*[*i*]))

**все**

**если** (((*valY*[*i*-**1**] <= *valY*[*i*]) **и** (*valY*[*i*+**1**] <= *valY*[*i*])) **или** ((*valY*[*i*-**1**] >= *valY*[*i*]) **и** (*valY*[*i*+**1**] >= *valY*[*i*])))

**то** *derY*[*i*] **:=** **0**

**иначе** *derY*[*i*] **:=** **0.5** \* ((*valY*[*i*] - *valY*[*i*-**1**])/(*valT*[*i*] - *valT*[*i*-**1**]) + (*valY*[*i*+**1**] - *valY*[*i*])/(*valT*[*i*+**1**] - *valT*[*i*]))

**все**

**если** (((*valZ*[*i*-**1**] <= *valZ*[*i*]) **и** (*valZ*[*i*+**1**] <= *valZ*[*i*])) **или** ((*valZ*[*i*-**1**] >= *valZ*[*i*]) **и** (*valZ*[*i*+**1**] >= *valZ*[*i*])))

**то** *derZ*[*i*] **:=** **0**

**иначе** *derZ*[*i*] **:=** **0.5** \* ((*valZ*[*i*] - *valZ*[*i*-**1**])/(*valT*[*i*] - *valT*[*i*-**1**]) + (*valZ*[*i*+**1**] - *valZ*[*i*])/(*valT*[*i*+**1**] - *valT*[*i*]))

**все**

**кц**

**| Построим сплайн для X**

*j***:=1**;

**нц** **для** *i* **от** **1** **до** (*count*-**1**)

*T***:=***valT*[*i*+**1**]-*valT*[*i*];**|интервал между узлами интерполяции (Tn = tn+1 - tn)**

**если** (*T*=**0**) **то** *P***:=0**; *D***:=0**;

**иначе**

*L***:=***valX*[*i*+**1**]-*valX*[*i*]-**0.5**\**T*\*(*derX*[*i*+**1**]+*derX*[*i*]);**|расстояние от начального узла до точки переключения знака**

*P***:=** (-**1**\**L*-***sign***(*L*)\****sqrt***(*L*\**L*+**0.25**\**T*\**T*\*(*derX*[*i*+**1**]-*derX*[*i*])\*(*derX*[*i*+**1**]-*derX*[*i*])))/(**0.5**\**T*\**T*); **|модуль кванта преобразования**

**если** (*P*=**0**) **или** (*L*=**0**) **то** D**:=0**

**иначе** *D***:=**(*derX*[*i*]-*derX*[*i*+**1**]+*T*\**P*)/(**2**\**P*); **| D - расстояние до переключения знака**

**все**

**все**

*current***:=***valT*[*i*];

**если** (*D*<>**0**) **то** *swsign***:=** *valT*[*i*]+*D* **|координата переключения знака**

**иначе** *swsign***:=** *valT*[*i*]+ (*valT*[*i*+**1**]-*valT*[*i*])/**2**;

**все**

*step* **:=** (*valT*[*i*+**1**]-*current*)/*k*; **|считаем шаг интерполяции**

**нц** **пока** *current*<*valT*[*i*+**1**]

*t***:=***current*-*valT*[*i*];

**если** *current* <= *swsign* **то** *resX*[*j*]**:=** *valX*[*i*]+*derX*[*i*]\**t*-*t*\**t*\**P*/**2**

**иначе** *resX*[*j*]**:=***valX*[*i*+**1**]-*derX*[*i*+**1**]\*(*T*-*t*)+(*T*-*t*)\*(*T*-*t*)\**P*/**2**;

**все**

**|вывод j, " ", resX[j], нс**

**если** (*current*+*step* >*valT*[*i*+**1**] ) **то** *current***:=** *valT*[*i*+**1**]

**иначе** *current***:=***current*+*step*;

**все**

*j***:=***j*+**1**;

**кц**

**если** (*j* = *elecount*) **то** *resX*[*j*] **:=** *valX*[*i*+**1**]

**|вывод j, " ", resX[j], нс**

**все**

**кц**

**| Построим сплайн для Y**

*j***:=1**;

**нц** **для** *i* **от** **1** **до** (*count*-**1**)

*T***:=***valT*[*i*+**1**]-*valT*[*i*];**|интервал между узлами интерполяции (Tn = tn+1 - tn)**

**если** (*T*=**0**) **то** *P***:=0**; *D***:=0**;

**иначе**

*L***:=***valY*[*i*+**1**]-*valY*[*i*]-**0.5**\**T*\*(*derY*[*i*+**1**]+*derY*[*i*]);**|расстояние от начального узла до точки переключения знака**

*P***:=** (-**1**\**L*-***sign***(*L*)\****sqrt***(*L*\**L*+**0.25**\**T*\**T*\*(*derY*[*i*+**1**]-*derY*[*i*])\*(*derY*[*i*+**1**]-*derY*[*i*])))/(**0.5**\**T*\**T*); **|модуль кванта преобразования**

**если** (*P*=**0**) **или** (*L*=**0**) **то** D**:=0**

**иначе** *D***:=**(*derY*[*i*]-*derY*[*i*+**1**]+*T*\**P*)/(**2**\**P*); **| D - расстояние до переключения знака**

**все**

**все**

*current***:=***valT*[*i*];

**если** (*D*<>**0**) **то** *swsign***:=** *valT*[*i*]+*D* **|координата переключения знака**

**иначе** *swsign***:=** *valT*[*i*]+ (*valT*[*i*+**1**]-*valT*[*i*])/**2**;

**все**

*step* **:=** (*valT*[*i*+**1**]-*current*)/*k*; **|считаем шаг интерполяции**

**нц** **пока** *current*<*valT*[*i*+**1**]

*t***:=***current*-*valT*[*i*];

**если** *current* <= *swsign* **то** *resY*[*j*]**:=** *valY*[*i*]+*derY*[*i*]\**t*-*t*\**t*\**P*/**2**

**иначе** *resY*[*j*]**:=***valY*[*i*+**1**]-*derY*[*i*+**1**]\*(*T*-*t*)+(*T*-*t*)\*(*T*-*t*)\**P*/**2**;

**все**

**если** (*current*+*step* >*valT*[*i*+**1**] ) **то** *current***:=** *valT*[*i*+**1**]

**иначе** *current***:=***current*+*step*;

**все**

*j***:=***j*+**1**;

**кц**

**если** (*j* = *elecount*) **то** *resY*[*j*] **:=** *valY*[*i*+**1**] **все**

**кц**

**| Построим сплайн для Z**

*j***:=1**;

**нц** **для** *i* **от** **1** **до** (*count*-**1**)

*T***:=***valT*[*i*+**1**]-*valT*[*i*];**|интервал между узлами интерполяции (Tn = tn+1 - tn)**

**если** (*T*=**0**) **то** *P***:=0**; *D***:=0**;

**иначе**

*L***:=***valZ*[*i*+**1**]-*valZ*[*i*]-**0.5**\**T*\*(*derZ*[*i*+**1**]+*derZ*[*i*]);**|расстояние от начального узла до точки переключения знака**

*P***:=** (-**1**\**L*-***sign***(*L*)\****sqrt***(*L*\**L*+**0.25**\**T*\**T*\*(*derZ*[*i*+**1**]-*derZ*[*i*])\*(*derZ*[*i*+**1**]-*derZ*[*i*])))/(**0.5**\**T*\**T*); **|модуль кванта преобразования**

**если** (*P*=**0**) **или** (*L*=**0**) **то** D**:=0**

**иначе** *D***:=**(*derZ*[*i*]-*derZ*[*i*+**1**]+*T*\**P*)/(**2**\**P*); **| D - расстояние до переключения знака**

**все**

**все**

*current***:=***valT*[*i*];

**если** (*D*<>**0**) **то** *swsign***:=** *valT*[*i*]+*D* **|координата переключения знака**

**иначе** *swsign***:=** *valT*[*i*]+ (*valT*[*i*+**1**]-*valT*[*i*])/**2**;

**все**

*step* **:=** (*valT*[*i*+**1**]-*current*)/*k*; **|считаем шаг интерполяции**

**нц** **пока** *current*<*valT*[*i*+**1**]

*t***:=***current*-*valT*[*i*];

**если** *current* <= *swsign* **то** *resZ*[*j*]**:=** *valZ*[*i*]+*derZ*[*i*]\**t*-*t*\**t*\**P*/**2**

**иначе** *resZ*[*j*]**:=***valZ*[*i*+**1**]-*derZ*[*i*+**1**]\*(*T*-*t*)+(*T*-*t*)\*(*T*-*t*)\**P*/**2**;

**все**

**если** (*current*+*step* >*valT*[*i*+**1**] ) **то** *current***:=** *valT*[*i*+**1**]

**иначе** *current***:=***current*+*step*;

**все**

*j***:=***j*+**1**;

**кц**

**если** (*j* = *elecount*) **то** *resZ*[*j*] **:=** *valZ*[*i*+**1**] **все**

**кц**

**| Массив resX содержит координаты Х точек**

**| Массив resY содержит координаты Y точек**

**| Массив resZ содержит координаты Z точек**

*outf* **:=** **открыть на запись**(**"output1.txt"**)

**нц** **для** *i* **от** **1** **до** *elecount*

**вывод** *outf*, *resX*[*i*], **','**, *resY*[*i*], **','**, *resZ*[*i*], **нс**;

**вывод** *resX*[*i*], **','**, *resY*[*i*], **','**, *resZ*[*i*], **нс**;

**кц**

**закрыть** (*outf*)

**кон**

**Вариант №2.** Создать программу, позволяющую строить интерполирующую *трехмерную кривую* на основе дельта-сплайнов с постоянным шагом дискретизации  между узлами интерполяции с использованием операций многоразрядного умножения. Псевдокод.

**использовать** **Файлы**

**файл** *outf*

**цел** *i*, *j*, *k*, *na*, *nb*

**вещ** *d*, *dt*, *l*, *p*, *Tn*

**цел** *count*, *elecount*

**вещ** *dn*, *cn*, *dZ*, *dTna*, *dTnai1nt*, *dT1nt*

**цел** *lx*, *elemX*, *elemY*

**алг**

**нач**

**вывод** **"Введите число интервалов: "**

**ввод** *k*

**вывод** **"Введите число точек: "**

**ввод** *count*

**вещтаб** *valX*[**0**:*count*-**1**]

**вещтаб** *valY*[**0**:*count*-**1**]

**вещтаб** *valZ*[**0**:*count*-**1**]

**вещтаб** *valT*[**0**:*count*-**1**]

**нц** **для** *i* **от** **0** **до** *count*-**1**

*valT*[*i*] **:=** *i*

**вывод** **"Введите координату X точки "**, *i*+**1**, **": "**

**ввод** *valX*[*i*]

**вывод** **"Введите координату Y точки "**, *i*+**1**, **": "**

**ввод** *valY*[*i*]

**вывод** **"Введите координату Z точки "**, *i*+**1**, **": "**

**ввод** *valZ*[*i*]

**кц**

*elecount* **:=** (*count* - **1**)\**k* + **1**;

**вещтаб** *derX*[**0**:*count*-**1**]

**вещтаб** *derY*[**0**:*count*-**1**]

**вещтаб** *derZ*[**0**:*count*-**1**]

**вещтаб** *resX*[**0**:*elecount*-**1**]

**вещтаб** *resY*[**0**:*elecount*-**1**]

**вещтаб** *resZ*[**0**:*elecount*-**1**]

**| Зададим значения производных в крайних точках**

*derX*[**0**] **:=** **0**

*derY*[**0**] **:=** **0**

*derZ*[**0**] **:=** **0**

*derX*[*count*-**1**] **:=** **0**

*derY*[*count*-**1**] **:=** **0**

*derZ*[*count*-**1**] **:=** **0**

**| Найдём значения производных в остальных точках**

**нц** **для** *i* **от** **1** **до** (*count*-**2**)

**если** (((*valX*[*i*-**1**] <= *valX*[*i*]) **и** (*valX*[*i*+**1**] <= *valX*[*i*])) **или** ((*valX*[*i*-**1**] >= *valX*[*i*]) **и** (*valX*[*i*+**1**] >= *valX*[*i*])))

**то** *derX*[*i*] **:=** **0**

**иначе** *derX*[*i*] **:=** **0.5** \* ((*valX*[*i*] - *valX*[*i*-**1**])/(*valT*[*i*] - *valT*[*i*-**1**]) + (*valX*[*i*+**1**] - *valY*[*i*])/(*valT*[*i*+**1**] - *valT*[*i*]))

**все**

**если** (((*valY*[*i*-**1**] <= *valY*[*i*]) **и** (*valY*[*i*+**1**] <= *valY*[*i*])) **или** ((*valY*[*i*-**1**] >= *valY*[*i*]) **и** (*valY*[*i*+**1**] >= *valY*[*i*])))

**то** *derY*[*i*] **:=** **0**

**иначе** *derY*[*i*] **:=** **0.5** \* ((*valY*[*i*] - *valY*[*i*-**1**])/(*valT*[*i*] - *valT*[*i*-**1**]) + (*valY*[*i*+**1**] - *valY*[*i*])/(*valT*[*i*+**1**] - *valT*[*i*]))

**все**

**если** (((*valZ*[*i*-**1**] <= *valZ*[*i*]) **и** (*valZ*[*i*+**1**] <= *valZ*[*i*])) **или** ((*valZ*[*i*-**1**] >= *valZ*[*i*]) **и** (*valZ*[*i*+**1**] >= *valZ*[*i*])))

**то** *derZ*[*i*] **:=** **0**

**иначе** *derZ*[*i*] **:=** **0.5** \* ((*valZ*[*i*] - *valZ*[*i*-**1**])/(*valT*[*i*] - *valT*[*i*-**1**]) + (*valZ*[*i*+**1**] - *valZ*[*i*])/(*valT*[*i*+**1**] - *valT*[*i*]))

**все**

**кц**

**| Построим сплайн для X**

**нц** **для** *i* **от** **0** **до** (*count* - **2**)

*Tn* **:=** *valT*[*i*+**1**] - *valT*[*i*] **| интервал между узлами**

*l* **:=** *valX*[*i*+**1**] - *valX*[*i*] - **0.5**\**Tn*\*(*derX*[*i*+**1**]+*derX*[*i*])

*p* **:=** (-**1**\**l* - ***sign***(*l*)\****sqrt***(*l*\**l* + **0.25**\*(*Tn*\*\***2**)\*((*derX*[*i*] - *derX*[*i*+**1**])\*\***2**)))/(**0.5**\**Tn*\**Tn*)

**если** (*p* = **0**)

**то** *d* **:=** **0**

**иначе** *d* **:=** (*derX*[*i*+**1**] - *derX*[*i*] + *Tn*\**p*)/(**2**\**p*)

**все**

*dt* **:=** (*valT*[*i*+**1**] - *valT*[*i*])/*k* **| Шаг интерполяции**

*dn* **:=** ***sign***(*p*)

*cn* **:=** ***abs***(*p*)\**dt*\**dt*

**| Построим сплайн на участке**

*na* **:=** ***int***(*d*/*dt*)

*nb* **:=** ***int***((*Tn* - *d*)/*dt*)

*resX*[*i*\**k* + *k*] **:=** *valX*[*i*+**1**] **| Yna = Y(n+1)**

*dTna* **:=** *derX*[*i*+**1**]

**нц** **для** *j* **от** **0** **до** *na*-**1**

*dTnai1nt* **:=** *dTna*\**dt* - *dn*\**cn*

*dTna* **:=** *dTnai1nt*/*dt*

*resX*[*i*\**k*+*k*-*j*-**1**] **:=** *resX*[*i*\**k*+*k*-*j*] - *dTnai1nt* - **0.5**\**dn*\**cn*

**кц**

*dn* **:=** -***sign***(*p*)

*resX*[*i*\**k*] **:=** *valX*[*i*] **| Y0 = Yn**

*dZ* **:=** *derX*[*i*]

**нц** **для** *j* **от** **0** **до** *nb*-**1**

*dT1nt* **:=** *dZ*\**dt* + *dn*\**cn*

*dZ* **:=** *dT1nt*/*dt*

*resX*[*i*\**k* + *j* + **1**] **:=** *resX*[*i*\**k* + *j*] + *dT1nt* - **0.5**\**dn*\**cn*

**кц**

**кц**

**| Построим сплайн для Y**

**нц** **для** *i* **от** **0** **до** (*count* - **2**)

*Tn* **:=** *valT*[*i*+**1**] - *valT*[*i*] **| интервал между узлами**

*l* **:=** *valY*[*i*+**1**] - *valY*[*i*] - **0.5**\**Tn*\*(*derY*[*i*+**1**]+*derY*[*i*])

*p* **:=** (-**1**\**l* - ***sign***(*l*)\****sqrt***(*l*\**l* + **0.25**\*(*Tn*\*\***2**)\*((*derY*[*i*] - *derY*[*i*+**1**])\*\***2**)))/(**0.5**\**Tn*\**Tn*)

**если** (*p* = **0**)

**то** *d* **:=** **0**

**иначе** *d* **:=** (*derY*[*i*+**1**] - *derY*[*i*] + *Tn*\**p*)/(**2**\**p*)

**все**

*dt* **:=** (*valT*[*i*+**1**] - *valT*[*i*])/*k* **| Шаг интерполяции**

*dn* **:=** ***sign***(*p*)

*cn* **:=** ***abs***(*p*)\**dt*\**dt*

**| Построим сплайн на участке**

*na* **:=** ***int***(*d*/*dt*)

*nb* **:=** ***int***((*Tn* - *d*)/*dt*)

*resY*[*i*\**k* + *k*] **:=** *valY*[*i*+**1**] **| Yna = Y(n+1)**

*dTna* **:=** *derY*[*i*+**1**]

**нц** **для** *j* **от** **0** **до** *na*-**1**

*dTnai1nt* **:=** *dTna*\**dt* - *dn*\**cn*

*dTna* **:=** *dTnai1nt*/*dt*

*resY*[*i*\**k*+*k*-*j*-**1**] **:=** *resY*[*i*\**k*+*k*-*j*] - *dTnai1nt* - **0.5**\**dn*\**cn*

**кц**

*dn* **:=** -***sign***(*p*)

*resY*[*i*\**k*] **:=** *valY*[*i*] **| Y0 = Yn**

*dZ* **:=** *derY*[*i*]

**нц** **для** *j* **от** **0** **до** *nb*-**1**

*dT1nt* **:=** *dZ*\**dt* + *dn*\**cn*

*dZ* **:=** *dT1nt*/*dt*

*resY*[*i*\**k* + *j* + **1**] **:=** *resY*[*i*\**k* + *j*] + *dT1nt* - **0.5**\**dn*\**cn*

**кц**

**кц**

**| Построим сплайн для Z**

**нц** **для** *i* **от** **0** **до** (*count* - **2**)

*Tn* **:=** *valT*[*i*+**1**] - *valT*[*i*] **| интервал между узлами**

*l* **:=** *valZ*[*i*+**1**] - *valZ*[*i*] - **0.5**\**Tn*\*(*derZ*[*i*+**1**]+*derZ*[*i*])

*p* **:=** (-**1**\**l* - ***sign***(*l*)\****sqrt***(*l*\**l* + **0.25**\*(*Tn*\*\***2**)\*((*derZ*[*i*] - *derZ*[*i*+**1**])\*\***2**)))/(**0.5**\**Tn*\**Tn*)

**если** (*p* = **0**)

**то** *d* **:=** **0**

**иначе** *d* **:=** (*derZ*[*i*+**1**] - *derZ*[*i*] + *Tn*\**p*)/(**2**\**p*)

**все**

*dt* **:=** (*valT*[*i*+**1**] - *valT*[*i*])/*k* **| Шаг интерполяции**

*dn* **:=** ***sign***(*p*)

*cn* **:=** ***abs***(*p*)\**dt*\**dt*

**| Построим сплайн на участке**

*na* **:=** ***int***(*d*/*dt*)

*nb* **:=** ***int***((*Tn* - *d*)/*dt*)

*resZ*[*i*\**k* + *k*] **:=** *valZ*[*i*+**1**] **| Yna = Y(n+1)**

*dTna* **:=** *derZ*[*i*+**1**]

**нц** **для** *j* **от** **0** **до** *na*-**1**

*dTnai1nt* **:=** *dTna*\**dt* - *dn*\**cn*

*dTna* **:=** *dTnai1nt*/*dt*

*resZ*[*i*\**k*+*k*-*j*-**1**] **:=** *resZ*[*i*\**k*+*k*-*j*] - *dTnai1nt* - **0.5**\**dn*\**cn*

**кц**

*dn* **:=** -***sign***(*p*)

*resZ*[*i*\**k*] **:=** *valZ*[*i*] **| Y0 = Yn**

*dZ* **:=** *derZ*[*i*]

**нц** **для** *j* **от** **0** **до** *nb*-**1**

*dT1nt* **:=** *dZ*\**dt* + *dn*\**cn*

*dZ* **:=** *dT1nt*/*dt*

*resZ*[*i*\**k* + *j* + **1**] **:=** *resZ*[*i*\**k* + *j*] + *dT1nt* - **0.5**\**dn*\**cn*

**кц**

**кц**

**| Массив resX содержит координаты Х точек**

**| Массив resY содержит координаты Y точек**

**| Массив resZ содержит координаты Z точек**

*outf* **:=** **открыть на запись**(**"output2.txt"**)

**нц** **для** *i* **от** **0** **до** *elecount*-**1**

**вывод** *outf*, *resX*[*i*], **','**, *resY*[*i*], **','**, *resZ*[*i*], **нс**;

**вывод** *resX*[*i*], **','**, *resY*[*i*], **','**, *resZ*[*i*], **нс**;

**кц**

**закрыть** (*outf*)

**кон**

**Вариант №3.** Создать программу, позволяющую строить интерполирующую *трехмерную кривую* на основе дельта-сплайнов с постоянным шагом дискретизации  между узлами интерполяции без использования операций многоразрядного умножения. Псевдокод.

**использовать** **Файлы**

**файл** *outf*

**цел** *i*, *j*, *k*, *na*, *nb*

**вещ** *d*, *dt*, *l*, *p*, *Tn*

**вещ** *csn*, *nSna*, *nSnai1*, *nS*, *nSi*

**цел** *count*, *ly*, *elecount*

**алг**

**нач**

**вывод** **"Введите число интервалов: "**

**ввод** *k*

**вывод** **"Введите число точек: "**

**ввод** *count*

**вещтаб** *valX*[**0**:*count*-**1**]

**вещтаб** *valY*[**0**:*count*-**1**]

**вещтаб** *valZ*[**0**:*count*-**1**]

**вещтаб** *valT*[**0**:*count*-**1**]

**нц** **для** *i* **от** **0** **до** *count*-**1**

*valT*[*i*] **:=** *i*

**вывод** **"Введите координату X точки "**, *i*+**1**, **": "**

**ввод** *valX*[*i*]

**вывод** **"Введите координату Y точки "**, *i*+**1**, **": "**

**ввод** *valY*[*i*]

**вывод** **"Введите координату Z точки "**, *i*+**1**, **": "**

**ввод** *valZ*[*i*]

**кц**

*elecount* **:=** (*count* - **1**)\**k* + **1**;

**вещтаб** *derX*[**0**:*count*-**1**]

**вещтаб** *derY*[**0**:*count*-**1**]

**вещтаб** *derZ*[**0**:*count*-**1**]

**вещтаб** *resX*[**0**:*elecount*-**1**]

**вещтаб** *resY*[**0**:*elecount*-**1**]

**вещтаб** *resZ*[**0**:*elecount*-**1**]

**| Зададим значения производных в крайних точках**

*derX*[**0**] **:=** **0**

*derY*[**0**] **:=** **0**

*derZ*[**0**] **:=** **0**

*derX*[*count*-**1**] **:=** **0**

*derY*[*count*-**1**] **:=** **0**

*derZ*[*count*-**1**] **:=** **0**

**| Найдём значения производных в остальных точках**

**нц** **для** *i* **от** **1** **до** (*count*-**2**)

**если** (((*valX*[*i*-**1**] <= *valX*[*i*]) **и** (*valX*[*i*+**1**] <= *valX*[*i*])) **или** ((*valX*[*i*-**1**] >= *valX*[*i*]) **и** (*valX*[*i*+**1**] >= *valX*[*i*])))

**то** *derX*[*i*] **:=** **0**

**иначе** *derX*[*i*] **:=** **0.5** \* ((*valX*[*i*] - *valX*[*i*-**1**])/(*valT*[*i*] - *valT*[*i*-**1**]) + (*valX*[*i*+**1**] - *valY*[*i*])/(*valT*[*i*+**1**] - *valT*[*i*]))

**все**

**если** (((*valY*[*i*-**1**] <= *valY*[*i*]) **и** (*valY*[*i*+**1**] <= *valY*[*i*])) **или** ((*valY*[*i*-**1**] >= *valY*[*i*]) **и** (*valY*[*i*+**1**] >= *valY*[*i*])))

**то** *derY*[*i*] **:=** **0**

**иначе** *derY*[*i*] **:=** **0.5** \* ((*valY*[*i*] - *valY*[*i*-**1**])/(*valT*[*i*] - *valT*[*i*-**1**]) + (*valY*[*i*+**1**] - *valY*[*i*])/(*valT*[*i*+**1**] - *valT*[*i*]))

**все**

**если** (((*valZ*[*i*-**1**] <= *valZ*[*i*]) **и** (*valZ*[*i*+**1**] <= *valZ*[*i*])) **или** ((*valZ*[*i*-**1**] >= *valZ*[*i*]) **и** (*valZ*[*i*+**1**] >= *valZ*[*i*])))

**то** *derZ*[*i*] **:=** **0**

**иначе** *derZ*[*i*] **:=** **0.5** \* ((*valZ*[*i*] - *valZ*[*i*-**1**])/(*valT*[*i*] - *valT*[*i*-**1**]) + (*valZ*[*i*+**1**] - *valZ*[*i*])/(*valT*[*i*+**1**] - *valT*[*i*]))

**все**

**кц**

**| Построим сплайн для X**

**нц** **для** *i* **от** **0** **до** (*count* - **2**)

*Tn* **:=** *valT*[*i*+**1**] - *valT*[*i*] **| интервал между узлами**

*l* **:=** *valX*[*i*+**1**] - *valX*[*i*] - **0.5**\**Tn*\*(*derX*[*i*+**1**]+*derX*[*i*])

*p* **:=** (-**1**\**l* - ***sign***(*l*)\****sqrt***(*l*\**l* + **0.25**\*(*Tn*\*\***2**)\*((*derX*[*i*] - *derX*[*i*+**1**])\*\***2**)))/(**0.5**\**Tn*\**Tn*)

**если** (*p* = **0**)

**то** *d* **:=** **0**

**иначе** *d* **:=** (*derX*[*i*+**1**] - *derX*[*i*] + *Tn*\**p*)/(**2**\**p*)

**все**

*dt* **:=** (*valT*[*i*+**1**] - *valT*[*i*])/*k* **| Шаг интерполяции**

**| Построим сплайн на участке**

*na* **:=** ***int***(*d*/*dt*)

*nb* **:=** ***int***((*Tn* - *d*)/*dt*)

*csn* **:=** *p*\**dt*\**dt*

*nSna* **:=** *derX*[*i*+**1**]\**dt*

*nS* **:=** *derX*[*i*]\**dt*

*resX*[*i*\**k* + *k*] **:=** *valX*[*i*+**1**] **| Yna = Y(n+1)**

**нц** **для** *j* **от** **0** **до** *na*-**1**

*nSnai1* **:=** *nSna* - *csn*

*nSna* **:=** *nSnai1*

*resX*[*i*\**k*+*k*-*j*-**1**] **:=** *resX*[*i*\**k*+*k*-*j*] - *nSnai1* - **0.5**\**csn*

**кц**

*resX*[*i*\**k*] **:=** *valX*[*i*] **| Y0 = Yn**

**нц** **для** *j* **от** **0** **до** *nb*-**1**

*nSi* **:=** *nS* - *csn*

*nS* **:=** *nSi*

*resX*[*i*\**k* + *j* + **1**] **:=** *resX*[*i*\**k* + *j*] + *nSi* + **0.5**\**csn*

**кц**

**кц**

**| Построим сплайн для Y**

**нц** **для** *i* **от** **0** **до** (*count* - **2**)

*Tn* **:=** *valT*[*i*+**1**] - *valT*[*i*] **| интервал между узлами**

*l* **:=** *valY*[*i*+**1**] - *valY*[*i*] - **0.5**\**Tn*\*(*derY*[*i*+**1**]+*derY*[*i*])

*p* **:=** (-**1**\**l* - ***sign***(*l*)\****sqrt***(*l*\**l* + **0.25**\*(*Tn*\*\***2**)\*((*derY*[*i*] - *derY*[*i*+**1**])\*\***2**)))/(**0.5**\**Tn*\**Tn*)

**если** (*p* = **0**)

**то** *d* **:=** **0**

**иначе** *d* **:=** (*derY*[*i*+**1**] - *derY*[*i*] + *Tn*\**p*)/(**2**\**p*)

**все**

*dt* **:=** (*valT*[*i*+**1**] - *valT*[*i*])/*k* **| Шаг интерполяции**

**| Построим сплайн на участке**

*na* **:=** ***int***(*d*/*dt*)

*nb* **:=** ***int***((*Tn* - *d*)/*dt*)

*csn* **:=** *p*\**dt*\**dt*

*nSna* **:=** *derY*[*i*+**1**]\**dt*

*nS* **:=** *derY*[*i*]\**dt*

*resY*[*i*\**k* + *k*] **:=** *valY*[*i*+**1**] **| Yna = Y(n+1)**

**нц** **для** *j* **от** **0** **до** *na*-**1**

*nSnai1* **:=** *nSna* - *csn*

*nSna* **:=** *nSnai1*

*resY*[*i*\**k*+*k*-*j*-**1**] **:=** *resY*[*i*\**k*+*k*-*j*] - *nSnai1* - **0.5**\**csn*

**кц**

*resY*[*i*\**k*] **:=** *valY*[*i*] **| Y0 = Yn**

**нц** **для** *j* **от** **0** **до** *nb*-**1**

*nSi* **:=** *nS* - *csn*

*nS* **:=** *nSi*

*resY*[*i*\**k* + *j* + **1**] **:=** *resY*[*i*\**k* + *j*] + *nSi* + **0.5**\**csn*

**кц**

**кц**

**| Построим сплайн для Z**

**нц** **для** *i* **от** **0** **до** (*count* - **2**)

*Tn* **:=** *valT*[*i*+**1**] - *valT*[*i*] **| интервал между узлами**

*l* **:=** *valZ*[*i*+**1**] - *valZ*[*i*] - **0.5**\**Tn*\*(*derZ*[*i*+**1**]+*derZ*[*i*])

*p* **:=** (-**1**\**l* - ***sign***(*l*)\****sqrt***(*l*\**l* + **0.25**\*(*Tn*\*\***2**)\*((*derZ*[*i*] - *derZ*[*i*+**1**])\*\***2**)))/(**0.5**\**Tn*\**Tn*)

**если** (*p* = **0**)

**то** *d* **:=** **0**

**иначе** *d* **:=** (*derZ*[*i*+**1**] - *derZ*[*i*] + *Tn*\**p*)/(**2**\**p*)

**все**

*dt* **:=** (*valT*[*i*+**1**] - *valT*[*i*])/*k* **| Шаг интерполяции**

**| Построим сплайн на участке**

*na* **:=** ***int***(*d*/*dt*)

*nb* **:=** ***int***((*Tn* - *d*)/*dt*)

*csn* **:=** *p*\**dt*\**dt*

*nSna* **:=** *derZ*[*i*+**1**]\**dt*

*nS* **:=** *derZ*[*i*]\**dt*

*resZ*[*i*\**k* + *k*] **:=** *valZ*[*i*+**1**] **| Yna = Y(n+1)**

**нц** **для** *j* **от** **0** **до** *na*-**1**

*nSnai1* **:=** *nSna* - *csn*

*nSna* **:=** *nSnai1*

*resZ*[*i*\**k*+*k*-*j*-**1**] **:=** *resZ*[*i*\**k*+*k*-*j*] - *nSnai1* - **0.5**\**csn*

**кц**

*resZ*[*i*\**k*] **:=** *valZ*[*i*] **| Y0 = Yn**

**нц** **для** *j* **от** **0** **до** *nb*-**1**

*nSi* **:=** *nS* - *csn*

*nS* **:=** *nSi*

*resZ*[*i*\**k* + *j* + **1**] **:=** *resZ*[*i*\**k* + *j*] + *nSi* + **0.5**\**csn*

**кц**

**кц**

**| Массив resX содержит координаты Х точек**

**| Массив resY содержит координаты Y точек**

**| Массив resZ содержит координаты Z точек**

*outf* **:=** **открыть на запись**(**"output3.txt"**)

**нц** **для** *i* **от** **0** **до** *elecount*-**1**

**вывод** *outf*, *resX*[*i*], **','**, *resY*[*i*], **','**, *resZ*[*i*], **нс**;

**вывод** *resX*[*i*], **','**, *resY*[*i*], **','**, *resZ*[*i*], **нс**;

**кц**

**закрыть** (*outf*)

**кон**

**Вариант №4.** Создать программу, позволяющую строить интерполирующую *поверхность* на основе дельта-сплайнов с изменением шага дискретизации в функции от *t* между узлами интерполяции. Псевдокод.

**использовать** **Файлы**

**файл** *outf*

**цел** *i*, *j*, *ix*, *iy*, *k*

**вещ** *current*, *d*, *dx*, *dy*, *l*, *na*, *p*, *swsign*, *t*, *Tn*

**цел** *lx*, *ly*, *elemX*, *elemY*

**алг**

**нач**

**вывод** **"Введите число интервалов: "**

**ввод** *k*

**вывод** **"Введите число шагов по X: "**

**ввод** *lx*

**вещтаб** *valX*[**0**:*lx*-**1**]

**нц** **для** *i* **от** **0** **до** *lx*-**1**

**вывод** **"Введите значение X"**, *i*, **": "**

**ввод** *valX*[*i*]

**кц**

**вывод** **"Введите число шагов по Y: "**

**ввод** *ly*

**вещтаб** *valY*[**0**:*ly*-**1**]

**нц** **для** *i* **от** **0** **до** *ly*-**1**

**вывод** **"Введите значение Y"**, *i*, **": "**

**ввод** *valY*[*i*]

**кц**

**вещтаб** *valZ*[**0**:*lx*-**1**, **0**:*ly*-**1**]

**нц** **для** *i* **от** **0** **до** *lx*-**1**

**нц** **для** *j* **от** **0** **до** *ly*-**1**

**вывод** **"Введите значение точки ("**, *i*, **","**, *j*, **"): "**

**ввод** *valZ*[*i*,*j*]

**кц**

**кц**

*elemX* **:=** (*lx* - **1**)\**k* + **1**;

*elemY* **:=** (*ly* - **1**)\**k* + **1**;

**вещтаб** *derZ*[**0**:*elemX*-**1**, **0**:*elemY*-**1**]

**вещтаб** *resX*[**0**:*elemX*-**1**]

**вещтаб** *resY*[**0**:*elemY*-**1**]

**вещтаб** *resOXZ*[**0**:*elemX*-**1**, **0**:*ly*-**1**]

**вещтаб** *resOYZ*[**0**:*elemX*-**1**, **0**:*elemY*-**1**]

**| Построим одномерные кривые в плоскостях OXZ**

**нц** **для** *iy* **от** **0** **до** *ly* - **1**

**| Зададим значения производных Z в текущей плоскости в крайних точках**

*derZ*[**0**, *iy*] **:=** **0**

*derZ*[*lx* - **1**, *iy*] **:=** **0**

**| Найдём значения производных Z в текущей плоскости в остальных точках**

**нц** **для** *i* **от** **1** **до** (*lx* - **2**)

**если** (((*valZ*[*i*-**1**, *iy*] <= *valZ*[*i*, *iy*]) **и** (*valZ*[*i*+**1**, *iy*] <= *valZ*[*i*, *iy*])) **или** ((*valZ*[*i*-**1**, *iy*] >= *valZ*[*i*, *iy*]) **и** (*valZ*[*i*+**1**, *iy*] >= *valZ*[*i*, *iy*])))

**то** *derZ*[*i*, *iy*] **:=** **0**

**иначе** *derZ*[*i*, *iy*] **:=** **0.5** \* ((*valZ*[*i*, *iy*] - *valZ*[*i*-**1**, *iy*])/(*valX*[*i*] - *valX*[*i*-**1**]) + (*valZ*[*i*+**1**, *iy*] - *valZ*[*i*, *iy*])/(*valX*[*i*+**1**] - *valX*[*i*]))

**все**

**кц**

**| Пройдём по узлам интерполяции**

**нц** **для** *i* **от** **0** **до** (*lx* - **2**)

*Tn* **:=** *valX*[*i*+**1**] - *valX*[*i*] **| интервал между узлами**

*l* **:=** *valZ*[*i*+**1**, *iy*] - *valZ*[*i*, *iy*] - **0.5**\**Tn*\*(*derZ*[*i*+**1**, *iy*]+*derZ*[*i*, *iy*])

*p* **:=** (-**1**\**l* - ***sign***(*l*)\****sqrt***(*l*\**l* + **0.25**\*(*Tn*\*\***2**)\*((*derZ*[*i*, *iy*] - *derZ*[*i*+**1**, *iy*])\*\***2**)))/(**0.5**\**Tn*\**Tn*)

**если** (*p* = **0**)

**то** *d* **:=** **0**

**иначе** *d* **:=** (*derZ*[*i*+**1**, *iy*] - *derZ*[*i*, *iy*] + *Tn*\**p*)/(**2**\**p*)

**все**

*current* **:=** *valX*[*i*]

*dx* **:=** (*valX*[*i*+**1**] - *valX*[*i*])/*k* **| Шаг интерполяции**

*swsign* **:=** *valX*[*i*+**1**] - *d* **| Координата переключения знака**

**| Построим сплайн на участке**

*j* **:=** **0**

**нц** **пока** (*current* <= *valX*[*i*+**1**])

*resX*[*i*\**k* + *j*] **:=** *current*

*t* **:=** *dx*\**j*

**если** (*current* <= *swsign*)

**то** *resOXZ*[*i*\**k* + *j*, *iy*] **:=** *valZ*[*i*, *iy*] + *derZ*[*i*, *iy*]\**t* - *t*\**t*\**p*/**2**

**иначе** *resOXZ*[*i*\**k* + *j*, *iy*] **:=** *valZ*[*i*+**1**, *iy*] - *derZ*[*i*+**1**, *iy*]\*(*Tn*-*t*) + (*Tn*-*t*)\*(*Tn*-*t*)\**p*/**2**

**все**

*current* **:=** *current* + *dx*

*j* **:=** *j* + **1**

**кц**

**кц**

**кц**

**| Построим одномерные кривые в плоскостях OYZ**

**| Данные для построения берутся из OXZ - результатов вычислений на предыдущем шаге**

**нц** **для** *ix* **от** **0** **до** *elemX* - **1**

**| Зададим значения производных Z в текущей плоскости в крайних точках**

*derZ*[*ix*, **0**] **:=** **0**

*derZ*[*ix*, *ly* - **1**] **:=** **0**

**| Найдём значения производных Z в текущей плоскости в остальных точках**

**нц** **для** *i* **от** **1** **до** (*ly* - **2**)

**если** (((*resOXZ*[*ix*, *i*-**1**] <= *resOXZ*[*ix*, *i*]) **и** (*resOXZ*[*ix*, *i*+**1**] <= *resOXZ*[*ix*, *i*])) **или** ((*resOXZ*[*ix*, *i*-**1**] >= *resOXZ*[*ix*, *i*]) **и** (*resOXZ*[*ix*, *i*+**1**] >= *resOXZ*[*ix*, *i*])))

**то** *derZ*[*ix*, *i*] **:=** **0**

**иначе** *derZ*[*ix*, *i*] **:=** **0.5** \* ((*resOXZ*[*ix*, *i*] - *resOXZ*[*ix*, *i*-**1**])/(*valY*[*i*] - *valY*[*i*-**1**]) + (*resOXZ*[*ix*, *i*+**1**] - *resOXZ*[*ix*, *i*])/(*valY*[*i*+**1**] - *valY*[*i*]))

**все**

**кц**

**| Пройдём по узлам интерполяции**

**нц** **для** *i* **от** **0** **до** (*ly* - **2**)

*Tn* **:=** *valY*[*i*+**1**] - *valY*[*i*] **| интервал между узлами**

*l* **:=** *resOXZ*[*ix*, *i*+**1**] - *resOXZ*[*ix*, *i*] - **0.5**\**Tn*\*(*derZ*[*ix*, *i*+**1**]+*derZ*[*ix*, *i*])

*p* **:=** (-**1**\**l* - ***sign***(*l*)\****sqrt***(*l*\**l* + **0.25**\*(*Tn*\*\***2**)\*((*derZ*[*ix*, *i*+**1**] - *derZ*[*ix*, *i*])\*\***2**)))/(**0.5**\**Tn*\**Tn*)

**если** (*p* = **0**)

**то** *d* **:=** **0**

**иначе** *d* **:=** (*derZ*[*ix*, *i*+**1**] - *derZ*[*ix*, *i*] + *Tn*\**p*)/(**2**\**p*)

**все**

*current* **:=** *valY*[*i*]

*dy* **:=** (*valY*[*i*+**1**] - *valY*[*i*])/*k* **| Шаг интерполяции**

*swsign* **:=** *valY*[*i*+**1**] - *d* **| Координата переключения знака**

**| Построим сплайн на участке**

*j* **:=** **0**

**нц** **пока** (*current* <= *valY*[*i*+**1**])

*resY*[*i*\**k* + *j*] **:=** *current*

*t* **:=** *dy*\**j*

**если** (*current* <= *swsign*)

**то** *resOYZ*[*ix*, *i*\**k* + *j*] **:=** *resOXZ*[*ix*, *i*] + *derZ*[*ix*, *i*]\**t* - *t*\**t*\**p*/**2**

**иначе** *resOYZ*[*ix*, *i*\**k* + *j*] **:=** *resOXZ*[*ix*, *i*+**1**] - *derZ*[*ix*, *i*+**1**]\*(*Tn*-*t*) + (*Tn*-*t*)\*(*Tn*-*t*)\**p*/**2**

**все**

*current* **:=** *current* + *dy*

*j* **:=** *j* + **1**

**кц**

**кц**

**кц**

**| Массив resX содержит координаты Х плоскостей OXZ**

**| Массив resY содержит координаты Y плоскостей OYZ**

**| Массив resOYZ содержит значения узлов интерполированной поверхности**

*outf* **:=** **открыть на запись**(**"output4.txt"**)

**|нц для j от 0 до elemX-1**

**|вывод outf, j, ' ', resOYZ[j, 0], нс;**

**|кц**

**нц** **для** *i* **от** **0** **до** *elemY*-**1**

**нц** **для** *j* **от** **0** **до** *elemX*-**1**

**вывод** *outf*, *resOYZ*[*j*, *i*], **' '**;

**кц**

**вывод** *outf*, **нс**

**кц**

**закрыть** (*outf*)

**кон**

**Вариант №5.** Создать программу, позволяющую строить интерполирующую *поверхность* на основе дельта-сплайнов с постоянным шагом дискретизации  между узлами интерполяции с использованием операций многоразрядного умножения. Псевдокод.

**использовать** **Файлы**

**файл** *outf*

**цел** *i*, *j*, *ix*, *iy*, *k*, *na*, *nb*

**вещ** *current*, *d*, *dx*, *dy*, *l*, *p*, *Tn*

**вещ** *dn*, *cn*, *dZ*, *dZna*, *dZnai1nt*, *dZ1nt*

**цел** *lx*, *ly*, *elemX*, *elemY*

**алг**

**нач**

**вывод** **"Введите число интервалов: "**

**ввод** *k*

**вывод** **"Введите число шагов по X: "**

**ввод** *lx*

**вещтаб** *valX*[**0**:*lx*-**1**]

**нц** **для** *i* **от** **0** **до** *lx*-**1**

**вывод** **"Введите значение X"**, *i*, **": "**

**ввод** *valX*[*i*]

**кц**

**вывод** **"Введите число шагов по Y: "**

**ввод** *ly*

**вещтаб** *valY*[**0**:*ly*-**1**]

**нц** **для** *i* **от** **0** **до** *ly*-**1**

**вывод** **"Введите значение Y"**, *i*, **": "**

**ввод** *valY*[*i*]

**кц**

**вещтаб** *valZ*[**0**:*lx*-**1**, **0**:*ly*-**1**]

**нц** **для** *i* **от** **0** **до** *lx*-**1**

**нц** **для** *j* **от** **0** **до** *ly*-**1**

**вывод** **"Введите значение точки ("**, *i*, **","**, *j*, **"): "**

**ввод** *valZ*[*i*,*j*]

**кц**

**кц**

*elemX* **:=** (*lx* - **1**)\**k* + **1**;

*elemY* **:=** (*ly* - **1**)\**k* + **1**;

**вещтаб** *derZ*[**0**:*elemX*-**1**, **0**:*elemY*-**1**]

**вещтаб** *resX*[**0**:*elemX*-**1**]

**вещтаб** *resY*[**0**:*elemY*-**1**]

**вещтаб** *resOXZ*[**0**:*elemX*-**1**, **0**:*ly*-**1**]

**вещтаб** *resOYZ*[**0**:*elemX*-**1**, **0**:*elemY*-**1**]

**| Построим одномерные кривые в плоскостях OXZ**

**нц** **для** *iy* **от** **0** **до** *ly* - **1**

**| Зададим значения производных Z в текущей плоскости в крайних точках**

*derZ*[**0**, *iy*] **:=** **0**

*derZ*[*lx* - **1**, *iy*] **:=** **0**

**| Найдём значения производных Z в текущей плоскости в остальных точках**

**нц** **для** *i* **от** **1** **до** (*lx* - **2**)

**если** (((*valZ*[*i*-**1**, *iy*] <= *valZ*[*i*, *iy*]) **и** (*valZ*[*i*+**1**, *iy*] <= *valZ*[*i*, *iy*])) **или** ((*valZ*[*i*-**1**, *iy*] >= *valZ*[*i*, *iy*]) **и** (*valZ*[*i*+**1**, *iy*] >= *valZ*[*i*, *iy*])))

**то** *derZ*[*i*, *iy*] **:=** **0**

**иначе** *derZ*[*i*, *iy*] **:=** **0.5** \* ((*valZ*[*i*, *iy*] - *valZ*[*i*-**1**, *iy*])/(*valX*[*i*] - *valX*[*i*-**1**]) + (*valZ*[*i*+**1**, *iy*] - *valZ*[*i*, *iy*])/(*valX*[*i*+**1**] - *valX*[*i*]))

**все**

**кц**

**| Пройдём по узлам интерполяции**

**нц** **для** *i* **от** **0** **до** (*lx* - **2**)

*Tn* **:=** *valX*[*i*+**1**] - *valX*[*i*] **| интервал между узлами**

*l* **:=** *valZ*[*i*+**1**, *iy*] - *valZ*[*i*, *iy*] - **0.5**\**Tn*\*(*derZ*[*i*+**1**, *iy*]+*derZ*[*i*, *iy*])

*p* **:=** (-**1**\**l* - ***sign***(*l*)\****sqrt***(*l*\**l* + **0.25**\*(*Tn*\*\***2**)\*((*derZ*[*i*, *iy*] - *derZ*[*i*+**1**, *iy*])\*\***2**)))/(**0.5**\**Tn*\**Tn*)

**если** (*p* = **0**)

**то** *d* **:=** **0**

**иначе** *d* **:=** (*derZ*[*i*+**1**, *iy*] - *derZ*[*i*, *iy*] + *Tn*\**p*)/(**2**\**p*)

**все**

*dx* **:=** (*valX*[*i*+**1**] - *valX*[*i*])/*k* **| Шаг интерполяции**

*dn* **:=** ***sign***(*p*)

*cn* **:=** ***abs***(*p*)\**dx*\**dx*

**| Построим сплайн на участке**

*na* **:=** ***int***(*d*/*dx*)

*nb* **:=** ***int***((*Tn* - *d*)/*dx*)

*resOXZ*[*i*\**k* + *k*, *iy*] **:=** *valZ*[*i*+**1**, *iy*] **| Yna = Y(n+1)**

*dZna* **:=** *derZ*[*i*+**1**, *iy*]

*current* **:=** *valX*[*i*+**1**]

**нц** **для** *j* **от** **0** **до** *na*-**1**

*resX*[*i*\**k* + *k* - *j*] **:=** *current*

*dZnai1nt* **:=** *dZna*\**dx* - *dn*\**cn*

*dZna* **:=** *dZnai1nt*/*dx*

*resOXZ*[*i*\**k*+*k*-*j*-**1**, *iy*] **:=** *resOXZ*[*i*\**k*+*k*-*j*, *iy*] - *dZnai1nt* - **0.5**\**dn*\**cn*

*current* **:=** *current* - *dx*

**кц**

*dn* **:=** -***sign***(*p*)

*resOXZ*[*i*\**k*, *iy*] **:=** *valZ*[*i*, *iy*] **| Y0 = Yn**

*dZ* **:=** *derZ*[*i*, *iy*]

*current* **:=** *valX*[*i*]

**нц** **для** *j* **от** **0** **до** *nb*-**1**

*resX*[*i*\**k* + *j*] **:=** *current*

*dZ1nt* **:=** *dZ*\**dx* + *dn*\**cn*

*dZ* **:=** *dZ1nt*/*dx*

*resOXZ*[*i*\**k* + *j* + **1**, *iy*] **:=** *resOXZ*[*i*\**k* + *j*, *iy*] + *dZ1nt* - **0.5**\**dn*\**cn*

*current* **:=** *current* + *dx*

**кц**

**кц**

**кц**

**| Построим одномерные кривые в плоскостях OYZ**

**| Данные для построения берутся из OXZ - результатов вычислений на предыдущем шаге**

**нц** **для** *ix* **от** **0** **до** *elemX* - **1**

**| Зададим значения производных Z в текущей плоскости в крайних точках**

*derZ*[*ix*, **0**] **:=** **0**

*derZ*[*ix*, *ly* - **1**] **:=** **0**

**| Найдём значения производных Z в текущей плоскости в остальных точках**

**нц** **для** *i* **от** **1** **до** (*ly* - **2**)

**если** (((*resOXZ*[*ix*, *i*-**1**] <= *resOXZ*[*ix*, *i*]) **и** (*resOXZ*[*ix*, *i*+**1**] <= *resOXZ*[*ix*, *i*])) **или** ((*resOXZ*[*ix*, *i*-**1**] >= *resOXZ*[*ix*, *i*]) **и** (*resOXZ*[*ix*, *i*+**1**] >= *resOXZ*[*ix*, *i*])))

**то** *derZ*[*ix*, *i*] **:=** **0**

**иначе** *derZ*[*ix*, *i*] **:=** **0.5** \* ((*resOXZ*[*ix*, *i*] - *resOXZ*[*ix*, *i*-**1**])/(*valY*[*i*] - *valY*[*i*-**1**]) + (*resOXZ*[*ix*, *i*+**1**] - *resOXZ*[*ix*, *i*])/(*valY*[*i*+**1**] - *valY*[*i*]))

**все**

**кц**

**| Пройдём по узлам интерполяции**

**нц** **для** *i* **от** **0** **до** (*ly* - **2**)

*Tn* **:=** *valY*[*i*+**1**] - *valY*[*i*] **| интервал между узлами**

*l* **:=** *resOXZ*[*ix*, *i*+**1**] - *resOXZ*[*ix*, *i*] - **0.5**\**Tn*\*(*derZ*[*ix*, *i*+**1**]+*derZ*[*ix*, *i*])

*p* **:=** (-**1**\**l* - ***sign***(*l*)\****sqrt***(*l*\**l* + **0.25**\*(*Tn*\*\***2**)\*((*derZ*[*ix*, *i*+**1**] - *derZ*[*ix*, *i*])\*\***2**)))/(**0.5**\**Tn*\**Tn*)

**если** (*p* = **0**)

**то** *d* **:=** **0**

**иначе** *d* **:=** (*derZ*[*ix*, *i*+**1**] - *derZ*[*ix*, *i*] + *Tn*\**p*)/(**2**\**p*)

**все**

*current* **:=** *valY*[*i*]

*dy* **:=** (*valY*[*i*+**1**] - *valY*[*i*])/*k* **| Шаг интерполяции**

*dn* **:=** ***sign***(*p*)

*cn* **:=** ***abs***(*p*)\**dy*\**dy*

**| Построим сплайн на участке**

*na* **:=** ***int***(*d*/*dy*)

*nb* **:=** ***int***((*Tn* - *d*)/*dy*)

*resOYZ*[*ix*, *i*\**k* + *k*] **:=** *resOXZ*[*ix*, *i*+**1**] **| Yna = Y(n+1)**

*dZna* **:=** *derZ*[*ix*, *i*+**1**]

*current* **:=** *valY*[*i*+**1**]

**нц** **для** *j* **от** **0** **до** *na*-**1**

*resY*[*i*\**k* + *k* - *j*] **:=** *current*

*dZnai1nt* **:=** *dZna*\**dy* - *dn*\**cn*

*dZna* **:=** *dZnai1nt*/*dy*

*resOYZ*[*ix*, *i*\**k*+*k*-*j*-**1**] **:=** *resOYZ*[*ix*, *i*\**k*+*k*-*j*] - *dZnai1nt* - **0.5**\**dn*\**cn*

*current* **:=** *current* - *dy*

**кц**

*dn* **:=** -***sign***(*p*)

*resOYZ*[*ix*, *i*\**k*] **:=** *resOXZ*[*ix*, *i*] **| Y0 = Yn**

*dZ* **:=** *derZ*[*ix*, *i*]

*current* **:=** *valY*[*i*]

**нц** **для** *j* **от** **0** **до** *nb*-**1**

*resY*[*i*\**k* + *j*] **:=** *current*

*dZ1nt* **:=** *dZ*\**dy* + *dn*\**cn*

*dZ* **:=** *dZ1nt*/*dy*

*resOYZ*[*ix*, *i*\**k* + *j* + **1**] **:=** *resOYZ*[*ix*, *i*\**k* + *j*] + *dZ1nt* - **0.5**\**dn*\**cn*

*current* **:=** *current* + *dy*

**кц**

**кц**

**кц**

**| Массив resX содержит координаты Х плоскостей OXZ**

**| Массив resY содержит координаты Y плоскостей OYZ**

**| Массив resOYZ содержит значения узлов интерполированной поверхности**

*outf* **:=** **открыть на запись**(**"output5.txt"**)

**|нц для j от 0 до elemX-1**

**|вывод outf, j, ' ', resOYZ[j, 0], нс;**

**|кц**

**нц** **для** *i* **от** **0** **до** *elemY*-**1**

**нц** **для** *j* **от** **0** **до** *elemX*-**1**

**вывод** *outf*, *resOYZ*[*j*, *i*], **' '**;

**кц**

**вывод** *outf*, **нс**

**кц**

**закрыть** (*outf*)

**кон**

**Вариант №6.** Создать программу, позволяющую строить интерполирующую *поверхность* на основе дельта-сплайнов с постоянным шагом дискретизации  между узлами интерполяции без алгоритмического использования операций многоразрядного умножения. Псевдокод.

**использовать** **Файлы**

**файл** *outf*

**цел** *i*, *j*, *ix*, *iy*, *k*, *na*, *nb*

**вещ** *current*, *d*, *dx*, *dy*, *l*, *p*, *Tn*

**вещ** *csn*, *nSna*, *nSnai1*, *nS*, *nSi*

**цел** *lx*, *ly*, *elemX*, *elemY*;

**алг**

**нач**

**вывод** **"Введите число интервалов: "**

**ввод** *k*

**вывод** **"Введите число шагов по X: "**

**ввод** *lx*

**вещтаб** *valX*[**0**:*lx*-**1**]

**нц** **для** *i* **от** **0** **до** *lx*-**1**

**вывод** **"Введите значение X"**, *i*, **": "**

**ввод** *valX*[*i*]

**кц**

**вывод** **"Введите число шагов по Y: "**

**ввод** *ly*

**вещтаб** *valY*[**0**:*ly*-**1**]

**нц** **для** *i* **от** **0** **до** *ly*-**1**

**вывод** **"Введите значение Y"**, *i*, **": "**

**ввод** *valY*[*i*]

**кц**

**вещтаб** *valZ*[**0**:*lx*-**1**, **0**:*ly*-**1**]

**нц** **для** *i* **от** **0** **до** *lx*-**1**

**нц** **для** *j* **от** **0** **до** *ly*-**1**

**вывод** **"Введите значение точки ("**, *i*, **","**, *j*, **"): "**

**ввод** *valZ*[*i*,*j*]

**кц**

**кц**

*elemX* **:=** (*lx* - **1**)\**k* + **1**;

*elemY* **:=** (*ly* - **1**)\**k* + **1**;

**вещтаб** *derZ*[**0**:*elemX*-**1**, **0**:*elemY*-**1**]

**вещтаб** *resX*[**0**:*elemX*-**1**]

**вещтаб** *resY*[**0**:*elemY*-**1**]

**вещтаб** *resOXZ*[**0**:*elemX*-**1**, **0**:*ly*-**1**]

**вещтаб** *resOYZ*[**0**:*elemX*-**1**, **0**:*elemY*-**1**]

**| Построим одномерные кривые в плоскостях OXZ**

**нц** **для** *iy* **от** **0** **до** *ly* - **1**

**| Зададим значения производных Z в текущей плоскости в крайних точках**

*derZ*[**0**, *iy*] **:=** **0**

*derZ*[*lx* - **1**, *iy*] **:=** **0**

**| Найдём значения производных Z в текущей плоскости в остальных точках**

**нц** **для** *i* **от** **1** **до** (*lx* - **2**)

**если** (((*valZ*[*i*-**1**, *iy*] <= *valZ*[*i*, *iy*]) **и** (*valZ*[*i*+**1**, *iy*] <= *valZ*[*i*, *iy*])) **или** ((*valZ*[*i*-**1**, *iy*] >= *valZ*[*i*, *iy*]) **и** (*valZ*[*i*+**1**, *iy*] >= *valZ*[*i*, *iy*])))

**то** *derZ*[*i*, *iy*] **:=** **0**

**иначе** *derZ*[*i*, *iy*] **:=** **0.5** \* ((*valZ*[*i*, *iy*] - *valZ*[*i*-**1**, *iy*])/(*valX*[*i*] - *valX*[*i*-**1**]) + (*valZ*[*i*+**1**, *iy*] - *valZ*[*i*, *iy*])/(*valX*[*i*+**1**] - *valX*[*i*]))

**все**

**кц**

**| Пройдём по узлам интерполяции**

**нц** **для** *i* **от** **0** **до** (*lx* - **2**)

*Tn* **:=** *valX*[*i*+**1**] - *valX*[*i*] **| интервал между узлами**

*l* **:=** *valZ*[*i*+**1**, *iy*] - *valZ*[*i*, *iy*] - **0.5**\**Tn*\*(*derZ*[*i*+**1**, *iy*]+*derZ*[*i*, *iy*])

*p* **:=** (-**1**\**l* - ***sign***(*l*)\****sqrt***(*l*\**l* + **0.25**\*(*Tn*\*\***2**)\*((*derZ*[*i*, *iy*] - *derZ*[*i*+**1**, *iy*])\*\***2**)))/(**0.5**\**Tn*\**Tn*)

**если** (*p* = **0**)

**то** *d* **:=** **0**

**иначе** *d* **:=** (*derZ*[*i*+**1**, *iy*] - *derZ*[*i*, *iy*] + *Tn*\**p*)/(**2**\**p*)

**все**

*dx* **:=** (*valX*[*i*+**1**] - *valX*[*i*])/*k* **| Шаг интерполяции**

**| Построим сплайн на участке**

*na* **:=** ***int***(*d*/*dx*)

*nb* **:=** ***int***((*Tn* - *d*)/*dx*)

*csn* **:=** *p*\**dx*\**dx*

*nSna* **:=** *derZ*[*i*+**1**, *iy*]\**dx*

*nS* **:=** *derZ*[*i*, *iy*]\**dx*

*resOXZ*[*i*\**k* + *k*, *iy*] **:=** *valZ*[*i*+**1**, *iy*] **| Yna = Y(n+1)**

*current* **:=** *valX*[*i*+**1**]

**нц** **для** *j* **от** **0** **до** *na*-**1**

*resX*[*i*\**k* + *k* - *j*] **:=** *current*

*nSnai1* **:=** *nSna* - *csn*

*nSna* **:=** *nSnai1*

*resOXZ*[*i*\**k*+*k*-*j*-**1**, *iy*] **:=** *resOXZ*[*i*\**k*+*k*-*j*, *iy*] - *nSnai1* - **0.5**\**csn*

*current* **:=** *current* - *dx*

**кц**

*resOXZ*[*i*\**k*, *iy*] **:=** *valZ*[*i*, *iy*] **| Y0 = Yn**

*current* **:=** *valX*[*i*]

**нц** **для** *j* **от** **0** **до** *nb*-**1**

*resX*[*i*\**k* + *j*] **:=** *current*

*nSi* **:=** *nS* - *csn*

*nS* **:=** *nSi*

*resOXZ*[*i*\**k* + *j* + **1**, *iy*] **:=** *resOXZ*[*i*\**k* + *j*, *iy*] + *nSi* + **0.5**\**csn*

*current* **:=** *current* + *dx*

**кц**

**кц**

**кц**

**| Построим одномерные кривые в плоскостях OYZ**

**| Данные для построения берутся из OXZ - результатов вычислений на предыдущем шаге**

**нц** **для** *ix* **от** **0** **до** *elemX* - **1**

**| Зададим значения производных Z в текущей плоскости в крайних точках**

*derZ*[*ix*, **0**] **:=** **0**

*derZ*[*ix*, *ly* - **1**] **:=** **0**

**| Найдём значения производных Z в текущей плоскости в остальных точках**

**нц** **для** *i* **от** **1** **до** (*ly* - **2**)

**если** (((*resOXZ*[*ix*, *i*-**1**] <= *resOXZ*[*ix*, *i*]) **и** (*resOXZ*[*ix*, *i*+**1**] <= *resOXZ*[*ix*, *i*])) **или** ((*resOXZ*[*ix*, *i*-**1**] >= *resOXZ*[*ix*, *i*]) **и** (*resOXZ*[*ix*, *i*+**1**] >= *resOXZ*[*ix*, *i*])))

**то** *derZ*[*ix*, *i*] **:=** **0**

**иначе** *derZ*[*ix*, *i*] **:=** **0.5** \* ((*resOXZ*[*ix*, *i*] - *resOXZ*[*ix*, *i*-**1**])/(*valY*[*i*] - *valY*[*i*-**1**]) + (*resOXZ*[*ix*, *i*+**1**] - *resOXZ*[*ix*, *i*])/(*valY*[*i*+**1**] - *valY*[*i*]))

**все**

**кц**

**| Пройдём по узлам интерполяции**

**нц** **для** *i* **от** **0** **до** (*ly* - **2**)

*Tn* **:=** *valY*[*i*+**1**] - *valY*[*i*] **| интервал между узлами**

*l* **:=** *resOXZ*[*ix*, *i*+**1**] - *resOXZ*[*ix*, *i*] - **0.5**\**Tn*\*(*derZ*[*ix*, *i*+**1**]+*derZ*[*ix*, *i*])

*p* **:=** (-**1**\**l* - ***sign***(*l*)\****sqrt***(*l*\**l* + **0.25**\*(*Tn*\*\***2**)\*((*derZ*[*ix*, *i*+**1**] - *derZ*[*ix*, *i*])\*\***2**)))/(**0.5**\**Tn*\**Tn*)

**если** (*p* = **0**)

**то** *d* **:=** **0**

**иначе** *d* **:=** (*derZ*[*ix*, *i*+**1**] - *derZ*[*ix*, *i*] + *Tn*\**p*)/(**2**\**p*)

**все**

*current* **:=** *valY*[*i*]

*dy* **:=** (*valY*[*i*+**1**] - *valY*[*i*])/*k* **| Шаг интерполяции**

**| Построим сплайн на участке**

*na* **:=** ***int***(*d*/*dy*)

*nb* **:=** ***int***((*Tn* - *d*)/*dy*)

*csn* **:=** *p*\**dx*\**dx*

*nSna* **:=** *derZ*[*ix*, *i*+**1**]\**dy*

*nS* **:=** *derZ*[*ix*, *i*]\**dy*

*resOYZ*[*ix*, *i*\**k* + *k*] **:=** *resOXZ*[*ix*, *i*+**1**] **| Yna = Y(n+1)**

*current* **:=** *valY*[*i*+**1**]

**нц** **для** *j* **от** **0** **до** *na*-**1**

*resY*[*i*\**k* + *k* - *j*] **:=** *current*

*nSnai1* **:=** *nSna* - *csn*

*nSna* **:=** *nSnai1*

*resOYZ*[*ix*, *i*\**k*+*k*-*j*-**1**] **:=** *resOYZ*[*ix*, *i*\**k*+*k*-*j*] - *nSnai1* - **0.5**\**csn*

*current* **:=** *current* - *dy*

**кц**

*resOYZ*[*ix*, *i*\**k*] **:=** *resOXZ*[*ix*, *i*] **| Y0 = Yn**

*dy* **:=** *derZ*[*ix*, *i*]

*current* **:=** *valY*[*i*]

**нц** **для** *j* **от** **0** **до** *nb*-**1**

*resY*[*i*\**k* + *j*] **:=** *current*

*nSi* **:=** *nS* - *csn*

*nS* **:=** *nSi*

*resOYZ*[*ix*, *i*\**k* + *j* + **1**] **:=** *resOYZ*[*ix*, *i*\**k* + *j*] + *nSi* + **0.5**\**csn*

*current* **:=** *current* + *dx*

**кц**

**кц**

**кц**

**| Массив resX содержит координаты Х плоскостей OXZ**

**| Массив resY содержит координаты Y плоскостей OYZ**

**| Массив resOYZ содержит значения узлов интерполированной поверхности**

*outf* **:=** **открыть на запись**(**"output6.txt"**)

**|нц для j от 0 до elemX-1**

**|вывод outf, j, ' ', resOYZ[j, 0], нс;**

**|кц**

**нц** **для** *i* **от** **0** **до** *elemY*-**1**

**нц** **для** *j* **от** **0** **до** *elemX*-**1**

**вывод** *outf*, *resOYZ*[*j*, *i*], **' '**;

**кц**

**вывод** *outf*, **нс**

**кц**

**закрыть** (*outf*)

**кон**

**Кравченко Павел Павлович**

**Жиглатый Артемий Александрович**

Учебно-методическое пособие

“Построение сплайновых кривых и поверхностей на основе дельта-преобразований второго порядка”

по дисциплине

СПЕЦИАЛИЗИРОВАННЫЕ МЕТОДЫ И АЛГОРИТМЫ ОБРАБОТКИ ЦИФРОВОЙ ИНФОРМАЦИИ

Ответственный за выпуск Кравченко П.П.

Редактор

Корректор

Подписано в печать 2016 г.

Заказ № . Тираж 20 экз.

Формат 60х84 1/16 . Печ.л. 5.2. Уч.-изд. л. 5

Издательство Южного федерального университета

344091, г.Ростов-на-Дону, пр. Стачки,200/1

Тел. (863)2478051.

Отпечатано в Секторе обеспечения полиграфической

Продукцией кампуса в г. Таганроге отдела полиграфической,

корпоративной и сувенирной продукции

ИПК КИБИ МЕДИА ЦЕНТРА ЮФУ

ГСП 17А, Таганрог, 28, Энгельса, 1.

Тел. (8634)371717.