

# Об алгоритме построения конформной квази-иерархической треугольной сетки, слабо $\delta$ -аппроксимирующей заданные ломаные<sup>§</sup>

В. Н. Чугунов<sup>©</sup>

*Предложен алгоритм построения конформной квази-иерархической треугольной сетки, аппроксимирующей с точностью  $\delta$  набор заданных ломаных. Возможность сдвига ломаных в пределах их  $\delta$ -окрестности гарантирует разрешимость задачи. Результирующая сетка имеет небольшое число треугольников и допускает реализацию многосеточного метода. Установлена конечность данного алгоритма и доказана оценка на рост числа треугольников в результирующей сетке при уменьшении параметра  $\delta$  (порядка  $\log_2^2 \delta^{-1}$ ). Приведены результаты работы алгоритма для конкретного заданного набора ломаных.*

## 1. Введение

Задачи построения конформной треугольной сетки, удовлетворяющей некоторым ограничениям, часто встречаются на практике. Набор ограничений диктуется постановкой задачи. Типичными ограничениями являются отсутствие малых углов у треугольников сетки, аппроксимация границы области ребрами сетки со вторым порядком относительно длин ребер.

---

<sup>§</sup>Работа выполнена при частичной финансовой поддержке Российского фонда фундаментальных исследований (гранты РФФИ №№05-01-00721 и 06-01-08052) и программы фундаментальных исследований отделения математических наук РАН «Вычислительные и информационные проблемы решения больших задач» по проекту «Матричные методы и технологии для задач со сверхбольшим числом неизвестных», а также исследовательского гранта ExxonMobil Corp.

<sup>©</sup>Институт вычислительной математики РАН (119333, Москва, ул. Губкина, д.8) e-mail: vadim@bach.inm.ras.ru

Известны задачи, в которых возникают дополнительные ограничения, связанные с аппроксимацией *множественных* внутренних границ. Примером может служить двумерная подзадача, возникающая при моделировании трехмерных течений в пористых средах, обладающих слоистой структурой. Для решения последней задачи удобно использовать призматические сетки с основаниями призм, лежащими на границах геологических слоев. Сами геологические слои, связанные с различными породами, имеют переменную толщину и могут сужаться вплоть до полного исчезновения на линиях вырождения, а также менять толщину скачкообразно в районе вертикальных разломов земной коры. Для успешного решения задач моделирования необходимо, чтобы призматическая сетка отражала эти особенности, а именно, линии вырождения и поверхности вертикальных разломов были аппроксимированы ребрами и гранями призм. Одним из способов построения желаемой сетки является проектирование разломов и линий вырождения на двумерную область и задание их ломаными. Имея двумерную сетку со сторонами треугольников, лежащими на ломаных, можно путем дублирования двумерной сетки для каждого геологического слоя получить трехмерную сетку, учитывающую имеющиеся особенности структуры пористой среды. Построение сетки с внутренними множественными ограничениями может быть также использовано как при решении других сложных задач, так и само по себе.

Набор условий, накладываемых на алгоритм, определяется следующими математическими требованиями, предъявляемыми к сетке. Во-первых, хотелось иметь возможность использовать получаемую сетку при реализации многосеточных методов [2], являющихся наиболее эффективными методами решения краевых задач. Это приводит к условию, чтобы полученная сетка была заключительным звеном некоторой последовательности логически вложенных друг в друга сеток, для которой задано правило перехода от одной сетки к другой. Во-вторых, требования устойчивости аппроксимации краевых задач ограничивает класс двумерных сеток триангуляциями с треугольниками регулярной формы [7, 9]. Наряду с удовлетворением этим математическим условиям, цель алгоритма за-

ключается в том, чтобы результирующая сетка подчинялась также набору физических требований. А именно, построенная двумерная сетка должна обладать небольшим числом треугольников в силу ограничений, накладываемых возможностями вычислительной техники при решении задач, использующих треугольную сетку. Кроме этого, так как данные о внутренних границах не могут быть точными из-за погрешностей измерений, всегда существует некоторый порог для точности аппроксимации заданных ломаных, что допускает их незначительную локальную модификацию. Вышеприведенные соображения определяют рассматриваемый класс сеток и ограничений.

Существует несколько методов генерации треугольных сеток [7, 9, 10], удовлетворяющей множественным ограничениям, а также целый ряд их программных реализаций [15, 16]. Первый алгоритм принадлежит Бейкеру, Гроссе и Рафферти [3]. Они предложили метод триангуляции области, являющейся множеством многоугольников с углами не менее  $13^\circ$ . В итоговой сетке углы всех треугольников не превосходят  $90$  градусов и наименьший угол не меньше  $13^\circ$ . Однако, получающаяся сетка является равномерной и, поэтому, имеет очень много сеточных элементов.

Проблема большого числа треугольников при решении рассматриваемой задачи позднее была устранена Верном, Эпштейном и Гильбертом [5]. Они создали сеточный генератор с треугольниками, имеющими ограничения на форму и размер, подчиняющиеся начальным данным. Кроме того, авторы показали, что строящаяся ими сетка имеет треугольники с отношением сторон, не превосходящим числа  $5$ . Их алгоритм основан на идее предварительного рекурсивного разбиения области на квадраты различной формы.

Другим способом решения поставленной задачи является триангуляция Делонэ. Обзор адекватных методов для построения триангуляций Делонэ с множественными ограничениями можно найти в [1, 13]. Наиболее известные результаты представлены в работах Чью и Рапперта. Чью [8] представил триангуляцию Делонэ, в которой все треугольники имеют углы между  $30$  и  $120$  градусов, для области с наименьшим углом, образованным границами, не меньше  $30$  граду-

сов. Его алгоритм производит равномерную сетку.

Рапперт [12] расширил идеи Чью, предложив алгоритм триангуляции планарного прямолинейного графа такой, что все треугольники выходной сетки имеют углы между  $\alpha$  и  $\pi - 2\alpha$  ( $0 < \alpha < 20$ ), где  $\alpha$  зависит от входных ограничений. Сетка является неравномерной. Алгоритм Рапперта, получая на вход некоторую триангуляцию Делонэ, точно аппроксимирующую ребра графа, выдает сетку с треугольниками, удовлетворяющими ограничениям на форму. При этом начальная аппроксимация сохраняется. Таким образом, данный алгоритм лишь улучшает имеющуюся триангуляцию. В статье [12] доказывается, что число треугольников в выходной сетке получается из начального числа, умножением на константу, зависящую от  $\alpha$ .

В представляемой работе исследуется иной подход к решению задачи. В отличие от описываемых выше методов в статье представлен генератор, который получает на вход треугольную конформную сетку, точно аппроксимирующую только внешнюю границу и не учитывающую внутренние ограничения. Результирующая конформная неравномерная сетка с треугольниками, удовлетворяющими ограничениям на форму, сохраняя точную аппроксимацию внешней границы, приближенно аппроксимирует внутренние границы с заданной точностью  $\delta$ . При этом сама аппроксимация внутренних границ во время построения сетки может сдвигаться от границы на величину не более  $\delta$  с целью безотказной работы алгоритма. Представляемый генератор обладает рядом преимуществ. Во-первых, процесс построения сетки является конструктивной процедурой. Итоговая сетка получается из начальной последовательным применением одной из двух операций: сдвига узла и разбиения треугольника на два методом бисекции. Это обстоятельство дает возможность ввести иерархию сеток для использования многосеточных методов. Во-вторых, приближенная аппроксимация внутренних границ и возможность ее сдвига позволяют применять алгоритм для разнообразных конфигураций заданных внутренних ограничений. В-третьих, число треугольников в полученной сетке будет небольшим благодаря приближенной аппроксимации внутренних границ. Мы даем на него оценку,

зависящую не от начальной сетки, а от точности аппроксимации  $\delta$ .

Постановки задач, рассматриваемых в этой статье, обсуждаются в разделе 2. Алгоритм решения задачи слабой  $\delta$ -аппроксимации ломаных, будет описан в разделе 3. Характеристика свойств сетки, которую строит представляемый алгоритм, содержится в пункте 4. В разделе 5 приведены примеры работы представляемого алгоритма.

Автор выражает благодарность Василевскому Ю. В. за постановку задачи, консультации при ее решении и написании статьи.

## 2. Постановка задачи

В более общем виде задача аппроксимации ломаных может быть сформулирована следующим образом. Дана грубая начальная конформная треугольная сетка, покрывающая заданную полигональную область  $\Omega$ , точно аппроксимирующая границу области  $\Omega$ , и некоторое множество ломаных, задаваемых упорядоченным набором точек на плоскости. Заданная сетка будет порождать множество логически иерархических конформных сеток, получаемых последовательным многоуровневым разбиением некоторых треугольников на два подтреугольника методом бисекции [4, 11]. Конформные сетки, получаемые сдвигом узлов иерархических сеток, называются квази-иерархическими сетками. Множество квази-иерархических сеток позволяет организовать многосеточные алгоритмы на основе исключительно геометрической (топологической) информации, поэтому именно эти сетки рассматриваются ниже.

Для каждого треугольника определены две численные характеристики — качество формы и качество аппроксимации (множества ломаных). Под качеством формы  $Q_i$  треугольника  $i$  понимается величина, определяемая по формуле [6, 14]:

$$Q_i = \frac{P_0^2}{S_0} * \frac{S_i}{P_i^2} = 12\sqrt{3} * \frac{S_i}{P_i^2}, \quad (1)$$

где  $S_i$  — площадь,  $P_i$  — периметр треугольника  $i$ ,  $P_0$ ,  $S_0$  — периметр и площадь любого равностороннего треугольника. Отметим,

что  $Q_i \leq 1$ , причем  $Q_i = 1$  только для равностороннего треугольника. Для вычисления качества аппроксимации  $\hat{Q}_i$  треугольника  $i$  сначала определяется качество аппроксимации  $\hat{Q}_i^{(l,j)}$  по отношению к каждому звену  $j$  каждой ломаной  $l$ : если звено  $j$  ломаной  $l$  пересекает треугольник  $i$ , то

$$\hat{Q}_i^{(l,j)} = 1 - \frac{2 * \min(S_1, S_2)}{S}, \quad (2)$$

где  $S$  — площадь треугольника, а  $S_1$  и  $S_2$  — площади фигур, на которые звено  $j$  ломаной  $l$  разбивает треугольник  $i$ ; в противном случае (пересечения нет) положим

$$\hat{Q}_i^{(l,j)} = 1. \quad (3)$$

Таким образом,  $\hat{Q}_i^{(l,j)} = 1$  для треугольника  $i$ , не разбиваемого звеном  $j$  ломаной  $l$ . Как только качество аппроксимации  $\hat{Q}_i^{(l,j)}$  определено для каждого звена  $j$  каждой ломаной  $l$ , качество аппроксимации  $\hat{Q}_i$  треугольника  $i$  берется как минимум среди найденных  $\hat{Q}_i^{(l,j)}$

$$\hat{Q}_i = \min_l \min_j \hat{Q}_i^{(l,j)}. \quad (4)$$

Для всей сетки  $\epsilon$  также введено качество формы сетки  $Q_\epsilon$  и качество аппроксимации сетки  $\hat{Q}_\epsilon$  как минимум из соответствующих качеств треугольников

$$\begin{aligned} Q_\epsilon &= \min_i Q_i, \\ \hat{Q}_\epsilon &= \min_i \hat{Q}_i. \end{aligned} \quad (5)$$

В *оптимизационной задаче точной аппроксимации ломаных* квази-иерархическими сетками требуется для заданного числа  $0 < q_0 < 1$  построить конформную квази-иерархическую треугольную сетку  $\epsilon_0$  на  $\Omega$ , удовлетворяющую условию

$$\epsilon_0 = \arg \min_{\epsilon: Q_\epsilon \geq q_0, \hat{Q}_\epsilon = 1} nt(\epsilon), \quad (6)$$

где  $nt(\epsilon)$  — число треугольников в сетке  $\epsilon$ . Другими словами, цель задачи — для заданного числа  $q_0$  построить конформную треугольную сетку, удовлетворяющую условиям: сетка квази-иерархична; заданные ломаные аппроксимируются сторонами треугольников, т.е.

ломаные лежат на ребрах треугольников; треугольники имеют хорошее качество формы; число треугольников минимально.

Однако, оптимизационная задача точной аппроксимации ломаных достаточно сложна для решения. Заменяем ее менее трудной: будем считать, что заданные ломаные могут аппроксимироваться сторонами треугольников с некоторой заранее задаваемой точностью  $\delta$ . Пусть  $d(i)$  — диаметр треугольника  $i$ . Определим для каждого треугольника  $\delta$ -качество аппроксимации  $\hat{Q}_i^\delta$

$$\hat{Q}_i^\delta = \begin{cases} \hat{Q}_i, & \text{если } d(i) > \delta, \\ 1, & \text{в противном случае,} \end{cases} \quad (7)$$

а для сетки  $\delta$ -качество аппроксимации  $\hat{Q}_\epsilon^\delta$

$$\hat{Q}_\epsilon^\delta = \min_i \hat{Q}_i^\delta. \quad (8)$$

То есть, ломаная может пересекать треугольник  $i$  по линии, не совпадающей с ребром, если  $d(i) \leq \delta$ .

Оптимизационная задача  $\delta$ -аппроксимации ломаных состоит в том, чтобы для заданных чисел  $0 < q_0 < 1$ ,  $0 < \delta \ll 1$  построить конформную квази-иерархическую треугольную сетку  $\epsilon_0$  на  $\Omega$  такую, что

$$\epsilon_0 = \arg \min_{\epsilon: Q_\epsilon \geq q_0, \hat{Q}_\epsilon^\delta = 1} nt(\epsilon). \quad (9)$$

Предлагаемый ниже алгоритм предназначен для решения еще более упрощенной задачи, называемой в дальнейшем *задачей слабой  $\delta$ -аппроксимации ломаных*. В этой задаче разрешается сдвигать ломаные на расстояние не больше  $\delta$ . Как будет показано ниже, это обеспечивает построение приближенного решения задачи в некотором классе сеток. Фактически, мы рассматриваем построение квази-иерархической сетки  $\epsilon_0$ , удовлетворяющей условиям  $Q_{\epsilon_0} \geq q_0$ ,  $\hat{Q}_{\epsilon_0}^\delta = 1$  и получающейся из заданной путем как можно меньшего числа действий с узлами сетки и треугольниками. Приближенность решения обусловлена заменой условия строгой минимизации числа треугольников на желание получить сетку лишь с небольшим,

по-возможности, числом сеточных элементов<sup>33</sup>.

Отметим, что замена оптимизационной задачи точной аппроксимации ломаных задачей слабой  $\delta$ -аппроксимации ломаных обусловлена наличием погрешности в представлении данных. Переход к упрощенной постановке, сохраняющей сущность двумерной подзадачи в рамках трехмерной задачи фильтрации, обеспечивает простоту и надежность предлагаемого алгоритма, в котором будут использоваться лишь две основные операции: сдвиг узла и разбиение треугольника на два подтреугольника [4, 11]. В качестве параметра  $\delta$  можно рассматривать точность представления начальных данных. Из описания алгоритма будет видно, что при  $\delta$ , стремящимся к нулю, заданные ломаные не сдвигаются, стороны треугольников почти точно их аппроксимируют, а число треугольников растет умеренно.

### 3. Описание алгоритма

**3.1. Обозначения и базовые операции.** Прежде, чем изложить алгоритм построения сетки, опишем предположения, при которых будет решаться задача слабой  $\delta$ -аппроксимации ломаных, и введем некоторые необходимые в дальнейшем обозначения.

Данную задачу будем решать при следующих предположениях: число ломаных и число звеньев в одной ломаной не больше некоторых заданных значений; любая ломаная не имеет самопересечений; каждая ломаная лежит в замыкании  $\Omega$ ; ни одна из внутренних точек любой ломаной не лежит на границе области  $\Omega$ <sup>34</sup>.

При формулировании алгоритма нам понадобятся различные типы узлов сетки. Для этого все узлы сетки мы разделим на неподвижные узлы, подвижные вдоль границы и подвижные в любом направлении. В дальнейшем нам также понадобится понятие качество формы узла, которое положим равным минимуму из качеств форм всех треугольников, содержащих данный узел и образующих

<sup>33</sup>Структура алгоритма позволяет дополнительно локально измельчать построенную сетку.

<sup>34</sup>Если какая-то ломаная имеет внутреннюю точку (или отрезок) на границе, разобьем ее на две, тогда точка на границе будет граничной для обеих ломаных (в случае отрезка удаляем этот отрезок из ломаной).



суперэлемент данного узла. Также определим понятие суперэлемент треугольника как объединение суперэлементов его вершин.

Введем число  $\varepsilon$  как минимум из расстояний между двумя звеньями ломаных, не имеющих общих точек, и расстояний между звеном ломаной и отрезком границы всей области, который данное звено не пересекает. Тогда длина любого звена любой ломаной не меньше  $\varepsilon$ .

При описании процесса построения сетки мы будем использовать две вспомогательные процедуры.

Одна из них — это процедура *бисекции*, которая применима к сетке, у которой помечены некоторые треугольники. Процедура бисекции заключается в разбиении на два каждого из помеченных треугольников, а также некоторых других для сохранения конформности сетки. Важным свойством *BS* алгоритма бисекции является локальность: бисекция одного помеченного треугольника может порождать бисекцию лишь треугольников из суперэлемента помеченного треугольника [1,2]. В дальнейшем нам потребуется еще одно свойство *KBS* алгоритма бисекции, которое заключается в том, что применение к данной сетке любого числа уровней бисекций не ухудшит качество формы сетки более, чем в два раза [11].

Также нам будет необходима операция *проверки возможности сдвига*  $\mathcal{S}(J, R, j_0, r_0, \hat{q})$  одного узла из заданного подмножества  $J$  точек сетки в одну из точек предписанного множества  $R$ . Если все узлы в  $J$  неподвижны, процедура выдает информацию о невозможности сдвига, иначе она сначала строит подмножество  $\tilde{J}$  из всех подвижных узлов в  $J$  и моделирует сдвиг каждого элемента  $j \in \tilde{J}$  в каждую из точек  $r \in R$  с целью найти такую пару элементов  $j_0 \in \tilde{J}$  и  $r_0 \in R$ , что при перемещении  $j_0$  в  $r_0$  треугольники не налегают друг на друга и не выворачиваются, а качество формы  $q(j_0)$  узла  $j_0$  будет равным

$$q(j_0)|_{j_0 \rightarrow r_0} = \max_{j \in \tilde{J}} \max_{r \in R} q(j)|_{j \rightarrow r},$$

для всех пар  $j, r$ , обеспечивающих конформность получаемой сетки. Если такая пара нашлась, и получаемое качество формы  $q(j_0)$  узла  $j_0$  после сдвига не ниже  $\hat{q}$ , то процедура выдает на выходе пару

$j_0, r_0$ , в противном случае выдается информация о невозможности сдвига. Заметим, что если вершина подвижна только вдоль границы, то для нее сдвиг должен моделироваться только для тех точек  $R$ , которые находятся на границе. Отметим важное свойство  $SD$  операции сдвига. Для любого треугольника, пересекаемого некой прямой, содержащей его вершину, сдвиг ближайшей из других вершин ( $j$ ) на прямую не ухудшит качество формы сдвигаемого узла  $q(j)$  более чем в  $\kappa(q)$  раз.

Теперь можно перейти к описанию алгоритма решения нашей задачи, который будет использовать процедуры бисекции и сдвигов узлов. Пусть начальная сетка имеет качество формы  $q$ . Будем предполагать, что порог  $q_0$  из (9) не превосходит  $q/4$ , что необходимо для обеспечения возможности совершать бисекции и сдвиги.

**3.2. Описание алгоритма.** Рассматриваемый алгоритм построения треугольной конформной квази-иерархической сетки  $\epsilon_0$  с небольшим числом треугольников, имеющей  $\delta$ -качество аппроксимации  $\hat{Q}_{\epsilon_0}^\delta = 1$  и качество формы  $Q_{\epsilon_0} \geq q_0$ , на предварительном этапе может сдвинуть сами ломаные на величину не более  $\delta$ . Это обеспечивает разрешимость задачи слабой  $\delta$ -аппроксимации ломаных для заданного  $q_0$ . Алгоритм включает в себя три этапа, дальнейшее описание каждого из которых, при необходимости, будет заканчиваться обоснованием возможности совершения действий, составляющих основу изложенного этапа.

**Этап 1** — аппроксимация общих точек соседних звеньев ломаных (т.н. точек излома), граничных точек ломаных и точек пересечения ломаных узлами сетки с помощью допустимых операций, и, если необходимо, сдвиг ломаных на величину не более  $\delta$ . Этот этап состоит из нескольких шагов, проиллюстрированных на рис. 1–4.

**Шаг 1-1.** Строим множество  $Z$ , состоящее из точек излома, граничных точек ломаных и точек пересечения заданных ломаных (рис. 1).

**Шаг 1-2.** Удаляем из  $Z$  точки, которые лежат в узлах сетки, объявив соответствующие узлы сетки неподвижными. Для каждого треугольника  $i$  находим подмножество точек  $Z_i \in Z$ , принадлежащих этому треугольнику. Если треугольник имеет непустое множе-

ство  $Z_i$ , формируем множество  $L_i$  из трех его вершин и выполняем процедуру проверки возможности сдвига  $S(L_i, Z_i, l_0, z_0, 4 * q_0)$ . Если она дает положительный результат, то совершаем сдвиг  $l_0$  в  $z_0$ , объявляем вершину  $l_0$  неподвижной и исключаем  $z_0$  из  $Z$ , иначе, в случае если  $d(i) > \delta$ , метим данный треугольник для разбиения (рис. 2). После просмотра всех треугольников при наличии помеченных выполняем процедуру бисекции, которая дает дополнительные степени свободы. Если множество  $Z$  пусто, то переходим к этапу 2, иначе повторяем шаг 1-2 до тех пор, пока не получим сетку, в которой любой треугольник  $i$  либо не содержит точек из  $Z$ , либо содержит только одну точку и  $d(i) \leq \delta$  (рис. 3).

**Шаг 1-3.** Для каждой точки из  $Z$  совершаем процедуру сдвига самой точки из  $Z$  на величину не более  $\delta$ , что приводит к сдвигу ломаных не более, чем на  $\delta$ . Для этого на треугольнике, ее содержащем, строим локальную сетку, включающую сами вершины треугольника, и из них формируем множество  $R_i$ . Определив множество  $L_i$  из трех вершин треугольника, выполняем процедуру проверки возможности сдвига  $S(L_i, R_i, l_0, r_0, 2 * q_0)$ , которая дает положительный результат в силу того, что сами вершины треугольника уже принадлежат локальной сетке и их качество формы не ниже  $2 * q_0$ , так как начальная сетка имела качество не хуже  $4 * q_0$ , а на шаге 1-2 подвергалась лишь бисекции и сдвигам  $S$  с качеством  $4 * q_0$ . Передвигаем точку  $Z$  в эту выбранную точку локальной сетки  $r_0$ . Если точка из  $Z$  — точка пересечения двух или более ломаных, то находим в этих ломаных звенья, содержащих эту точку и добавляем точку из  $Z$  в упорядоченные списки точек, задающих эти ломаные. Теперь в точку  $Z$  мы можем сдвинуть вершину треугольника  $l_0$  (рис. 4) и зафиксировать ее.

На этом первый этап алгоритма завершен. В результате получаем конформную сетку с узлами, содержащими все элементы множества  $Z$ , и качеством формы, не хуже  $2 * q_0$ .

Отметим, что шаг 1-2 может производить сетку, сгущающуюся к некоторому конечному числу точек, не большему чем мощность изначального множества  $Z$ .

Достичь условия, чтобы все точки  $Z$  лежали в узлах сетки при

заданном  $q_0$  возможно лишь благодаря шагу 1-3. Если  $q_0$  достаточно велико, то в результате работы только шагов 1-1 и 1-2 совместить все точки  $Z$  с узлами сетки не в любом случае возможно, так как большое  $q_0$  не всегда дает возможность сдвига узла, а значит, процесс из шагов 1-1 и 1-2 без ограничения на диаметр треугольников может заиклиться. Шаг 1-3 двигает сами точки  $Z$  так, чтобы для заданного  $q_0$  можно было осуществить сдвиг. В этих случаях точка из  $Z$  сама передвигается в узел сетки, обеспечивая требуемый результат.

Если важно гарантированно сохранить взаимное расположение ломаных, необходимо выбрать  $\delta$  меньше четверти  $\varepsilon$ . Действительно, перед шагом 1-3 точки, еще неаппроксимированные узлами сетки, принадлежат треугольникам с диаметром не более  $\delta$ . Поэтому сдвиг точек  $Z$  может их сблизить, но лишь до расстояния  $2\delta = \varepsilon/2$ , что обеспечит отсутствие возможного слияния разных звеньев различных ломаных.

Из описания шага 1 видно, что величина сдвига ломаных зависит от геометрии ломаных и от  $\delta$ ; кроме того, если  $\delta$  устремить к нулю, то, начиная с некоторого момента, сдвига ломаных вообще не происходит. Сдвиг ломаных происходит локально, так как после каждого сдвига точки множества  $Z$  в новую точку помещается узел сетки, и рассматриваемый элемент  $Z$  исключается из множества  $Z$ . Поэтому любая точка новой ломаной будет находиться на расстоянии не более  $\delta$  от своего прообраза на старой ломаной даже для суперпозиции нескольких сдвигов.

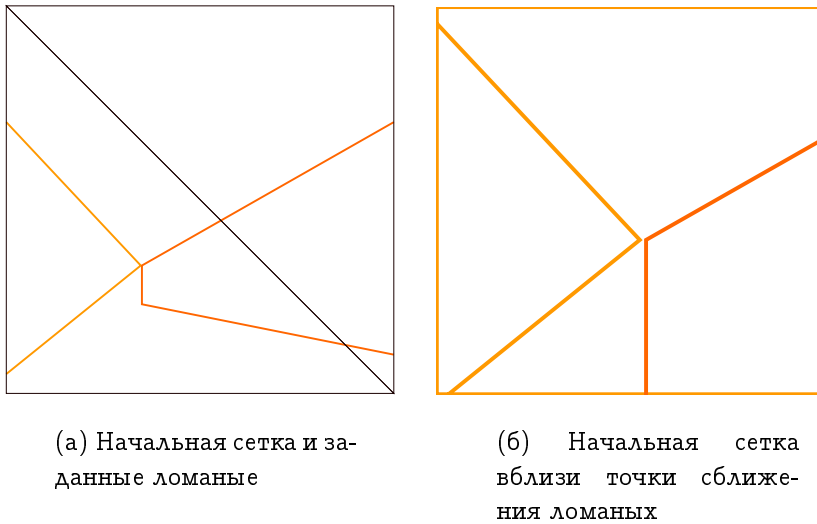
**Этап 2** — приближение сторон треугольников к звеньям ломаных с использованием лишь допустимых операций.

**Шаг 2-1.** Если все треугольники имеют  $\delta$ -качество аппроксимации равное единице, то переходим к этапу 3.

**Шаг 2-2.** Рассмотрим каждый треугольник  $i$  с  $\delta$ -качеством аппроксимации, отличным от единицы. Если его качество аппроксимации отлично от единицы для некоторого звена  $j$  некоторой ломаной  $l$ , и хотя бы одна из его вершин, не лежащая на  $l$ , — неподвижна, то метим треугольник  $i$  для разбиения.

**Шаг 2-3.** Если есть помеченные треугольники, совершаем про-

Рис. 1. Начальная сетка



(а) Начальная сетка и заданные ломаные

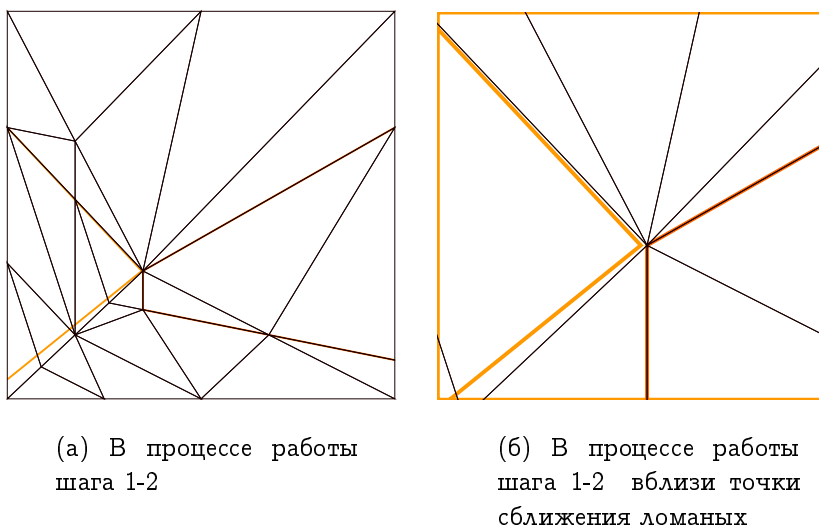
(б) Начальная сетка вблизи точки сближения ломаных

цедуру бисекции и переходим к шагу 2-1.

**Шаг 2-4.** Если все треугольники имеют  $\delta$ -качество аппроксимации равно единице, то переходим к этапу 3.

**Шаг 2-5.** Для каждого треугольника  $i$ , имеющего  $\delta$ -качество аппроксимации, отличное от единицы, находим вершину, ближайшую к одному из звеньев ломаных, которое пересекает данный треугольник не по ребру, и формируем из нее множество  $L_i$ . На соответствующем звене строим локальную сетку на той части, которая пересекает все три суперэлемента для трех вершин треугольника, образуем из нее множество  $R_i$  и вызываем процедуру проверки возможности сдвига  $\mathcal{S}(L_i, R_i, l_0, r_0, 2 * q_0)$ . Если она дает отрицательный результат и  $d(i) > \delta$ , то метим этот треугольник для последующего разбиения. Если она дает пару  $l_0$  и  $r_0$ , то сносим  $l_0$  в  $r_0$ ,

Рис. 2. В процессе работы



(а) В процессе работы шага 1-2

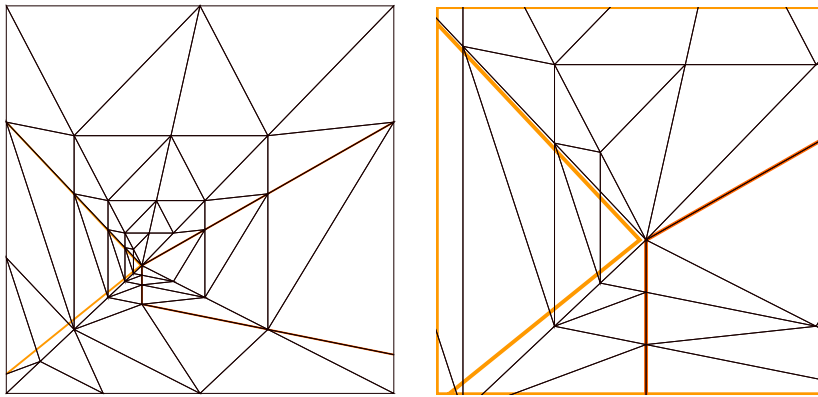
(б) В процессе работы шага 1-2 вблизи точки сближения ломаных

фиксируем  $l_0$ .

**Шаг 2-6.** Если на шаге 2-5 произошла сдвижка хотя бы одного узла, то считаем, что нет помеченных треугольников, и переходим к шагу 2-4. В противном случае совершаем процедуру бисекции и переходим к шагу 2-4.

В результате работы второго этапа получаем конформную треугольную сетку,  $\delta$ -качество аппроксимации треугольников в которой равно единице и качество формы треугольников не меньше  $q_0$ . Про сдвиги  $S$  узлов мы требуем, чтобы качество было не хуже  $2 * q_0$ , так как треугольники, которые уже аппроксимировали ломаные, при выполнении алгоритма могут подвергаться бисекциям для обеспечения конформности сетки. Но по свойству  $KBS$  их качество не ухудшится более чем в два раза, то есть их качество будет не

Рис. 3. После шага 1–2



(а) После шага 1–2

(б) После шага 1–2 вблизи точки сближения ломаных

меньше  $q_0$ . Из описания видно, что второй этап — это повторяющиеся процедуры выполнения шагов 2-1 — 2-3 и шагов 2-4 — 2-6. В качестве примера на (рис. 5–6) приведены сетки после второго и четвертого повторений шагов 2-4 — 2-6.

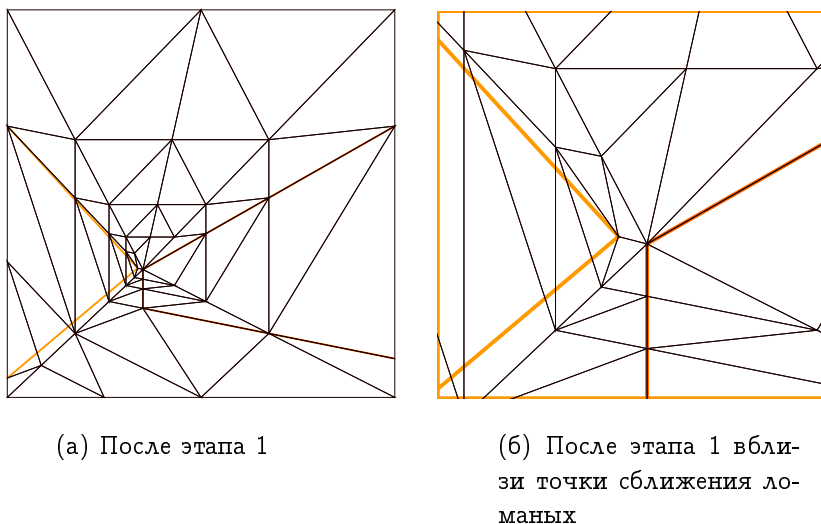
**Этап 3** — улучшение качества формы сетки за счет сдвижки подвижных узлов, если это возможно.

#### 4. Анализ алгоритма.

В этом разделе опишем основные свойства алгоритма и сделаем некоторые важные замечания.

**Свойство I.** Представленный алгоритм строит треугольную конформную квази-иерархическую сетку  $\epsilon_0$ , которая имеет  $\delta$ -качество аппроксимации  $\hat{Q}_{\epsilon_0}^\delta = 1$  и качество формы  $Q_{\epsilon_0} \geq q_0$ , предвари-

Рис. 4. После этапа 1



(а) После этапа 1

(б) После этапа 1 вблизи точки сближения ломаных

тельно при необходимости сдвинув ломаные на величину не более  $\delta$ .

Отметим, что в случае  $\delta < \varepsilon/4$  расстояние между любыми двумя непересекающимися звеньями остается меньше  $\varepsilon/2$ . Таким образом, взаимное расположение ломаных сохраняется после сдвига.

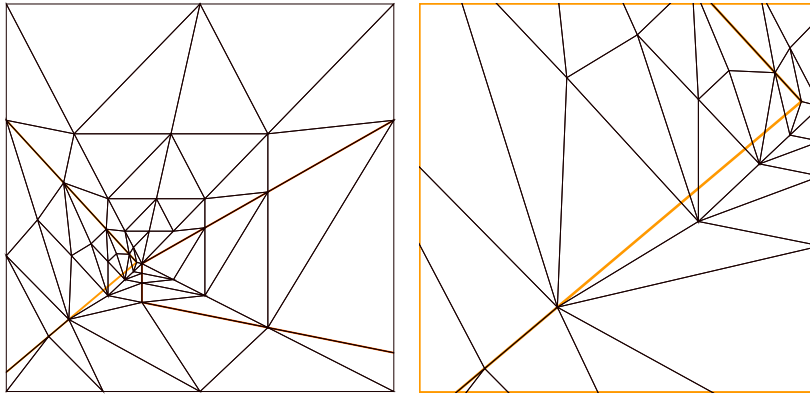
**Свойство II.** Алгоритм построения конформной треугольной квази-иерархической сетки, слабо аппроксимирующей ломаные с точностью  $\delta$ , конечен.

Действительно, поскольку диаметр треугольника не меньше  $\delta$ , то число уровней разбиения не превышает  $2 \log_2 \delta^{-1}$ , т. е. конечно.

**Свойство III.** Пусть сетка после шага 2-3 имеет качество формы  $\check{q}$ . Шаги 2-5—2-6 алгоритма могут ухудшить качество формы не больше, чем в  $2\kappa^2(\check{q})$  раз, где  $\kappa(\check{q})$  — константа из свойства  $SD$  опе-



Рис. 5. После двух повторений шагов 2-4 — 2-6



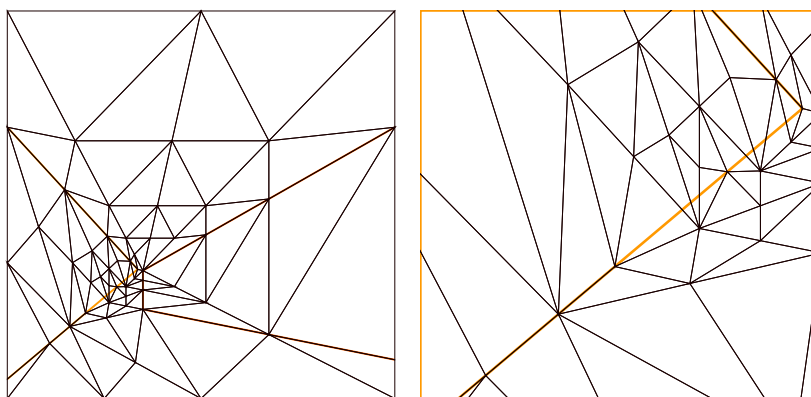
(а) После двух повторений шагов 2-4 — 2-6

(б) После двух повторений шагов 2-4 — 2-6 вблизи точки сближения ломаных

рации сдвига. Поэтому порог качества формы должен выбираться с ограничением  $q_0 < \tilde{q}/2\kappa^2(\tilde{q})$ . Действительно, рассмотрим отрезок звена ломаной, который еще не аппроксимирован ребрами треугольников, не содержит внутри узлов сетки, и не является частью другого большего отрезка, также не аппроксимированного ребрами треугольников. Отметим, что концы этого отрезка лежат в узлах сетки, а в силу окончания шагов 2-1 — 2-3 любой треугольник, пересекающий данный отрезок, не имеет неподвижных вершин, не лежащих на рассматриваемом отрезке.

Рассмотрим следующую последовательность действий для аппроксимации данного отрезка. Возьмем треугольник  $i$ , пересекающийся с данным отрезком не по ребру, и содержащий один из концов

Рис. 6. После четырех повторений шагов 2-4 — 2-6



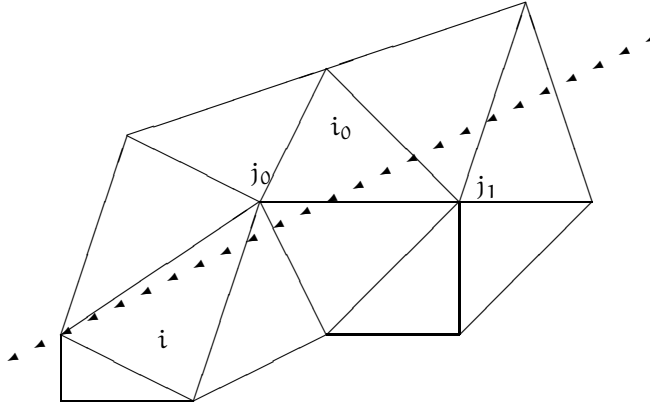
(а) После четырех повторений шагов 2-4 — 2-6

(б) После четырех повторений шагов 2-4 — 2-6 вблизи точки сближения ломаных

отрезка (рис. 7).

При данном малом  $q_0$  мы можем сдвинуть на ломаную ближайшую из вершин  $j_0$  данного треугольника. По свойству  $SD$  качество формы сдвинутого узла ухудшится не более, чем в  $k$  раз ( $\check{q}/k$ ). При этом сдвиг  $j_0$  сразу ставит на ломаную целое ребро, изменяя качество формы только у треугольников из суперэлемента  $j_0$ , но улучшая до единицы качество аппроксимации для треугольника  $i$  относительно звена, содержащего рассматриваемый отрезок. После сдвига  $j_0$  в суперэлементе  $j_0$  появляется лишь один треугольник (скажем  $i_0$ ) с плохим качеством аппроксимации относительно исследуемого звена, и содержащий узел  $j_0$  (рис. 8). Его качество формы не хуже  $\check{q}/k$ .

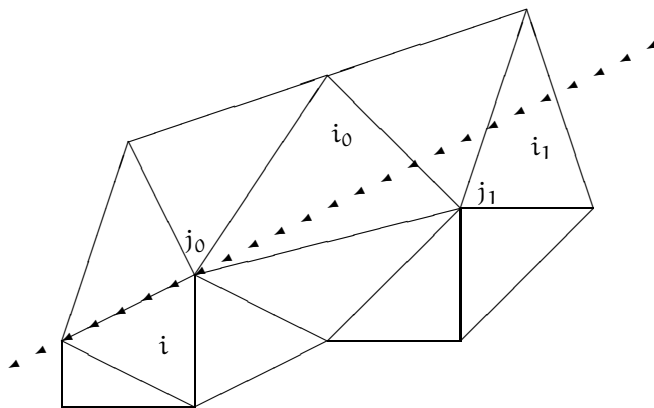
Рис. 7. Начальное расположение треугольников и звена ломаной



Далее, алгоритм сдвинет на ломаную узел  $j_1$ , ближайшую вершину треугольника  $i_0$ . Несмотря на возможно новое ухудшение качество формы треугольника  $i_0$  до  $\tilde{q}/\kappa^2$  после сдвига узла  $j_1$  (согласно свойству  $SD$ ), он начинает аппроксимировать ломаную. Поэтому далее суперэлемент  $j_0$  не будет подвергаться сдвигам.

По аналогии с узлом  $j_0$  после сдвига узла  $j_1$  у нас появится треугольник  $i_1$  с качеством аппроксимации, отличным от единицы, и содержащий вершину  $j_1$ . Этот треугольник имеет качество формы не меньше  $\tilde{q}/\kappa$  (свойство  $SD$ ), так как он не принадлежит суперэлементу  $j_0$  и до сдвига  $j_1$  имел качество формы не меньше  $\tilde{q}$ , поэтому для него можно применить операцию сдвига узла. Значит, возможен последовательный процесс аппроксимации ломаных, который не приведет к цепочке наращиваемого ухудшения качества формы каких-нибудь треугольников, а будет поддерживать качество формы аппроксимирующих треугольников не ниже  $\tilde{q}/\kappa^2$ . Последующие возможные бисекции аппроксимирующих треугольников для сохранения конформности сетки могут ухудшить их качество фор-

Рис. 8. Расположение треугольников и звена ломаной после сдвига узла.



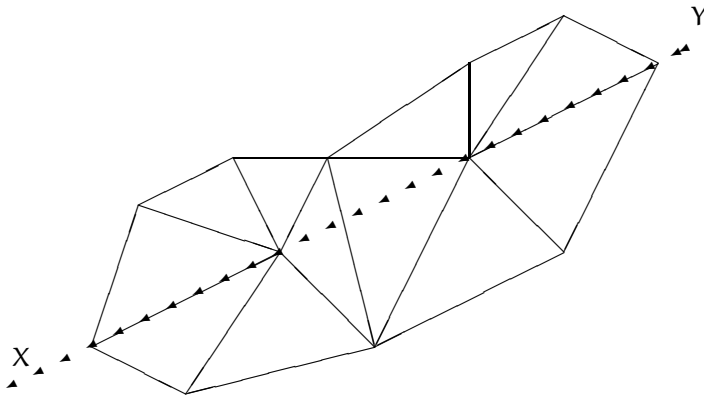
мы не более чем в два раза (до  $\tilde{q}/2\kappa^2$ ) по свойству  $\mathcal{KSD}$ .

Поскольку любая последовательность действий в шагах 2-5-2-6 представляется как совокупность последовательностей, аналогичных только что рассмотренной последовательности, оценки качества формы переносятся на случай произвольного выполнения сдвигов. Это и доказывает свойство III.

Теперь перейдем к вопросу об оценке числа треугольников в зависимости от величины  $\delta$ . Рассмотрим сетку, получающуюся равномерным измельчением исходной. Диаметр любого треугольника в полученной сетке меньше  $\delta$ , а поэтому имеет площадь меньше  $O(\delta^2)$ . Следовательно, число треугольников в результирующей сетке  $O(\delta^{-2})$ .

Оценим число треугольников для сетки, которая имеет сгущение к линиям. Рассмотрим любое звено заданной ломаной. Пусть его длина  $l$ , а число треугольников, которое оно пересекает (включая

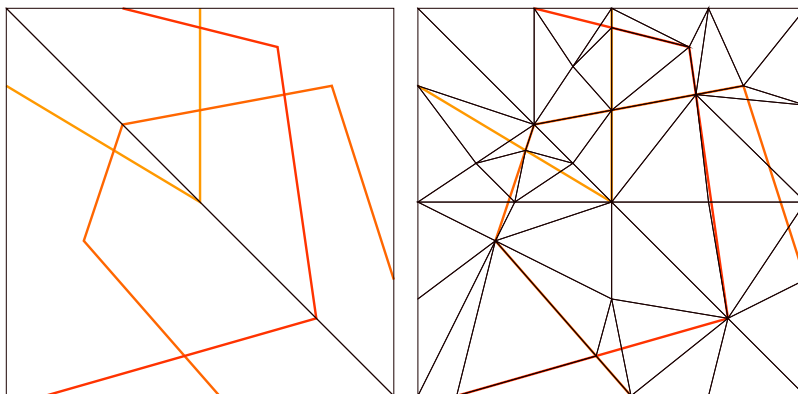
Рис. 9. Расположение треугольников, приводящее к возможному сгущению сетки



пересечение по стороне)  $n$ . Тогда  $l$  пропорционально  $n * \delta$ , т. е.  $n$  пропорционально  $l/\delta$ . Следовательно, число треугольников в такой сетке  $O(\delta^{-1})$ .

Теперь проанализируем алгоритм, описанный в предыдущем параграфе. Вопрос об оценке числа треугольников зависит от взаимного расположения заданных ломаных. Если  $\delta \geq \varepsilon$ , то алгоритм решения задачи слабой  $\delta$ -аппроксимации ломаных около точек сближения непересекающихся звеньев может породить сетку, сгущающуюся к линии. В частности, если существуют два звена, расположенные параллельно на расстоянии  $\varepsilon$ , то алгоритм произведет сетку со сгущением к звену, и поэтому имеющую число треугольников порядка  $O(\delta^{-1})$ . Поэтому в дальнейшем будем считать, что  $\delta \ll \varepsilon$ .

**Свойство IV.** Алгоритм, описанный в предыдущем параграфе, при  $\delta \ll \varepsilon$  будет иметь число треугольников не больше  $\log_2^2 \delta^{-1}$ . Это следует непосредственно из следующей теоремы.

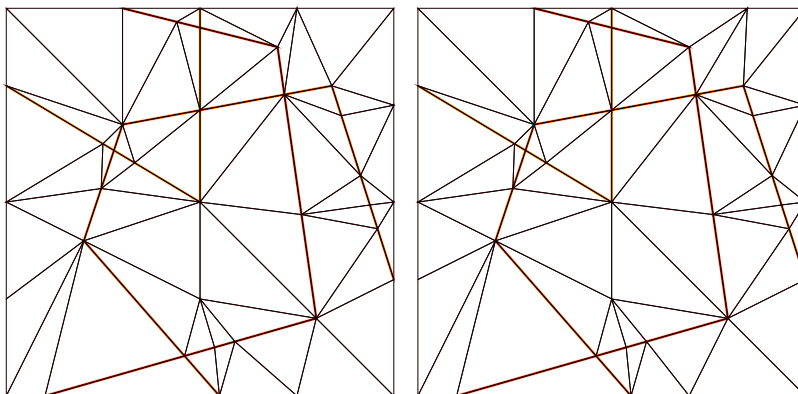
Рис. 10. Начальная сетка и сетка после этапа 1,  $q_0 = 0.15$ 

**Теорема.** Пусть дана грубая начальная конформная треугольная сетка, покрывающая заданную полигональную область  $\Omega$  и точно аппроксимирующая границу области  $\Omega$ . Описанный в предыдущем разделе алгоритм построения конформной квази-иерархической треугольной сетки, аппроксимирующей заданное множество ломаных с точностью  $\delta \ll \varepsilon$ , порождает сетку, сгущающуюся при достаточно малом  $q_0$  к числу точек не больше  $O(\log_2 \delta^{-1})$ .

Действительно, рассмотрим сетку, сгущающуюся к одной точке. Для достижения условия, что диаметр треугольника вблизи точки сгущения стал меньше  $\delta$ , необходимо совершить  $O(\log_2 \delta^{-1})$  бисекций, а каждая бисекция порождает лишь ограниченное число треугольников по свойству  $BS$ . Следовательно, сетка с одной точкой сгущения имеет  $O(\log_2 \delta^{-1})$  число треугольников. Если число точек сгущения  $O(\log_2 \delta^{-1})$ , то число треугольников в сетке имеет порядок  $(\log_2^2 \delta^{-1})$ .

**Доказательство.**

Из описания алгоритма легко видеть, что смысл первого этапа заключается в аппроксимации только лишь точек множества  $Z$ .

Рис. 11. Сетки после этапов 2 и 3,  $q_0 = 0.15$ 

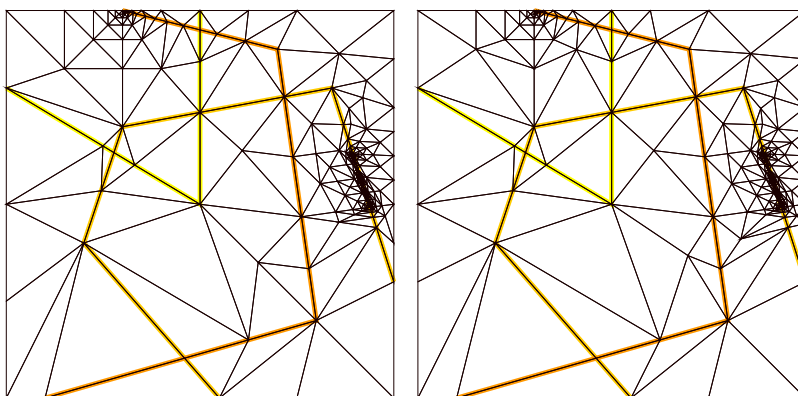
Поэтому по его окончании мы можем (в худшем случае) получить сетку, имеющую сгущение лишь к некоторым точкам из множества  $Z$ , число которых конечно.

Так как третий этап алгоритма вообще не производит бисекций, то для доказательства теоремы достаточно показать, что именно второй этап алгоритма при достаточно малом  $\delta$  производит сетку, сгущающуюся к числу точек порядка  $O(\log_2 \delta^{-1})$ .

Так как  $\delta \ll \varepsilon$ , число повторений шагов 2-1 — 2-3 не зависит от  $\delta$ . Действительно, поскольку расстояние от любой неподвижной точки до любого звена, не содержащего эту точку, не меньше  $0.9\varepsilon$ , то измельчение треугольников, пересекающих звено, до диаметра  $0.5\varepsilon$  обеспечит отсутствие у них неподвижных вершин. Получаем, что число повторений шагов 2-1 — 2-3 определяется лишь расположением задаваемых ломаных и не зависит от  $\delta$ . Поэтому проанализируем лишь шаги 2-4 — 2-6. Пусть после шага 2-3 сетка имела качество формы  $\tilde{q}$ , а  $q_0 = 0.1 * \tilde{q}$ .

Рассмотрим любой отрезок ломаной, который перед шагом 2-4 не аппроксимирован ребрами треугольников, а его концы  $X$  и  $Y$  сов-

Рис. 12. Сетки после этапов 2 и 3,  $q_0 = 0.21$ . Сгущения порождены шагами 2.4-2.6



падают с узлами сетки. Не ограничивая общности, можно считать, что ни один из узлов сетки не лежит на этом отрезке, в противном случае, выделим из взятого отрезка подотрезок меньшей длины без внутренних сеточных узлов. Пусть число треугольников, пересекаемых отрезком  $XU$  равно  $m$ . Так как на первом этапе алгоритм производит сетку, сгущающуюся к некоторым (или никаким) из точек  $Z$ , то число  $m$  не превышает  $O(\log_2 \delta^{-1})$ .

Начав работу с этим отрезком, алгоритм станет сносить на него вершины. В силу малости  $q_0$  шаги 2-4 — 2-6 будут проходить без бисекций до тех пор, пока среди  $m$  пересекающих  $XU$  треугольников не выделится  $m_1$  пар треугольников, имеющих общее ребро, пересекающее  $XU$ , и противоположные общему ребру вершины, лежащие на  $XU$  (рис. 9). Получив множество таких пар, алгоритм попытается снести на ломанную ближайшую общую вершину двух треугольников. Это не всегда возможно, так как сносимая вершина принадлежит суперэлементам сразу двух вершин, уже снесенных на  $XU$  в результате последовательных процессов аппроксимации лома-



ной, происходящих по разные стороны от рассматриваемой пары треугольников. Ограничение снизу на качество формы этих суперэлементов может воспрепятствовать сносу вершины, обеспечивающей аппроксимацию ломаной. В этом случае начнется процесс бисекций, который будет порождать сетку с сгущением к точке пересечения общего ребра исследуемой пары треугольников и отрезка  $XU$ . Поэтому в результате работы алгоритма мы получим сетку, сгущающуюся не более чем к  $m_1$  точкам, число которых не более, чем  $O(\log_2 \delta^{-1})$ .  $\square$

Отметим практическую важность данной теоремы: улучшение аппроксимации ломаных ( $\delta \rightarrow 0$ ) требует умеренного роста числа элементов сетки, что позволяет строить сетки с небольшим числом треугольников.

## 5. Примеры сеток, построенных с помощью алгоритма.

Рассмотрим два примера работы нашего алгоритма, в которых иллюстрируется свойство IV. Напомним, что оценка числа треугольников в теореме является оценкой сверху: появление точек сгущения на этапах 1 и 2 зависит от взаимного расположения заданных ломаных и начальной сетки и от порога качества формы треугольников  $q_0$ . Как правило, при умеренных значениях  $q_0$  сгущения не происходит ни на этапе 1, ни на этапе 2 даже при малых  $\delta$ , как это продемонстрировано на рис. 10–11 для  $\delta = 0.008$  и  $q_0 = 0.15$ . Более высокий порог качества формы приводит к появлению точек сгущения, см. рис. 12. Отметим, что малость  $\delta$  влияет лишь на глубину сгущения, но не на появление точек сгущения.

## 6. Заключение

В работе рассмотрено несколько постановок оптимизационных задач по генерации сеток, удовлетворяющих множественным ограничениям. Для одной из них предложен и проанализирован алгоритм построения квази-иерархической сетки, которая может рассматриваться как некоторое приближение к решению в соответст-

ющем классе сеток. Характерной особенностью алгоритма является его робастность, что достигается аппроксимацией ограничений с точностью  $\delta$  и возможностью локального  $\delta$ -возмущения заданных ограничений. Улучшение аппроксимации ломаных ( $\delta \rightarrow 0$ ) приводит к росту числа элементов сетки, не превышающем  $\log_2^2 \delta^{-1}$ .

### Список литературы

- [1] Скворцов А. Триангуляция Делонэ и ее применение. — Томск, Издательство Том. ун-та. 2002.
- [2] Шайдунов В. В. Многосеточные методы конечных элементов. — Москва, Наука. 1989.
- [3] Baker B., Grosse E., Rafferty C. S. Nonobtuse triangulation of polygons // *Disc. and Comput. Geom.* 1998. V. 3. P. 147–168.
- [4] Bänsch E. Local mesh refinement in 2 and 3 dimensions // *IMPACT of Computing in Science and Engrg.* 1991. V. 3. P. 181–191.
- [5] Bern M., Eppstein D., Gilbert J. R. Provably good mesh generation // *Proceedings of the 31st Annual Symposium on Foundations of Computer Science.* 1990. IEEE. P. 231–241.
- [6] Buscaglia G., Dari E. Anisotropic mesh optimization and its application in adaptivity // *Inter. J. Numer. Meth. Engng.* 1997. V. 40. P. 4119–4136.
- [7] Carey G. Computational grids. Generation, adaptation, and solution strategies. — London. Taylor and Francis. 1997.
- [8] Chew L. P. Guaranteed-quality mesh generator for curved surfaces // *Proceedings of the Ninth Annual Symposium on Computational Geometry.* 1993. ACM. P. 274–280.
- [9] Frey P., George P.-L. Maillages: applications aux élément finis. — Paris. Hermès Science Publications. 1999.
- [10] Liseikin V. Grid generation methods. — Springer. Berlin Heidelberg. 1999.

- 
- [11] Rivara M. Mesh refinement processes based on the generalized bisection of simplexes // *SIAM J. Numer. Anal.* 1984. V. 21. P. 604–613.
- [12] Ruppert J. A Delaunay refinement algorithm for quality 2-dimensional mesh generator // *Journal of Algorithms*. 1995. V. 18. N. 3. P. 548–585.
- [13] Shewchuk J. Delaunay refinement algorithms for triangular mesh generation // *Computational Geometry: theory and applications*. 2002. V. 22 (1-3). P. 21–74. См. также <http://www-2.cs.cmu.edu/~jrs/jrspapers.html#cdt>
- [14] Vassilevski Yu., Lipnikov K. Adaptive algorithm for generation of quasi-optimal meshes // *Comp. Math. Math. Phys.*. 1999. V. 39. P. 1532–1551.
- [15] <http://www-users.informatic.rwth-aachen.de/~roberts/meshgeneration.html>
- [16] [http://www.engr.usask.ca/~macphed/finite/fe\\_resources/mesh.html](http://www.engr.usask.ca/~macphed/finite/fe_resources/mesh.html)