

УДК 519.6

РЕАЛИЗАЦИЯ ФРОНТАЛЬНОГО АЛГОРИТМА ПОСТРОЕНИЯ НЕРЕГУЛЯРНЫХ ЧЕТЫРЕХУГОЛЬНЫХ СЕТОК

А. В. Сковпень
(РФЯЦ-ВНИИТФ)

Разработаны двумерные алгоритм и программа построения нерегулярных четырехугольных сеток. Алгоритм построен на базе алгоритма Q-Morph, предложенного С. Оуэном и относится к непрямым фронтальным методам построения четырехугольных нерегулярных сеток. При построении сетки используются предварительная триангуляция области и последующая трансформация треугольников с образованием четырехугольных ячеек. Исходными данными являются координаты узлов границы. Алгоритм работает в автоматическом режиме и не модифицирует граничных узлов. Получаемая сетка является полностью четырехугольной, если граница области состоит из четного числа ребер. При нечетном количестве ребер границы в сетке появляется одна треугольная ячейка. Созданная на основе алгоритма программа может быть использована в качестве независимого сеточного модуля в программах численного моделирования.

Введение

Нерегулярная сетка является инструментом, позволяющим моделировать поведение объектов со сложной геометрией. Она, в отличие от регулярной, может быть автоматически построена для произвольного контура в процессе расчета, что позволяет создать геометрически безавестную методику на подвижных сетках. Возможность создания полностью автоматического сеточного ядра обуславливает широкое использование нерегулярных сеток в коммерческих программах инженерных расчетов. Для регулярных или блочно-регулярных сеток создание аналогичного полностью автоматического строителя не является окончательно решенной проблемой. Это связано с особыми условиями, накладываемыми на положение и разбиение границ.

Методы построения нерегулярной четырехугольной двумерной сетки можно разделить на прямые и не прямые. Прямые методы используют непосредственное построение ячеек внутри заданного контура. К ним относятся фронтальные методы [1–3] и методы, использующие блочную декомпозицию или суперпозицию. Непрямые методы используют предварительно построенную треугольную сетку. Их можно разделить на два направления: 1) методы, строящие четырехугольные ячейки путем объединения и разбиения треугольных ячеек [4–6]; 2) фронтальные методы, использующие объединение, разбиение треугольных ячеек и другие топологические операции с треугольниками [7, 8].

Из всех этих методов наиболее эффективным по ряду критериев является метод Q-Morph, предложенный Стивеном Оуэном в 1998 г. Сетка в нем строится столь же качественная, как и в лучших прямых фронтальных методах, а используемая трансформация треугольников делает его надежным и быстрым.

Этот метод был взят автором данной статьи в качестве базового для создания алгоритма и программы построения нерегулярных четырехугольных сеток. В процессе разработки в базовый алгоритм были внесены новшества и некоторые его части были модифицированы. Эти изменения улучшили надежность алгоритма и в некоторых случаях качество сетки. Кроме того, для алгоритма была разработана эффективная технология топологической оптимизации. В данной работе представлено краткое описание модифицированных частей разработанного алгоритма (включая

сжатое изложение технологий из метода Q-Morph) и приведены примеры построения сетки. Рабочее название описываемого в данной работе алгоритма — QMV (Q-Morph Variation).

1. Краткая схема алгоритма

Предлагаемый метод построения нерегулярных сеток основан на триангуляции области и последующей трансформации треугольников в четырехугольные ячейки. Трансформация треугольников в четырехугольники происходит по слоям, начиная от исходной границы. Пример работы алгоритма приведен на рис. 1.

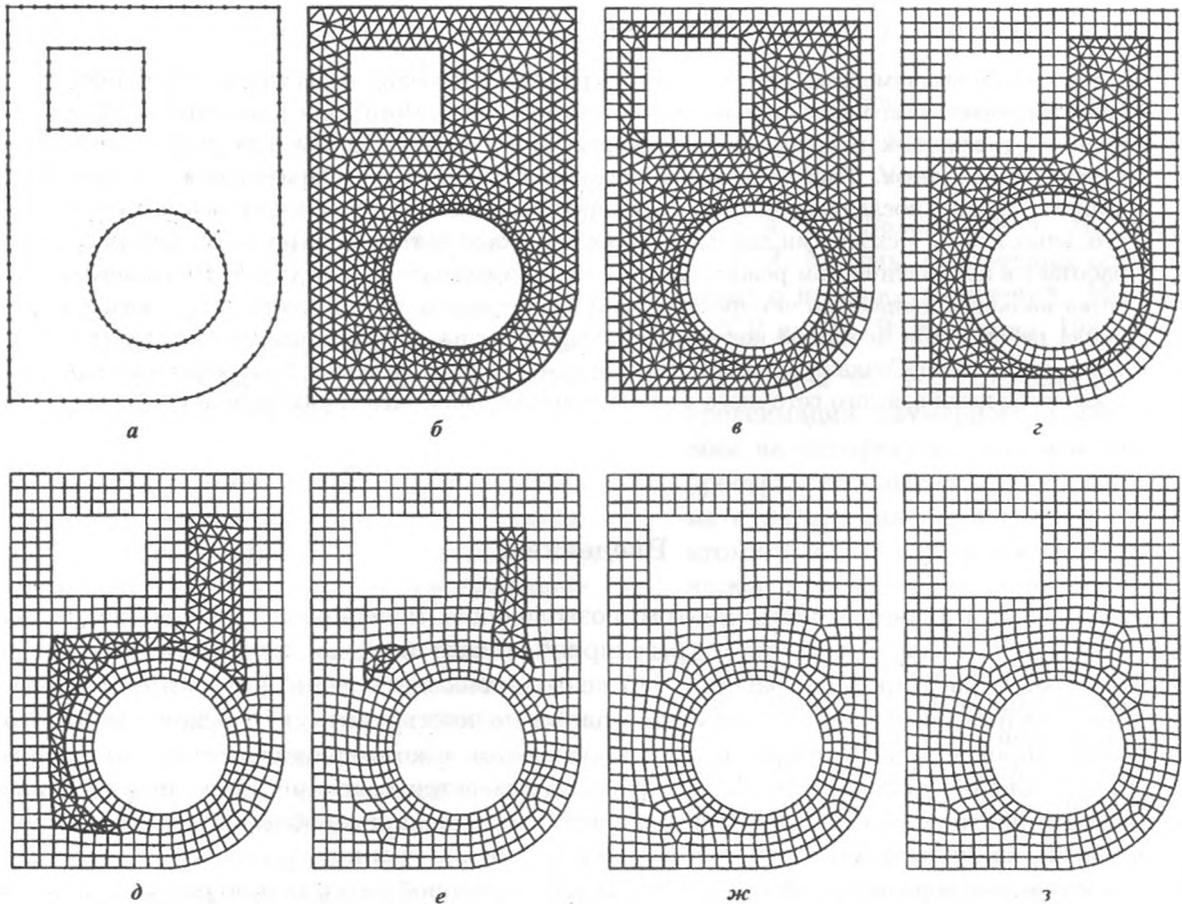


Рис. 1. Этапы построения сетки: *a* — начальный контур; *б* — триангуляция; *в* — один слой; *г* — два слоя; *д* — три слоя; *е* — четыре слоя; *ж* — пять слоев; *з* — сглаживание и топологическая оптимизация

Алгоритм состоит из следующих основных этапов:

1. *Построение начальной треугольной сетки.* Для построения начальной треугольной сетки подходит любой алгоритм, строящий качественную сетку. В данной работе был использован алгоритм AFLR, предложенный в [9]. Желательно, чтобы треугольная сетка соответствовала размерной функции для четырехугольной сетки. Размерная функция — "желаемый" характерный размер ячейки (длина ребра), заданный в каждой точке пространства построения сетки. Обычно эта функция задается до построения сетки.
2. *Определение подвижного фронта.* Подвижный фронт — это замкнутая совокупность ребер, отделяющая уже построенные четырехугольные ячейки от треугольных. На начальном этапе фронт совпадает с границей области. Ссылки на ребра подвижного фронта хранятся в отдельном списке. После любых топологических операций, затрагивающих ребра подвижного

- фронта, необходимо приводить список ребер подвижного фронта в соответствие с текущим состоянием сетки. Алгоритм завершает работу, когда не остается ребер подвижного фронта.
3. *Проверка на специальные случаи закрытия фронта.* Если текущий фронт содержит 3, 4 или 6 ребер, то используются особые процедуры закрытия фронта с формированием новых ячеек.
 4. *Классификация ребер фронта.* Ребра фронта сортируются в соответствии с несколькими критериями. Этот процесс повторяется после построения каждой новой ячейки или любой топологической операции, затрагивающей ячейки, граничащие с ребрами фронта. Цель сортировки — найти наиболее подходящее для построения следующей новой ячейки ребро фронта (*активное ребро*).
 5. *Проверка на специальные случаи модификации фронта.* Для ребер фронта выполняется проверка на наличие специальных случаев. Если такие ситуации обнаружены, то производится локальное перестроение сетки и осуществляется переход на п. 8.
 6. *Построение новой ячейки от ребра фронта.* Новая четырехугольная ячейка строится от активного ребра фронта. Для построения ячейки используются две различные процедуры. Первая — основная и более быстрая, вторая используется в топологически сложных случаях. Если первая окончилась неудачно, т. е. новая ячейка не была построена, то используется вторая, которая гарантирует построение новой ячейки. После построения новой ячейки производится локальное сглаживание.
 7. *Проверка на специальные случаи замыкания фронта.* Если новая ячейка делит фронт на два или три фронтальных контура, производится проверка на соблюдение условия четности количества ребер в новых фронтах. При необходимости производятся специальные операции для обеспечения этого условия. Это позволяет строить полностью четырехугольную сетку.
 8. *Проверка наличия ребер подвижного фронта.* Если существуют ребра подвижного фронта, выполняется переход на п. 2.
 9. *Топологическая оптимизация.* Выполняются топологические операции, направленные на оптимизацию связности сетки.
 10. *Финальное сглаживание.* Выполняется сглаживание всех ячеек области.

2. Реализация алгоритма

2.1. Классификация ребер фронта. Среди ребер фронта на каждом шаге выбирается активное ребро — наиболее подходящее для создания новой ячейки. С этой целью производится их сортировка.

Используются несколько критериев, характеризующих состояния ребра:

state — два бита, первый характеризует состояние левого узла, второй — правого. Если угол между ребрами фронта в узле не превышает ε_{side} , то бит равен 1, иначе равен 0 (рис. 2); $\varepsilon_{side} = 3\pi/4$;
status0 определяется так:

при *state* = '00' *status0* = 0; при *state* = '10' *status0* = 1;

при *state* = '01' *status0* = 1; при *state* = '11' *status0* = 2;

status1 = 1, если у ребра есть узел, в котором бит *state* равен 1, и оба прилегающих к этому узлу фронтальных ребра принадлежат границе, иначе *status1* = 0;

string_level — количество слоев четырехугольных ячеек между ребром и границей области. Если ребро граничное, то *string_level* = 0. Если под ребром построен один слой четырехугольных ячеек, то *string_level* = 1 и т. д.;

spf_ratio — отношение длины ребра к значению размерной функции;

length — длина ребра.

Для процедуры сортировки фронтальных ребер определяется оператор < (*меньше*). Сортировка расставляет ребра (*edge*) по возрастанию. Новая ячейка строится от ребра в начале списка (оно *меньше* всех).

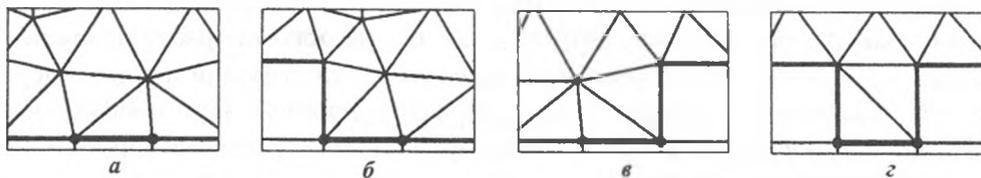


Рис. 2. Состояние ребра фронта: а — state='00'; б — state='10'; в — state='01'; г — state='11'

```
operator< (edge1, edge2);
{
// приоритет для углов на границе
if(edge1.status1!=edge2.status1) return(edge1.status1>edge2.status1);
// приоритет для ребер, у которых state = '11'
if(edge1.status0=2 и edge2.status0≠2 и edge1.spf_ratio<1.25) return true;
if(edge1.status0≠2 и edge2.status0=2 и edge2.spf_ratio<1.25) return false;
// приоритет для строк ячеек, расположенных ближе к границе
if(edge1.string_level!=edge2.string_level)return(edge1.string_level<edge2.string_level);
// если оба ребра имеют state= '10' или '01', то приоритет у ребра с меньшей длиной
if(edge1.status0=1 и edge2.status0=1) return(edge1.length<edge2.length);
// приоритет для ребра с большим status0
if(edge1.status0!=edge2.status0) return(edge1.status0>edge2.status0);
// приоритет для ребра с меньшей длиной
return(edge1.length<edge2.length);
}
```

2.2. Построение новой ячейки от ребра фронта. Новая четырехугольная ячейка строится от активного ребра, которое берется из начала отсортированного списка фронтальных ребер. Алгоритм базируется на двух попытках построения новой ячейки. Им соответствуют две процедуры.

Первая процедура заимствована из базового алгоритма Q-Morph [8]. Но она не может гарантировать построения новой ячейки. В ситуациях, когда другие ребра фронта расположены близко к активному ребру фронта, часто нет возможности определить боковые ребра или восстановить верхнее ребро. Этому могут мешать уже построенные четырехугольные ячейки.

Если попытка построить ячейку, используя первую процедуру, окончилась неудачно, то используется вторая процедура. Она фактически перебирает все возможные варианты формирования новой ячейки, базируясь на ребрах, примыкающих к активному ребру. Для каждого из вариантов, отсортированных по критерию, в основе которого лежит мера качества ячейки (см. ниже), производится попытка построения новой ячейки. Если какая-либо из них оказалась успешной, то процедура на этом завершается. Алгоритм второй процедуры гарантирует построение невывернутой четырехугольной ячейки. Это повышает надежность и устойчивость алгоритма.

Первая процедура построения ячейки. Первая процедура построения новой ячейки полностью аналогична процедуре построения новой ячейки, описанной в [1]. Краткий алгоритм ее следующий (рис. 3):

1. *Определение боковых ребер.* Определяются два боковых ребра, наиболее подходящие для построения четырехугольной ячейки. Боковые ребра определяются из существующих (если такие есть в секторе, изображенном на рис. 3, б) или восстанавливаются переключением ребер (приложение 1). Если в указанном секторе нет узла, подходящего для восстановления ребра, то ребро, противолежащее узлу активного ребра, разбивается и формируется новый узел.
2. *Восстановление верхнего ребра,* соединяющего верхние узлы боковых ребер. Используется стандартная процедура восстановления ребра (приложение 2).

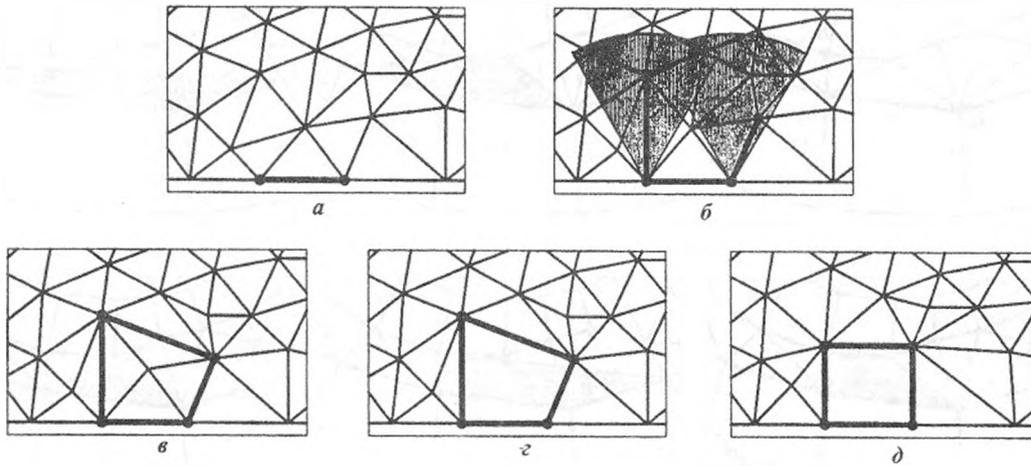


Рис. 3. Первая процедура построения ячейки: а — определение ребра; б — определение боковых ребер; в — восстановление верхнего ребра; г — создание ячейки; д — сглаживание

3. *Формирование четырехугольной ячейки.* Производится слияние треугольных ячеек, находящихся внутри контура, образованного ребром подвижного фронта, боковыми и верхним ребрами. Образуется новая четырехугольная ячейка.

Вторая процедура построения ячейки. Эта процедура используется в случае, если ячейку не удалось построить с использованием первой процедуры. В основе процедуры лежит перебор всех возможных вариантов построения новой ячейки на основе ребер, примыкающих к узлам активного ребра фронта. Графически это продемонстрировано на рис. 4.

Алгоритм второй процедуры построения новой ячейки:

1. *Формирование двух наборов узлов — set.1 и set.2.* Set.1 формируется для левого узла активного ребра, set.2 — для правого. В каждый набор включаются узлы, лежащие в области триангуляции или на фронте и имеющие общее ребро с соответствующим узлом активного ребра.
2. *Создание набора set.pairs, состоящего из пар узлов.* В набор set.pairs включаются все возможные комбинации пар узлов — один узел из set.1, второй из set.2.
3. *Сортировка пар узлов из set.pairs.* В основе критерия сортировки используется мера качества четырехугольных ячеек, которые могли бы быть построены от активного ребра, если в качестве верхних левого и правого узлов такой ячейки взять пару узлов из set.pairs. На основе этого критерия пары узлов в set.pairs сортируются таким образом, чтобы попытки построить ячейку начинались с самых "лучших" вариантов.
4. *Попытки построения ячеек с использованием пар узлов из set.pairs.* При построении выполняется определение и при необходимости восстановление двух боковых и верхнего ребра. Если все ребра строящейся ячейки определены, то формируется четырехугольная ячейка (выполняется слияние треугольников, как в первой процедуре построения ячейки).

Критерий сортировки пар узлов основан на *мере качества ячейки* [10]. Описание этой меры дано в подразд. 2.7. Из набора в set.pairs заранее удаляются пары узлов, которые могут дать вывернутую четырехугольную ячейку.

На рис. 4 продемонстрирована работа второй процедуры построения ячейки. Ребро E_a — активное ребро фронта. Построить от него ячейку, используя первую процедуру, невозможно, так как восстановлению верхнего ребра мешают ребра фронта. Создать новые боковые ребра методами первой процедуры также невозможно. Ребра, противолежащие узлам N_1 и N_2 , нельзя разбить или сделать их переключением, так как они фронтальные. На рис. 4 изображены варианты создания ячеек (они отмечены серым цветом). Показаны только варианты, для которых могут быть восстановлены

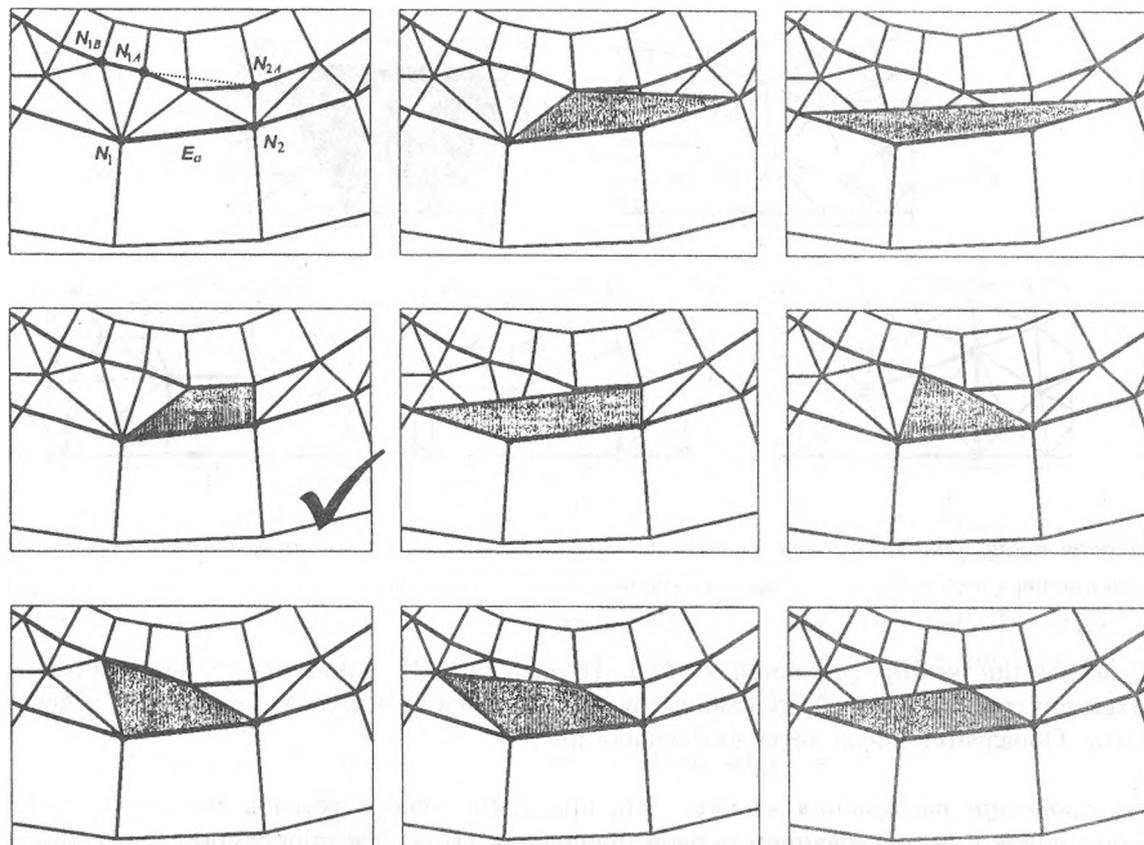


Рис. 4. Пример использования второй процедуры построения ячейки

ребра между узлами. Среди них наилучший вариант изображен во втором ряду слева (отмечен "галочкой"). Формируется новая ячейка, соответствующая этому варианту, и процедура завершает работу.

Вторая процедура гарантированно строит четырехугольную ячейку. Внутри фронтальных ребер при построении новой ячейки должно находиться более двух треугольников. Это гарантируется проверкой на специальные случаи закрытия фронта. Следовательно, треугольная ячейка, прилегающая к активному ребру фронта, имеет по крайней мере один смежный с ней треугольник. Объединив два невывернутых треугольника, всегда можно получить невывернутый четырехугольник. Такая комбинация — прилегающий к активному ребру треугольник и треугольник, смежный с ним, — попадает в набор вариантов, анализируемых во второй процедуре. Таким образом, новая ячейка будет построена в любом случае. Гарантируется, что она не будет вывернутой. Хотя очевидно, что она может быть невыпуклой и с "плохой" мерой качества, но эта проблема решается локальным сглаживанием.

2.3. Проверка на специальные случаи полного закрытия фронта. Если текущий фронт содержит три или четыре ребра, то все треугольные ячейки внутри ребер фронта удаляются и из этих ребер формируется треугольная или четырехугольная ячейка. Данная проверка производится до построения новой ячейки. Если фронт содержит три ребра, то, скорее всего, начальный контур имел нечетное число ребер и это первая и последняя треугольная ячейка в области.

Если фронт состоит из шести ребер, то вызывается процедура обработки шестиреберного контура, используемая при топологической оптимизации (подразд. 2.6). Она формирует две или три четырехугольные ячейки или полностью закрывает этот контур, уничтожая все треугольные ячейки внутри него. После специальных случаев полного закрытия фронта проводится локальное сглаживание.

2.4. Проверка на специальные случаи модификации фронта. Для ребер фронта производится проверка на наличие специальных случаев. Они связаны со специфическим положением

ребер фронта или с необходимостью обеспечить соответствие ячеек размерной функции. Проверка на наличие таких ситуаций производится на каждом шаге для всех ребер фронта, положение или связность которых изменились на последнем шаге. Если такие ситуации обнаружены, то производится перестроение сетки с образованием новых четырехугольных ячеек или без такового. После этого проводится локальное сглаживание. В алгоритме QMV используются операции модификации фронта, предложенные в работах [2, 8].

2.5. Проверка на специальные случаи замыкания фронта. Если количество ребер фронта нечетное, то невозможно построить внутри него сетку, полностью состоящую из четырехугольников. Придется вводить по крайней мере один треугольник. Поэтому важно отслеживать ситуации, когда количество ребер фронта может стать нечетным. Это может произойти, если построенная четырехугольная ячейка разделит фронт на два или три фронтальных контура.

Пример приведен на рис. 5. До замыкания фронта количество ребер фронта было четным — 14. При построении ячейки C_0 фронт разделился на два нечетных фронта: 5 и 9 ребер.

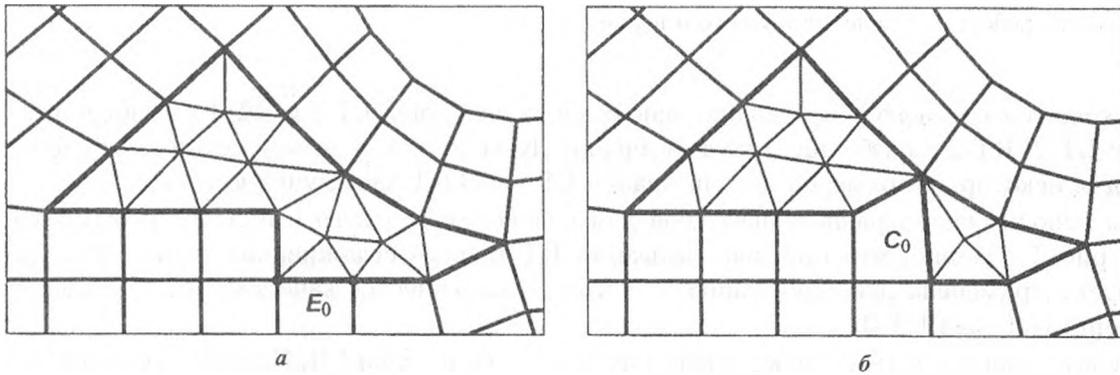


Рис. 5. Замыкание фронта с образованием двух нечетных фронтов: а — до замыкания фронта; б — после замыкания фронта

Ситуация замыкания фронта встречается достаточно часто, поэтому если не решать проблему нечетных контуров, то в построенной сетке окажется достаточно много треугольников. В [8] предлагается такой подход: если при построении ячейки боковое ребро замыкает фронт с образованием нечетных контуров, то оно делится узлом посередине. Из него образуются два ребра.

У такого подхода есть недостатки:

- возможно чрезмерное измельчение ребер;
- нет гарантии, что ячейка будет построена с использованием разделенного бокового ребра;
- нет гарантии, что фронт вообще замкнется: ребра фронта могут сойтись вплотную, пережать треугольные ячейки до предельной точности, но замыкания не произойдет.

Для сохранения четности фронтов был использован другой подход, основанный на делении ячеек, замкнувших фронт. Ячейка делится в соответствии с несколькими шаблонами в зависимости от того, к каким ребрам ячейки прилегают образовавшиеся нечетные фронты.

Пусть в ячейке обнаруживается пара ребер, таких что они принадлежат разным нечетным фронтам. Тогда ячейка делится на две или три ячейки таким образом, что каждое из этих ребер делится на два. В результате к каждому из нечетных фронтов добавляется по ребру и они становятся четными.

На рис. 6 приведены используемые шаблоны деления ячейки. При делении ячейки на двух ее ребрах создаются новые узлы. Треугольники, прилегающие к этим ребрам, делятся на два треугольника каждый.

Шаблонов деления ячейки можно построить больше. Для построения ячеек внутри контура из шести ребер их всего пять. Но те, что показаны на рис. 6, дают наилучший результат. Для

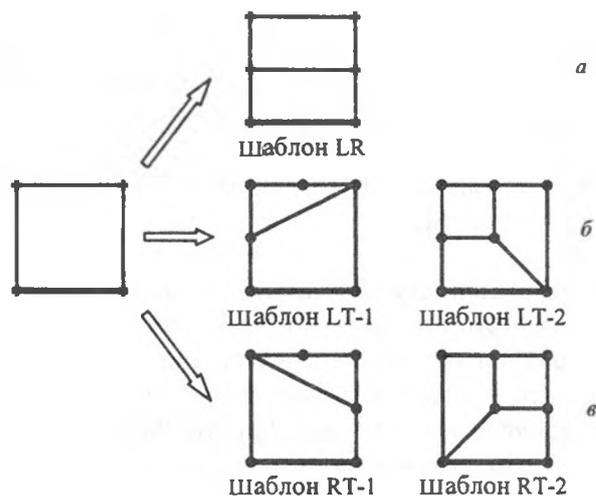


Рис. 6. Шаблоны деления ячейки, замыкающей фронт: *а* — деление левого и правого ребер; *б* — деление левого и верхнего ребер; *в* — деление правого и верхнего ребер

деления бокового и верхнего ребер можно использовать шаблоны LT-1 и RT-1 с разбиением на две ячейки или LT-2, RT-2 с разбиением на три ячейки. Качество получаемой сетки почти одинаково, но все же для некоторых примеров использование LT-1 и RT-1 дает лучшую сетку.

Примеры использования разных шаблонов для разбиения верхнего и правого ребер ячейки показаны на рис. 7. Видно, что при использовании RT-1 после сглаживания появляются ребра, к которым будет применена операция сшивки. Тогда топологическое качество будет выше, чем при использовании шаблона RT-2.

Для деления левого и правого ребер ячейки используется шаблон LR. Пример приведен на рис. 8. Если исходный фронт был нечетным (допустим, граница области содержит нечетное количество ребер), может сложиться ситуация, когда все три ребра ячейки, замкнувшей фронт, принадлежат различным нечетным фронтам. Тогда используется шаблон LR. Он сделает четными фронты слева и справа, верхний же останется нечетным. Хотя в данной ситуации можно использовать любой другой шаблон, LR топологически предпочтительнее.

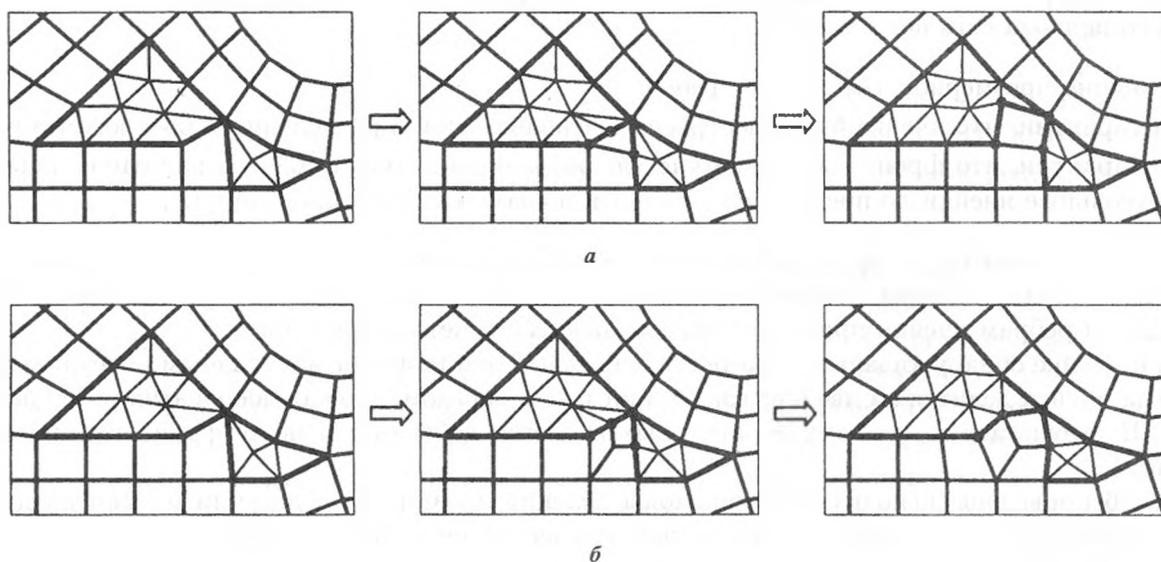


Рис. 7. Деление ячейки для верхнего и бокового нечетного фронтов: *а* — использование шаблона RT-1; *б* — использование шаблона RT-2

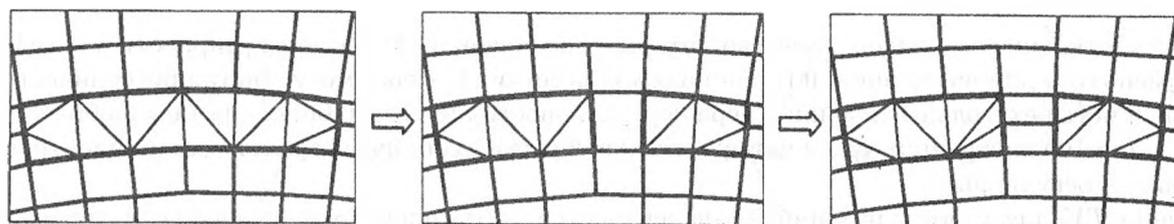


Рис. 8. Деление ячейки для левого и правого нечетного фронта (шаблон LR)

2.6. Топологическая оптимизация. Топологическая оптимизация — это совокупность операций, предназначенных для улучшения топологического качества сетки. Для четырехугольной сетки считается, что внутренний узел обладает оптимальной топологией, если он принадлежит четырем ячейкам. Такой узел называют регулярным. Среднее значение углов при этом узле в ячейках равно $\pi/2$. Если количество ячеек, связанных с внутренним узлом, не равно четырем, то узел называют нерегулярным. Главной целью топологической оптимизации является уменьшение количества нерегулярных узлов.

Если топология сетки плохая, то никакое сглаживание не поможет получить приемлемую по качеству сетку. Поэтому топологическое качество сетки — очень важный критерий, особенно при построении сеток максимально возможного качества. В данной работе используются базовые элементарные топологические операции, предложенные в разное время в работах [2, 1, 11]. Метод оценки топологического качества сетки и приоритетность операций построены на основе подхода, предложенного в [11].

При проведении топологических операций необходим некий критерий топологического качества, который может быть использован для оценки необходимости проведения операции и ее приоритетности по отношению к другим операциям. В качестве такого критерия используется мера нерегулярности TV (topological variance) в виде

$$TV = \frac{900\Delta n^2}{(\Delta n + 4)^2}. \tag{1}$$

Здесь Δn определяется следующим выражением:

$$\Delta n = \min \left(n - 1, \text{nint} \left(\frac{\sum_{i=1}^n \alpha_{i,ideal} - \alpha}{\pi/2} \right) \right),$$

где n — s -валентность узла (количество ячеек, имеющих общую вершину в узле); $\alpha_{i,ideal}$ — идеальный угол при узле в ячейке i : $\alpha_{ideal} = \pi/3$ для треугольной ячейки, $\alpha_{ideal} = \pi/2$ для четырехугольной ячейки; α — суммарный угол при узле во всех ячейках, присоединенных к нему, для внутреннего узла $\alpha = 2\pi$, для граничного узла $\alpha \leq 2\pi$; $\sum_{i=1}^n \alpha_{i,ideal}$ — сумма идеальных узлов всех ячеек, присоединенных к узлу. Функция nint определяет ближайшее целое к действительному числу. Ограничение $(n - 1)$ введено для корректного расчета в острых граничных углах. В знаменателе используется $\pi/2$, так как выражение ориентировано на доминантно-четырёхугольные сетки.

Для оценки топологического качества сетки области или группы узлов вводится критерий CTV (cumulative TV) — среднее топологическое отклонение:

$$CTV = \frac{\sum_{i=1}^{nnodes} TV_i}{nnodes},$$

где n_{nodes} — количество узлов, качество которых оценивается. Мера нерегулярности узла (1) может применяться для внутренних и граничных узлов сетки. К узлам могут быть присоединены треугольные и четырехугольные ячейки. Мера нерегулярности промасштабирована таким образом, что $CTV = 1$ для 100 внутренних узлов четырехугольной сетки, один из которых имеет c -валентность 3, остальные — регулярные.

Значения TV для узлов с различной c -валентностью и отклонением c -валентности приведены в табл. 1. Из таблицы видно, что уменьшение валентности узла значительно хуже, чем увеличение. То есть внутренний узел с валентностью 3 топологически хуже, чем с валентностью 5.

При топологической оптимизации происходит поиск ячеек, узлов и ребер, над которыми может быть проведена топологическая операция. Первоначально проводится проверка ее эффективности. Для этого подсчитывается STV — сумма TV всех узлов, вовлеченных в операцию до и после ее проведения. Вычисляется изменение STV в результате проведения операции:

$$\Delta STV = STV_{after} - STV_{before}.$$

Топологическая операция проводится только в случае, если $\Delta STV < 0$.

Базовые топологические операции. Внутренний узел, присоединенный к двум ребрам, представляет топологически крайне нежелательную ситуацию, так как средний угол в ячейке при нем равен π . К такому узлу применяется операция уничтожения узла (рис. 9). Здесь и далее используются традиционные обозначения: узлы с избыточной c -валентностью отмечены квадратиками, узлы с недостаточной c -валентностью — кружками.

Операция открытия элемента показана на рис. 10. В результате операции добавляется новый узел и создается новая ячейка.

При проведении топологической операции закрытия элемента, изображенной на рис. 11, производится слияние диагонально-противоположных узлов. Если один из этих узлов граничный, то его координата не меняется (см. рис. 11, б).

Таблица 1

Мера нерегулярности узлов в зависимости от их c -валентности и отклонения от c -валентности

c -валентность внутренних узлов в четырехугольной сетке	Δn	TV
2	-2	900
3	-1	100
4	0	0
5	1	36
6	2	100

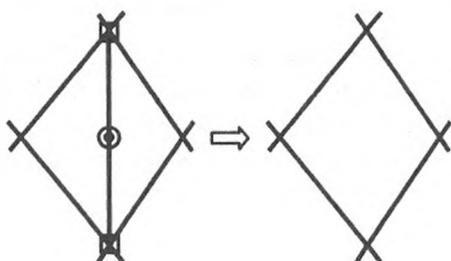


Рис. 9. Уничтожение узла ($\Delta STV = -972$)

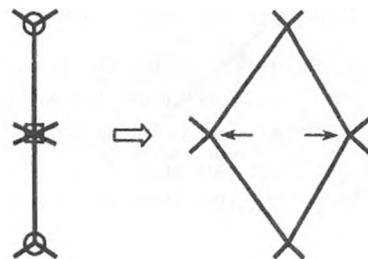


Рис. 10. Открытие элемента ($\Delta STV = -300$)

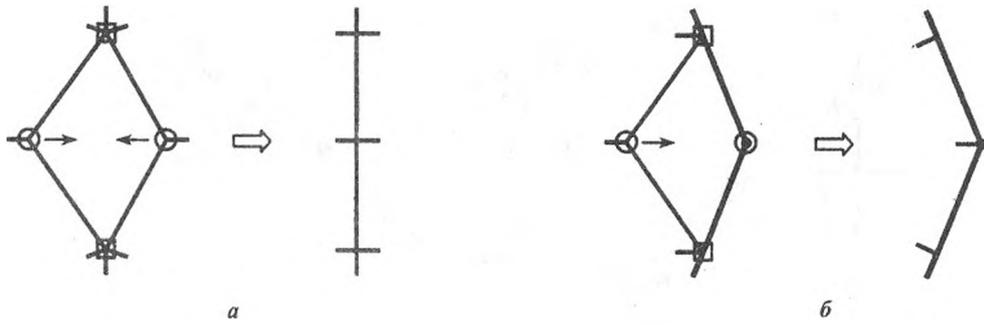


Рис. 11. Закрытие элемента: *a* — для внутренних узлов ($\Delta STV = -272$); *б* — у границы ($\Delta STV = -272$)

Топологические операции с контуром из шести ребер. Приведенные выше элементарные операции являются базовыми. Зачастую для получения оптимального результата требуется проведение комбинации элементарных операций для нескольких локальных узлов, причем некоторые из операций могут иметь $\Delta STV > 0$. Такой подход строится на распознавании топологических шаблонов, пригодных для проведения комбинаций операций. Например, в [11] приводится около тридцати таких комбинаций. Некоторые из них содержат до восьми элементарных операций.

Использование такого подхода дает хорошие результаты при проведении топологической оптимизации. Но практическая реализация отличается высокой сложностью. Поэтому при разработке данного алгоритма был реализован другой подход, базирующийся на топологических операциях с контурами из шести ребер (С6).

Используются четыре шаблона для идентификации С6 (рис. 12). Цель такой идентификации — выявить замкнутые совокупности из шести ребер независимо от того, сколько ячеек попадает внутрь контура, формируемого этими ребрами.

Во всех четырех шаблонах есть ячейка внутри С6, которая граничит со всеми остальными ячейками внутри С6. Это упрощает и ускоряет поиск всех возможных С6. Поиск С6 может идти в любом месте области ячеек, но для оптимизации этот поиск локализуется вблизи расположения нерегулярных узлов. Для каждого С6, идентифицированного в области интереса, делается проверка проведения восьми топологических операций. Три из них — это закрытие С6 (рис. 13, *a–в*), пять — пересоединение С6 (рис. 13, *г–з*).

При закрытии С6 все ячейки внутри него уничтожаются, а две пары узлов соединяются. В результате остаются три соединенных последовательно ребра. Закрытие С6 возможно по трем диагоналям шестиугольника: N_0N_3 (см. рис. 13, *a*), N_1N_4 (см. рис. 13, *б*), N_2N_5 (см. рис. 13, *в*). На операцию закрытия С6 накладывается дополнительное ограничение, связанное с размерной функцией: если характерный размер С6 больше, чем размер, соответствующий размерной функции, то операция не проводится. Это условие предназначено для того, чтобы закрывались достаточно маленькие ячейки. В противном случае после закрытия С6 размер соседних ячеек может увеличиться слишком сильно.

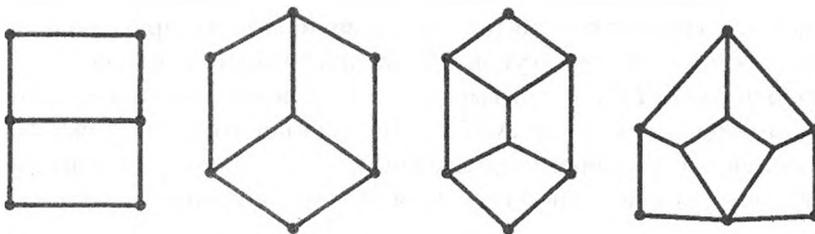


Рис. 12. Шаблоны формирования шестиреберного контура

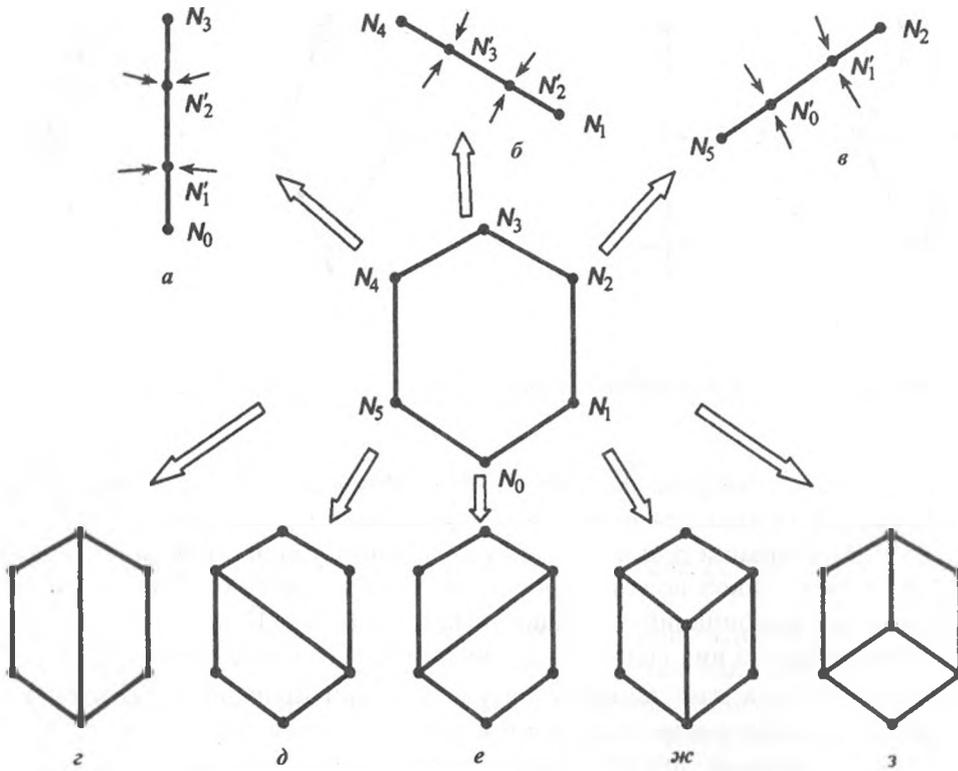


Рис. 13. Закрытие (а–в) и пересоединение (г–з) шестиреберного контура

При операции пересоединения С6 все ячейки внутри С6 уничтожаются, а вместо них создаются две или три новые ячейки. Дополнительных условий, связанных с размером ячеек, на эту операцию не накладывается.

При проверке возможности проведения операции с С6 проводится проверка для всех восьми вариантов и для каждого из них рассчитывается ΔSTV . Если есть варианты с $\Delta STV < 0$, то операция с минимальным ΔSTV считается оптимальной для данного С6. Пример выявления оптимальной операции для С6 представлен на рис. 14. Для С6, идентифицированного для трех ячеек, проверяются все восемь вариантов проведения топологической операции. До проведения операции $STV_{before} = 372$ для узлов на контуре и внутри него. Среди всех вариантов наилучшим является вариант *e*. Для него $STV_{after} = 100$ и соответственно $\Delta STV = -272$.

Порядок выполнения топологических операций. При проведении процедуры топологической оптимизации важным аспектом является алгоритм выбора наиболее подходящей топологической операции для сетки области. Работа алгоритма начинается с формирования набора ячеек, имеющих хотя бы один нерегулярный узел. Далее, в процессе топологической оптимизации этот набор "нерегулярных" ячеек поддерживается в соответствии с фактическим состоянием сетки. То есть производится его коррекция при возникновении и уничтожении нерегулярных узлов. Программа осуществляет проход по всем ячейкам этого набора и проверяет возможность проведения топологических операций для ячейки, ее ребер, узлов и контуров С6, которые можно построить вокруг нее. Операции, не имеющие отрицательного ΔSTV , не учитываются. Среди всех возможных операций выбирается та, которая имеет минимальное значение ΔSTV . Выбранная топологическая операция выполняется. Происходит обновление набора нерегулярных ячеек, и вся процедура повторяется снова. Алгоритм продолжает работу, пока можно выполнить хотя бы одну операцию с отрицательным ΔSTV .

Влияние топологической оптимизации на качество сетки. Пример сетки области до и после проведения топологической оптимизации приведен на рис. 15. Время, потраченное на топологическую оптимизацию, составляет $\sim 10-15\%$ от общего времени построения сетки. В процессе топо-

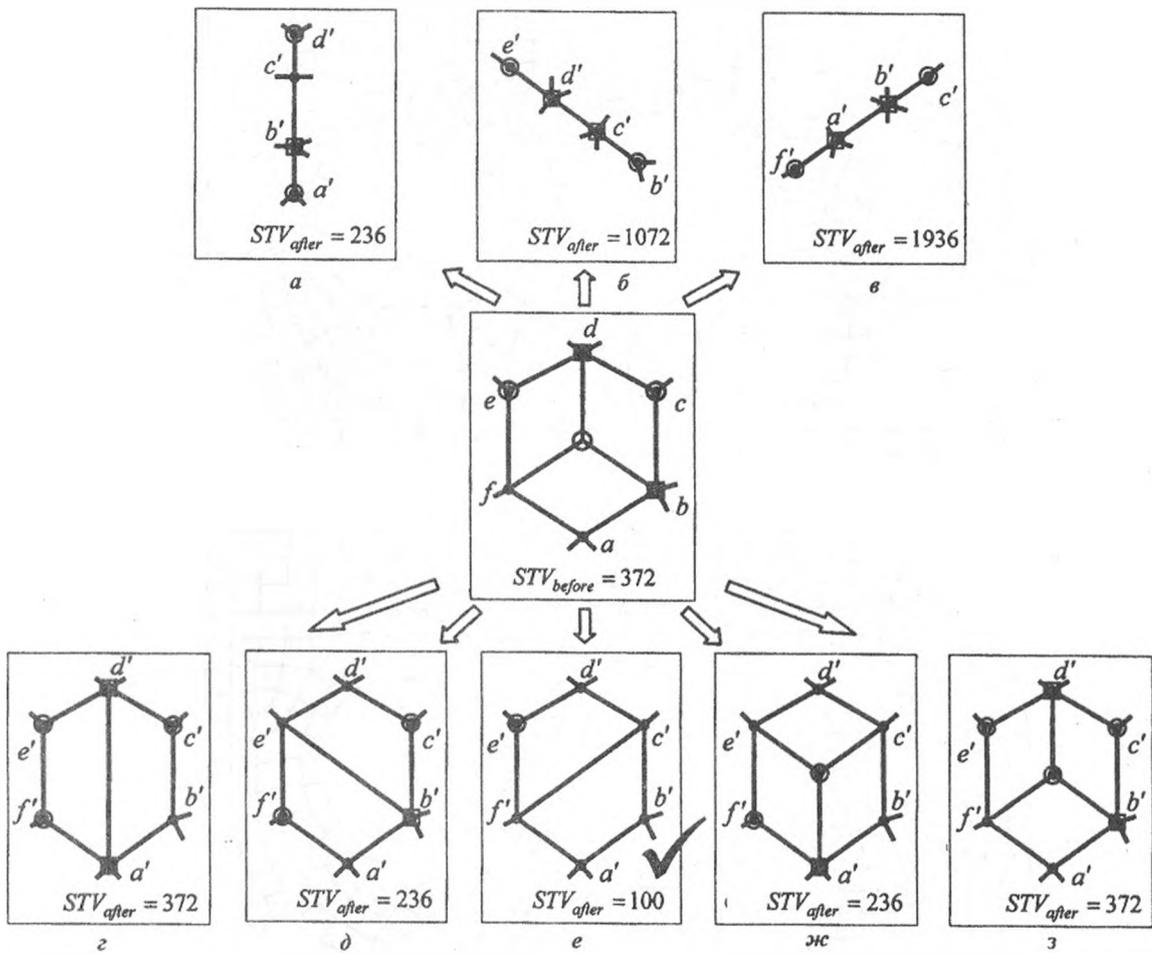


Рис. 14. Пример выбора топологической операции для С6

логической оптимизации было выполнено 9 операций закрытия элемента, 9 операций закрытия С6, 13 операций пересоединения С6.

Из примера видно, что топологическая оптимизация может существенно улучшить как топологическое, так и геометрическое качество сетки. Количественные топологические характеристики области приведены в табл. 2. Число нерегулярных узлов уменьшилось почти в 2 раза. Кроме того, есть тенденция к уменьшению общего количества ячеек и узлов в области.

Таблица 2

Влияние топологической оптимизации на сетку области

Параметр	До топологической оптимизации	После топологической оптимизации	Изменение, %
Кол-во узлов с валентностью 3	56	26	-54
Кол-во узлов с валентностью 5	58	35	-39
Кол-во узлов с валентностью 6	1	0	-100
CTV	11,908	7,135	-40
STV	7 788	4 438	-43
Количество ячеек в области	575	543	-5,6
Количество узлов в области	654	622	-4,9

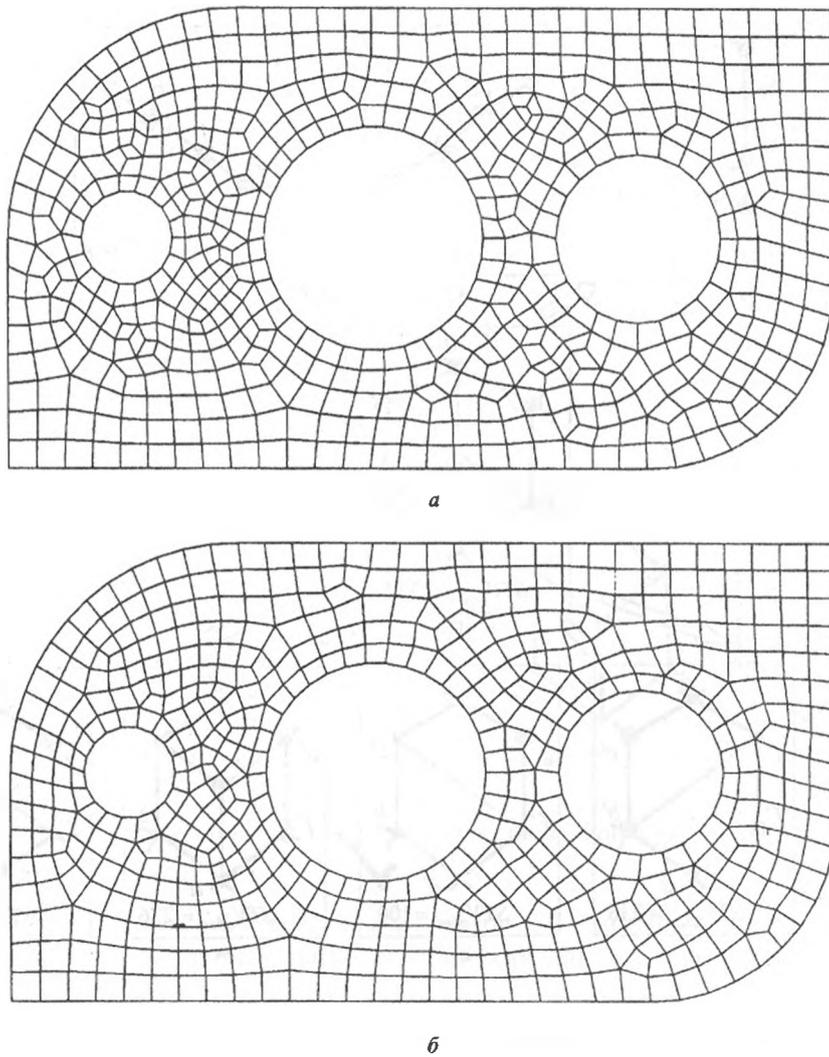


Рис. 15. Пример использования топологической оптимизации: *a* — до нее; *b* — после

2.7. Финальное сглаживание. После топологической оптимизации производится финальное сглаживание сетки области. Для этого используется ограниченное сглаживание по Лапласу, дополненное сглаживанием, основанным на оптимизации меры качества ячеек. Данное комбинированное сглаживание (Constrained Laplacian and Optimization – Based Smoothing) предложено в [10]. Оно имеет следующую схему. Сначала производится обычное сглаживание по Лапласу [12] или сглаживание по Лапласу с использованием весовых коэффициентов, равных расстояниям до узлов. Если сдвиг сглаживания приводит к ухудшению качества окружающих ячеек, то этот сдвиг запрещается. Процедура проверки на возможность проведения такого сдвига описывается совокупностью правил, которые учитывают степень изменения качества окружающих ячеек. Если сдвиг разрешен, то он производится, и на этом итерация заканчивается. В противном случае производится сглаживание, основанное на оптимизации меры качества ячеек. Смещение сглаживания определяется через градиенты мер качества всех окружающих узлов ячеек. В качестве оптимизируемого функционала для узла берется минимальная среди окружающих узлов ячеек мера качества. Цель сглаживания — найти новое положение узла, в котором функционал имеет максимальное значение.

Для треугольных ячеек в двумерном случае мера качества берется в виде

$$\alpha(ABC) = 2\sqrt{3} \left(\frac{\overrightarrow{BC} \times \overrightarrow{BA} \cdot \vec{n}}{\|AB\|^2 + \|BC\|^2 + \|CA\|^2} \right),$$

где \vec{n} — единичный вектор нормали треугольника. Эта мера была предложена Ло в 1985 [3] и 1988 гг. [4]. Для равностороннего треугольника $\alpha = 1$, для треугольника с нулевой площадью $\alpha = 0$, для вывернутого треугольника $\alpha < 0$.

Для четырехугольных ячеек мера качества берется в виде

$$\beta = \min(\alpha_1, \alpha_2, \alpha_3, \alpha_4) - \text{negval}, \quad (2)$$

где α_i — меры качества четырех треугольников, которые можно построить внутри четырехугольника, деля его диагоналями; *negval* — дополнительный "штрафной" параметр: *negval* = 1, если какой-либо угол ячейки меньше 6 градусов, или какие-либо его вершины лежат слишком близко, или четырехугольник вывернут (два значения α_i отрицательны); *negval* = 2, если три значения α_i отрицательны; *negval* = 3, если четыре значения α_i отрицательны.

Сглаживание по Лапласу с ограничением, дополненное сглаживанием, основанным на оптимизации качества ячеек, совмещает в себе "быстрое" сглаживание по Лапласу, которое хорошо "работает" на достаточно качественной сетке, и "медленное", но очень качественное сглаживание, основанное на оптимизации меры качества. Тестовые расчеты показывают, что для ~ 5–9% ячеек используется сглаживание с оптимизацией, для остальных достаточно сглаживания по Лапласу. Усиление "простого" сглаживания с помощью сглаживания, основанного на оптимизации, может использоваться не только со сглаживанием по Лапласу, но и с любым другим. Для простого сглаживания необходимо использовать критерий ограничения на сдвиг узла. Если оно не выполняется, то используется сглаживание, основанное на оптимизации.

3. Восстановление выпуклости ячеек

Для некоторых сложных граничных контуров построенная сетка может содержать невыпуклые четырехугольные ячейки. Этот эффект в основном проявляется, когда "противоположные" граничные ребра расположены слишком близко друг к другу. Кроме того, существуют такие конфигурации границы, для которых в принципе невозможно построить четырехугольную сетку, состоящую из выпуклых ячеек. Такие примеры приведены на рис. 16, а, в. На рис. 16, в в центре не четырехугольная ячейка, а внутренний граничный контур — из четырех ребер.

В предлагаемом методе есть возможность восстановить выпуклость всех ячеек. Для этого используются треугольные ячейки. Процедуры восстановления выпуклости представляют собой дополнительный блок, выполняемый после построения сетки. Использовать эту возможность или нет —

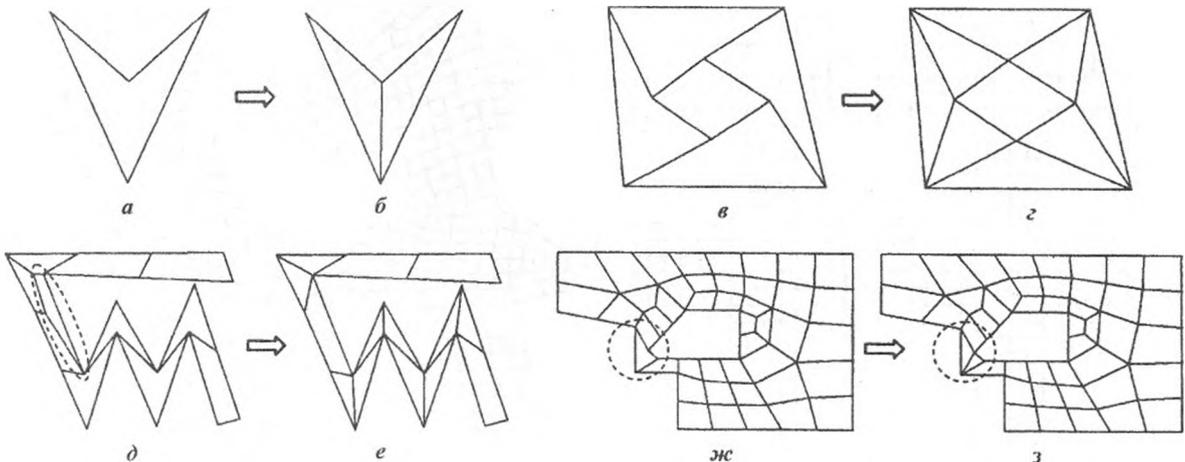


Рис. 16. Примеры использования блока восстановления выпуклости

решает пользователь, задавая параметры построения сетки. Для некоторых задач численного анализа выпуклость ячеек не является необходимым условием. Тогда эту дополнительную возможность можно отключить. Для большинства же методов твердотельного моделирования выпуклость ячеек является условием крайне желательным.

Блок восстановления выпуклости функционирует следующим образом. Первоначально все невыпуклые четырехугольные ячейки и их соседи разбиваются на треугольники. После этого производится топологическая оптимизация сетки. Топологические операции, описанные выше, дополняются операциями уничтожения треугольников. На все операции топологической оптимизации накладывается ограничение — операция запрещается, если в результате ее проведения может возникнуть невыпуклая ячейка. Таким образом, часть треугольников снова преобразуется в четырехугольники, а часть остается. Необходимо отметить, что на практике блок восстановления выпуклости подключается редко, но тем не менее он гарантирует получение сетки с выпуклыми ячейками при любых граничных условиях. Пример использования этого блока представлен на рис. 16. На рис. 16, *ж, з* область восстановления выпуклости обведена пунктирной линией.

Критерием использования восстановления выпуклости ячеек может служить не только невыпуклость ячейки. На практике целесообразно использовать меру качества ячеек (2), чтобы определять ячейки неприемлемого качества. Такой критерий был использован для ячейки на рис. 16, *д*. Эта ячейка обведена пунктирной линией. Хотя она и является выпуклой, но качество ее очень низкое. Поэтому она была включена в набор "плохих" ячеек блоком восстановления выпуклости.

4. Примеры построения сетки

Программа на основе алгоритма QMV тестировалась на различных начальных граничных контурах. Примеры построения сетки для некоторых из них приводятся ниже.

На рис. 17, *а* приводится пример с ортогональными границами. Сетка строится оптимально, причем координаты внутренних узлов ячеек строго совпадают с соответствующими координатами граничных узлов. На рис. 17, *б* приведена сетка для неортогонального контура.

На рис. 18, *а* приведено построение сетки для контура с топологией кольца. Количество узлов на внешней и внутренней границах взято одинаковым. Сетка для таких начальных условий хорошо строится регулярными методами. Предлагаемая программа также строит оптимальную сетку на таких контурах. На рис. 18, *б, в* приведен пример разбиения контуров с различным количеством узлов на внешней и внутренней границах. На рис. 18, *б* количество узлов отличается на 1, на рис. 18, *в* — на 2. Для таких контуров построение регулярной сетки уже невозможно, так как противоположные

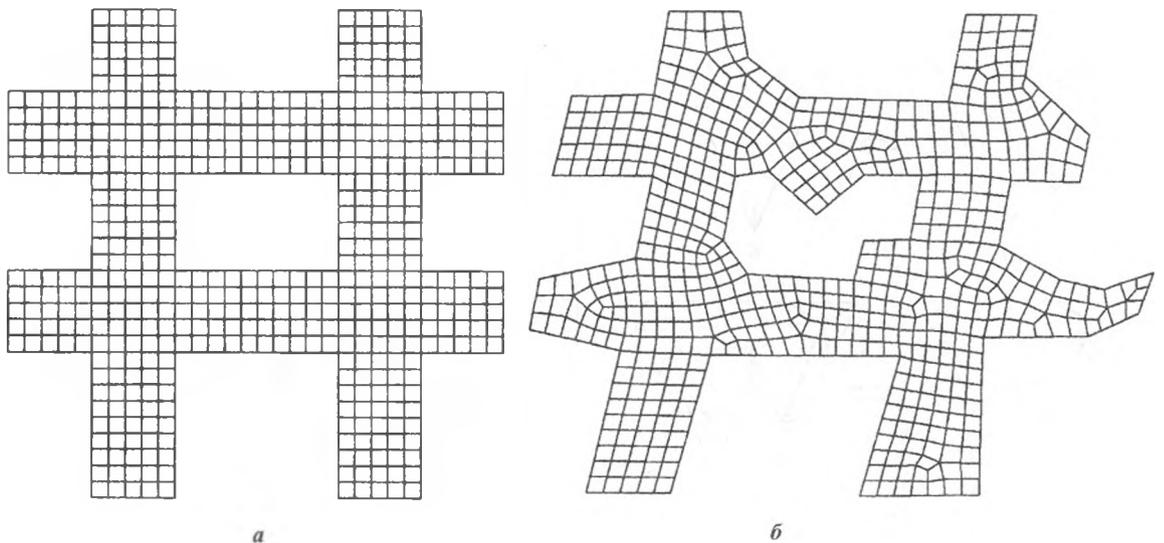


Рис. 17. Пример сетки для ортогональной (*а*) и неортогональной (*б*) границ

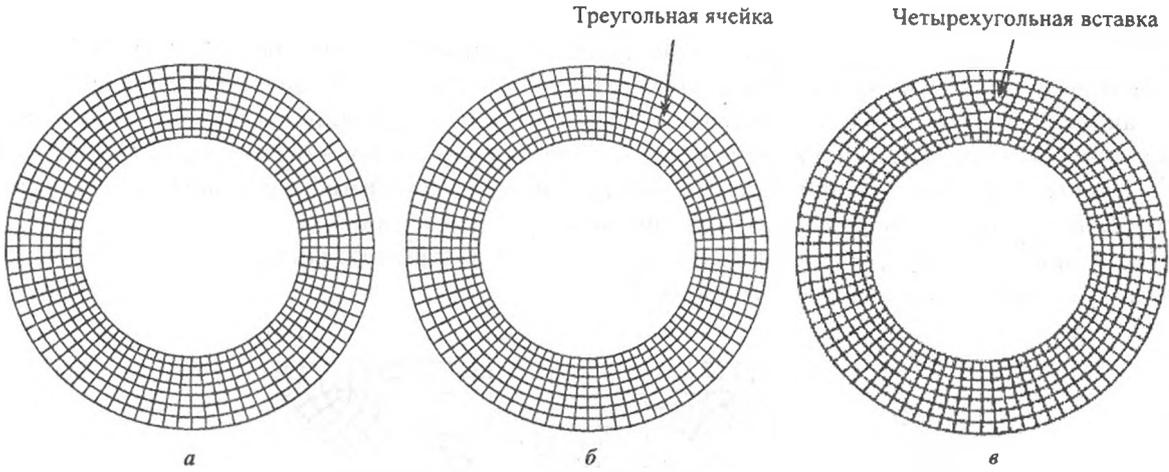


Рис. 18. Построение сетки для контура с топологией кольца: *a* — с одинаковым количеством узлов на внешней и внутренней границах; *б, в* — с разным количеством узлов на границах

стороны топологически неэквивалентны. Граница на рис. 18, *б* имеет нечетное количество ребер, следовательно, сетка должна содержать, как минимум, один треугольник. На рис. 18, *в* граница содержит четное число ребер. Построенная сетка полностью четырехугольная. Одна четырехугольная ячейка использована в качестве вставки.

На рис. 19 показано построение сетки для круга и кольца с граничными ребрами равной длины на внешней и внутренней границах.

На рис. 20 показано построение сетки для областей, топологически сложных для регулярного или блочно-регулярного разбиения. Например, построить регулярную сетку хорошего качества в треугольнике невозможно. Построение нерегулярной сетки в треугольнике приведено на рис. 20, *б*. Аналогичная ситуация для контуров с сужением — на рис. 20, *г*. Регулярная сетка для такой области имела бы значительно различающиеся по размеру ячейки. Во многих задачах механики сплошной среды на подвижных сетках сильно деформированные области зачастую имеют сложный контур с сильно искривленной границей. Пример построения сетки для такого произвольного контура показан на рис. 20, *в*.

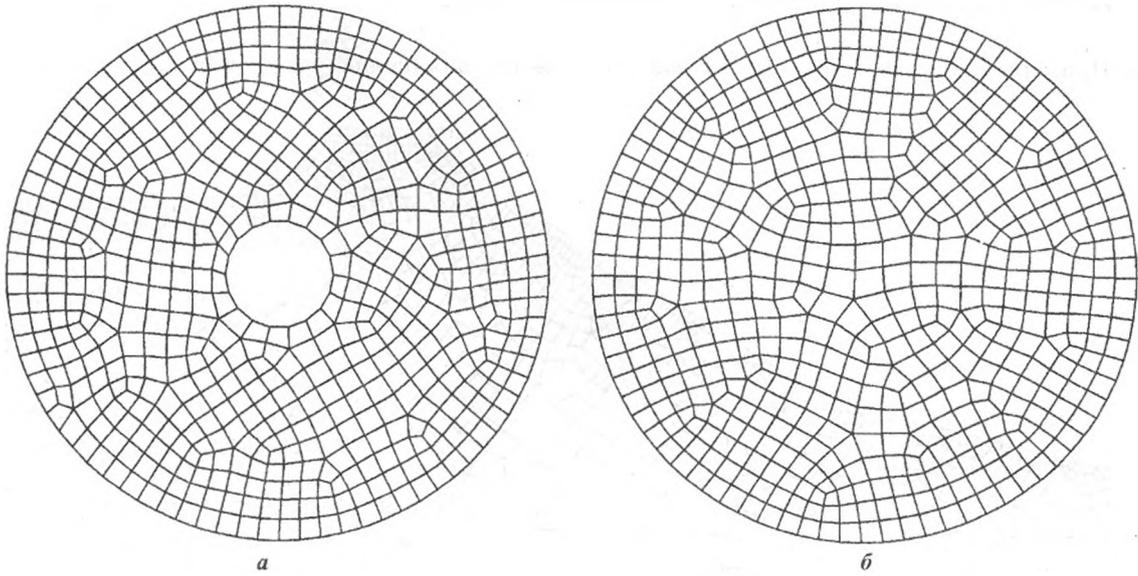


Рис. 19. Построение сетки для кольца (*a*) и круга (*б*)

Для некоторых задач численного моделирования используется сетка с большим градиентом размерной функции, т. е. длины граничных ребер различаются значительно (*large transitions*). Например, такое сеточное разбиение может использоваться для более точного расчета напряжений на границах (рис. 21). Границы с *large transitions* обычно используются как тестовые. Они позволяют выявить сильные и слабые стороны алгоритма при работе с "плохими" начальными условиями.

На рис. 22 показано построение сетки для произвольного контура.

На рис. 23 приведено промежуточное и окончательное построение сетки для сложного контура, обладающего симметрией при повороте на $2\pi/5$.

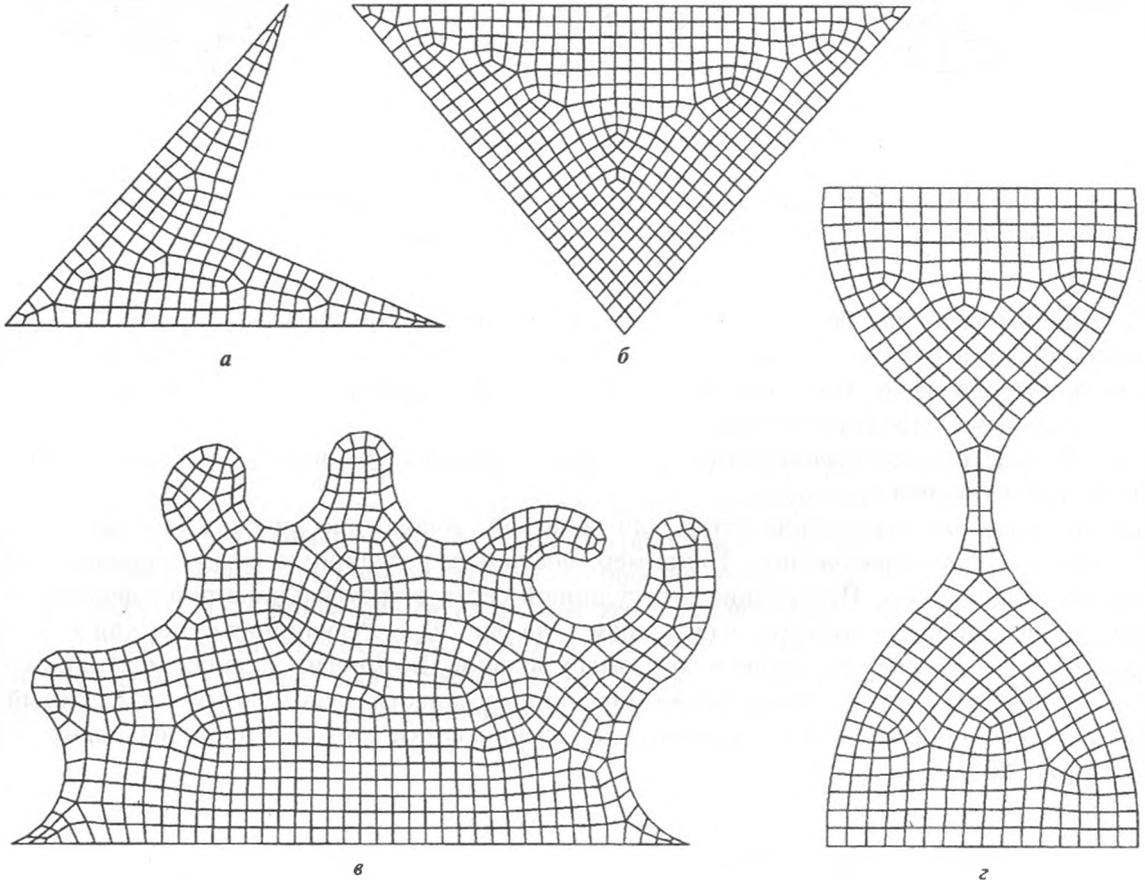


Рис. 20. Примеры построения сетки для областей, сложных для блочно-регулярного разбиения

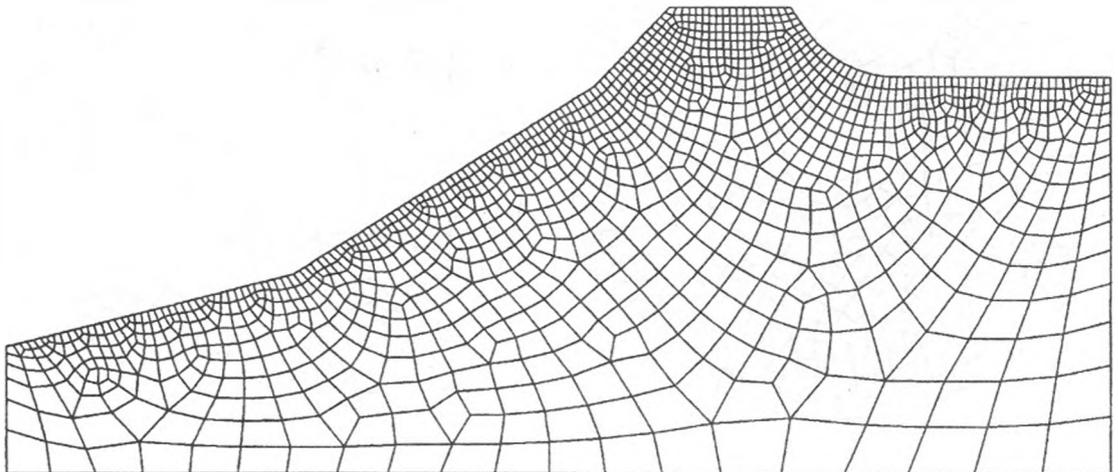


Рис. 21. Пример построения сетки для контура с сильно различающимися длинами ребер

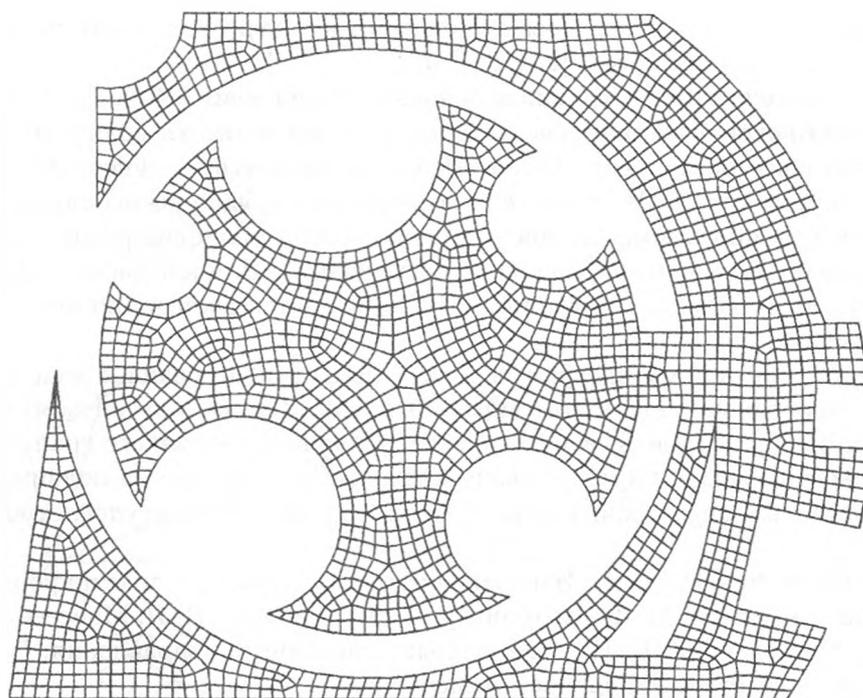


Рис. 22. Построение сетки для произвольного контура

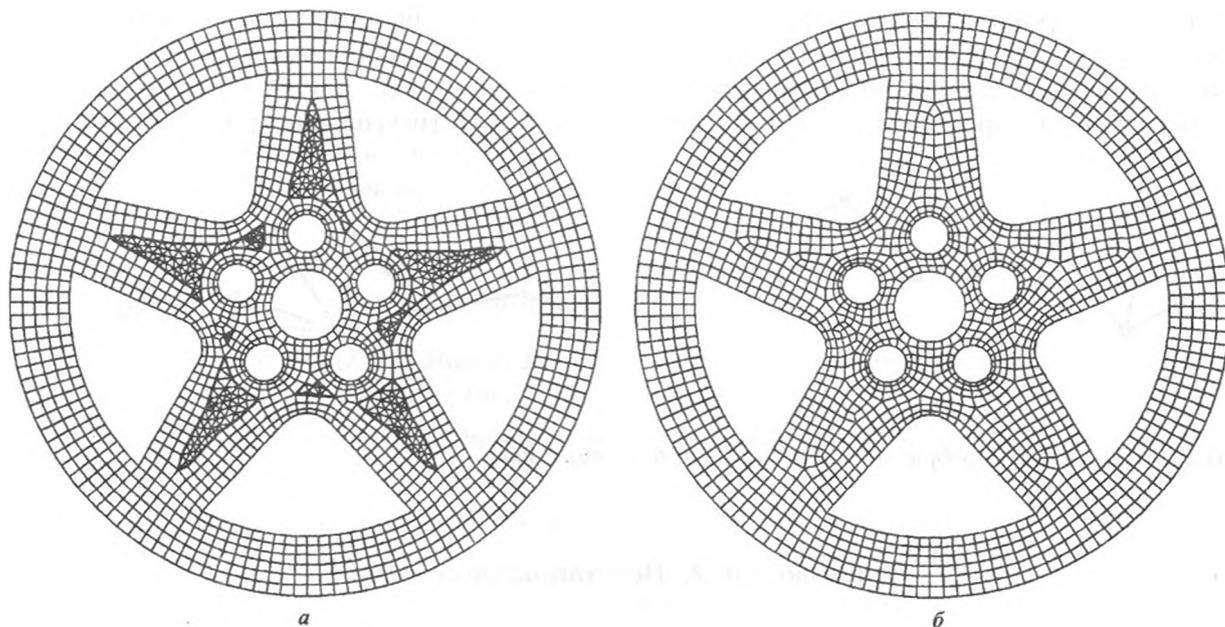


Рис. 23. Промежуточное (а) и окончательное (б) построение сетки для сложного контура

Заключение

Разработанный алгоритм QMV относится к непрямым фронтальным методам построения четырехугольных нерегулярных сеток. В нем используется предварительная триангуляция области и последующая трансформация треугольников в четырехугольные ячейки. Исходными данными для алгоритма являются координаты узлов границы. Алгоритм является полностью автоматическим и не модифицирует граничных узлов. В качестве базового взят метод Q-Morph, предложенный С. Оуэном в 1998 г. [8].

При создании алгоритма были использованы технологии построения сетки, сглаживания и топологической оптимизации, изложенные в [2, 11, 10, 7, 9].

В алгоритме QMV используются следующие основные дополнения к базовому алгоритму: второй этап блока построения новой ячейки, технология сохранения четности количества ребер фронтов, технология закрытия контура из шести ребер. Использование этих дополнений повышает надежность и устойчивость алгоритма. В процессе тестирования не было выявлено ситуаций с нештатным завершением работы программы. В той или иной степени модифицированы процедура выбора активного ребра фронта, локальное и финальное сглаживание. Для алгоритма QMV была разработана технология топологической оптимизации, которая является достаточно простой в реализации, надежной и эффективной.

Созданная на основе алгоритма программа может быть использована в качестве независимого сеточного модуля в программах численного моделирования. Так как алгоритм является полностью автоматическим и позволяет строить сетку для любого неинвертированного граничного контура, то появляется возможность создать геометрически безавостную методику на подвижных сетках. Для этого может использоваться перестройка сетки в локальных областях с неудовлетворительной мерой качества ячеек.

Автор выражает благодарность С. Н. Барабанову — за помощь в анализе информации на начальном этапе работы над проектом; М. Ю. Козманову, А. Д. Гаджиеву, Н. Я. Моисееву — за поддержку и содействие; В. А. Гаранже (ВЦ РАН) — за ценные замечания и исправления; В. Н. Ананийчуку, И. И. Карлыхановой — за информационное обеспечение.

Приложение 1. Переключение ребра

Переключение ребра (edge swapping) — топологическая операция, которая может быть выполнена для ребер, принадлежащих двум треугольникам. Ребро уничтожается, и вместо него создается другое, соединяющее вершины, противолежащие удаленному ребру (рис. П.1, а). Операция запрещена, если в результате ее проведения может появиться вывернутый треугольник (рис. П.1, б).

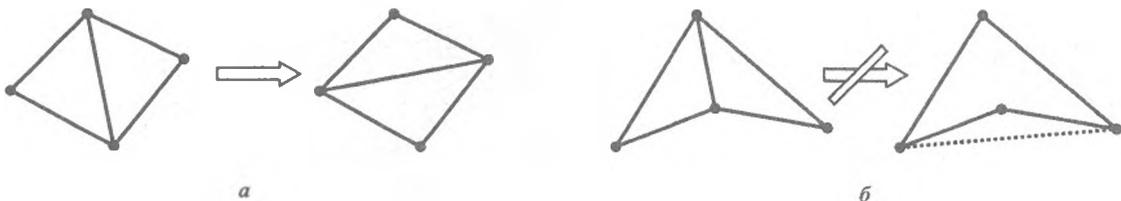


Рис. П.1. Переключение ребра: а — выполняется; б — запрещено

Приложение 2. Восстановление ребра

Восстановление ребра (edge recovery) — это технология, позволяющая создать ребро, соединяющее два произвольных узла. Она может быть использована, если отрезок, соединяющий эти два узла, пересекает только треугольные ячейки. Алгоритм восстановления ребра основан на переключении ребер, пересекающих этот отрезок. Впервые технология была предложена для восстановления границ в триангуляции Делоне [13]. Пример восстановления ребра E_R между узлами N_A и N_B показан на рис. П.2. В результате четырех операций переключения ребер восстанавливается ребро E_R .

Алгоритм восстановления ребра:

- 1) пусть S — отрезок, соединяющий узлы N_A и N_B ;
- 2) пусть $\Lambda(S)$ — список ребер, которые пересекают S . Ребра в списке представлены в том порядке, в котором они пересекаются с S от N_A к N_B ;
- 3) FOR EACH $E_i \in \Lambda(S)$

- 4) если можно сделать переключение ребра E_i , то
- 5) делаем переключение ребра E_i ;
- 6) пусть E_{iS} — ребро, образованное в результате переключения;
- 7) если E_{iS} пересекается с S , то помещаем E_{iS} в конец списка $\Lambda(S)$,
- 8) иначе помещаем E_i в конец списка $\Lambda(S)$;
- 9) если список $\Lambda(S)$ пуст, то выход;
- 10) NEXT E_i в списке $\Lambda(S)$.

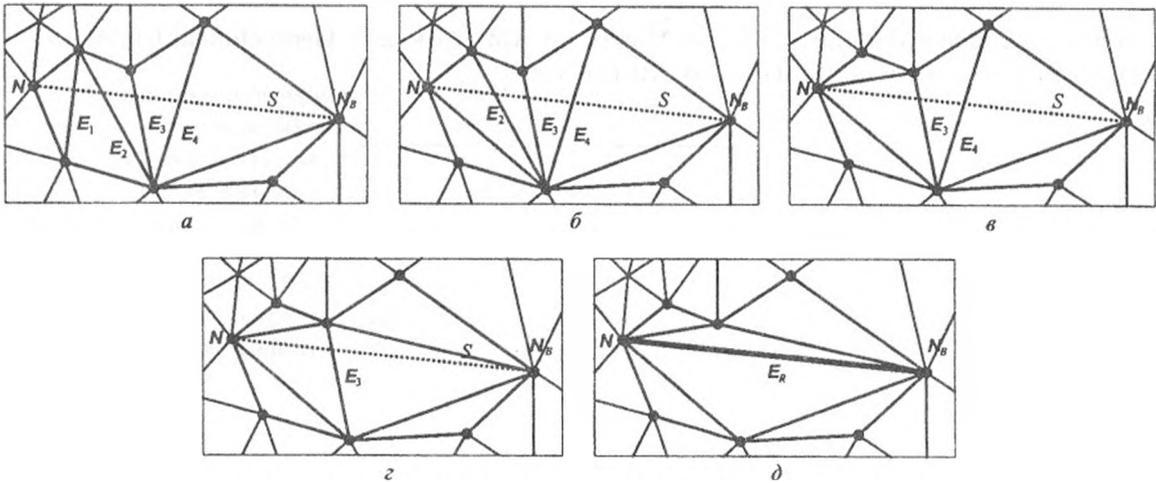


Рис. П.2. Восстановление ребра: а — начальная триангуляция; б — переключение ребра E_1 ; в — переключение ребра E_2 ; г — переключение ребра E_4 ; д — ребро E_R восстановлено

Список литературы

1. Zhu J. Z., Zienkiewicz O. C., Hinton E., Wu J. A new approach to the development of automatic quadrilateral mesh generation // Int. J. Numer. Meth. Eng. 1991. Vol. 32. P. 849–866.
2. Blacker T. D., Stephenson M. B. Paving: a new approach to automated quadrilateral mesh generation // Ibid. P. 811–847.
3. Lo S. H. A new mesh generation scheme for arbitrary planar domains // Ibid. 1985. Vol. 21. P. 1403–1426.
4. Lo S. H. Generating quadrilateral elements on plane and over curved surfaces // Comput. Struct. 1989. Vol. 31. P. 421–426.
5. Johnston B. P., Sullivan J. M. Jr., Kwasnik A. Automatic conversion of triangular finite element meshes to quadrilateral elements // Int. J. Numer. Meth. Eng. 1991. Vol. 31. P. 67–84.
6. Borouchaki H., Frey P. Adaptive triangular-quadrilateral mesh generation // Ibid. 1998. Vol. 41. P. 915–934.
7. Lee C. K., Lo S. H. A new scheme for the generation of a graded quadrilateral mesh // Comput. Struct. 1994. Vol. 52. P. 847–857.
8. Owen S. J., Staten M. L., Canann S. A., Saigal S. Advancing front quadrilateral meshing using triangle transformations // 7th Int. Meshing Roundtable. Dearborn, Michigan, Sandia National Labs. October 26–28, 1998. P. 409–428.

9. *Marcum D. L., Weatherill N. P.* Unstructured grid generation using iterative point insertion and local reconnection // AIAA J. 1995. Vol. 33, №9. P. 1619–1625.
 10. *Canann S. A., Tristano J. R., Staten M. L.* An approach to combined Laplacian and optimization-based smoothing for triangular, quadrilateral and quad-dominant meshes // 7th Int. Meshing Roundtable. Dearborn, Michigan, Sandia National Labs. October 26–28, 1998. P. 479–494.
 11. *Canann S. A., Muthukrishnan S. N., Phillips R. K.* Topological Improvement Procedures for Quadrilateral Finite Element Meshes. Costa Mesa: The MacNeal-Schwendler Corporation / PDA Engineering Technology Center, 1994.
 12. *Field D. A.* Laplacian smoothing and Delaunay triangulations // Commun. Appl. Numer. Meth. 1988. Vol. 4. P. 709.
 13. *Jones N. L.* Solid Modelling of Earth Masses for Applications in Geotechnical Engineering. Doctoral Dissertation. University of Texas at Austin, 1990.
-