

УДК 519.6

ГЕНЕРАТОР ПОВЕРХНОСТНЫХ НЕСТРУКТУРИРОВАННЫХ ЧЕТЫРЕХУГОЛЬНЫХ СЕТОК В ПРЕПРОЦЕССОРЕ ПАКЕТА ПРОГРАММ "ЛОГОС"

Е. Ю. Арапова, В. Г. Куделькин, Е. А. Павлов, С. Ю. Полякова, А. В. Тихонов
(ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области)

Приводится описание автоматического генератора поверхностных неструктурированных четырехугольных сеток для геометрических моделей в параметрическом представлении. Генератор предназначен для построения сеточных моделей при решении задач прочности в рамках пакета программ "Логос". Рассматриваются этапы генерации поверхностной сетки. Подробно рассмотрен один из этапов, посвященный генерации четырехугольных сеток в параметрической плоскости. Приведены примеры тестовых задач, в которых используются различные геометрические модели.

Ключевые слова: пакет программ "Логос", генератор поверхностных четырехугольных сеток, подвижный фронт, формирование четырехугольных ячеек, локальное сглаживание, топологическая оптимизация.

Введение

В современной научно-исследовательской деятельности большое внимание уделяется развитию методов численного моделирования физических процессов в областях сложной геометрии. При этом наиболее подходящим для численного решения задач прочности является *метод конечных элементов* (МКЭ) [1]. Обусловленность систем линейных алгебраических уравнений, возникающих при реализации МКЭ, тем лучше, чем больше минимальные углы между гранями (ребрами) конечных элементов. В частности, точность расчетов на сетках из треугольных элементов (ячеек) существенно ниже, чем на четырехугольных сетках.

Данная работа преследует две цели: 1) представить генератор неструктурированных четырехугольных сеток, входящий в состав пакета программ (ПП) "Логос" [2] и предназначенный для построения поверхностных сеток при подготовке задач в модуле "Логос Прочность"; 2) дать описание модифицированных алгоритмов разработанного генератора, которые позволили повысить качество формируемой сетки и надежность алгоритма.

Методы построения сеток делятся на два класса: прямые и не прямые (итерационные) [3]. Из-

за своей универсальности, в частности способности строить сетку в областях достаточно произвольной формы, не прямые методы получили наибольшее распространение.

Одним из наиболее эффективных по ряду критериев является не прямой метод подвижного фронта Q-Morph [3], предложенный Стивеном Оуэном в 1998 г. Сетка в нем строится столь же качественная, как и в лучших прямых методах. Используемые трансформации треугольных элементов позволяют контролировать размеры генерируемых элементов, анализировать границы области, а также делают его надежным и быстрым.

Существуют варианты метода подвижного фронта, которые напрямую строят сетки из конечных элементов непосредственно на поверхности. Как утверждают авторы статьи [4], к недостаткам этих методов можно отнести сложность реализации, низкую скорость построения и неустойчивость. Ими же в [4] был предложен метод построения поверхностной треугольной сетки с использованием генерации сетки на плоскости с учетом римановой метрики поверхности для контроля размера и формы треугольников в параметрическом пространстве. Получены удовлетворительные результаты, и данный генера-

тор включен в состав коммерческого программного продукта ANSYS [5]. Авторами настоящей работы было принято решение повторить опыт использования плоского генератора для построения поверхностной сетки [4], но с его обобщением на четырехугольные элементы.

Программной реализации метода Q-Morph посвящено большое количество публикаций. Авторами известных российских программ являются А. В. Сковпень (РФЯЦ-ВНИИТФ, г. Снежинск) [6] и В. И. Лысухин (ОАО "Туполев", г. Москва) [7]. В процессе разработки ими были модифицированы некоторые имеющиеся процедуры метода и добавлены новые. Кроме того, ими (каждым отдельно) была разработана технология топологической оптимизации отстраиваемой сетки. В настоящей работе в качестве базового алгоритма был взят алгоритм, предложенный В. И. Лысухиным в [7].

На основе анализа данных, полученных при построении четырехугольных сеток на достаточном количестве геометрических моделей (произвольной формы и с различным разбиением на граничных контурах), авторы сделали вывод о необходимости доработки реализуемого алгоритма, так как многие моменты, связанные с обработкой различных геометрических и топологических особенностей, в описании либо отсутствовали, либо были преподнесены достаточно поверхностно. Основным изменениям были подвергнуты процедуры обработки фронтов и построения нового четырехугольного элемента.

В процессе формирования четырехугольников стартовая треугольная сетка подвергается сильным трансформациям. Чем дальше от границы области, тем более неравномерными становятся размеры треугольников. Также могут появиться так называемые вывернутые треугольники и треугольники с нулевой площадью. Поэтому после успешного построения каждого четырехугольного элемента выполняется процедура локального сглаживания. За основу взят метод, описанный в работе Т. Д. Блэккера и М. Б. Стефенсона [8]. Тестовые расчеты выявили ряд проблем и недостатков в используемом алгоритме многоэтапного локального сглаживания, которые, в свою очередь, потребовали изменения некоторых этапов алгоритма. Дополнительно используется сглаживание либо по Лапласу, либо по Лапласу с ограничениями [9].

Наконец, было принято решение отказаться от разработанного и предложенного В. И. Лысухиным [7] метода топологической оптимизации.

Топологическая оптимизация — это совокупность операций, предназначенных для улучшения качества сетки. Для четырехугольной сетки считается, что внутренний узел обладает оптимальной топологией, если он принадлежит четырем ячейкам (условие регулярности). В базовой версии решение о проведении той или иной процедуры принимается на основании анализа соседних ячеек. Практическое использование показало низкую скорость выполнения программы, обусловленную многократно повторяющейся процедурой определения соседей для каждого четырехугольника и дублированием операций определения условий нерегулярности вокруг узла, потому что каждый узел принадлежит сразу нескольким четырехугольникам. Реализованная версия программы топологической оптимизации использует результаты работ [10–12].

1. Технология построения неструктурированных четырехугольных поверхностных сеток в препроцессоре ПП "Логос"

Построение сетки на геометрической модели в параметрическом представлении состоит из нескольких этапов:

1. Для построения сетки на поверхности пользователю необходимо задать разбиение ее границы на набор ребер с желаемым размером. Для замкнутых поверхностей, не имеющих границ, типа сферы или поверхностей сложной формы, например с изломами, стандартной считается практика, когда пользователь перед построением сетки упрощает геометрию, разбивая деталь на части с более простой топологией. Программный продукт "Логос" обладает большим набором средств для выполнения подобных операций [13].
2. Геометрическая модель отображается в параметрическую плоскость R^2 . В препроцессоре ПП "Логос" используется функция отображения, реализованная в коде библиотеки геометрического ядра C3D [14].
3. В плоскости R^2 строится треугольная сетка. Построение осуществляется генератором треугольных сеток, реализованным в библиотеке CM2 MeshTools [15].
4. В плоскости R^2 строится четырехугольная сетка методом подвижного фронта с учетом внутренней геометрии поверхности при вы-

числении размера элемента сетки и углов между ребрами.

- Обратным преобразованием построенная в R^2 сетка отображается в пространство R^3 . В результате исходная геометрическая модель покрывается поверхностной неструктурированной сеткой четырехугольных ячеек.

Этапы 2–5 выполняются автоматически, без вмешательства пользователя. На этапах 1–3 и 5 использовались уже реализованные в ПП "Логос" программные средства. Далее рассмотрим более детально алгоритмы, реализованные авторами данной статьи для этапа 4.

2. Общая схема алгоритма построения неструктурированных четырехугольных сеток

Как было сказано во Введении, в качестве базового используется алгоритм, предложенный в работе [7]. В целом он повторяет алгоритм Q-Morph и состоит из следующих основных этапов:

- 1. Определение подвижного фронта.** Фронтом называется замкнутая совокупность ребер, на которых будут формироваться новые четырехугольные элементы. Если область построения многосвязная, имеем несколько независимых фронтов. Естественно, в самом начале к фронтам относятся ребра, принадлежащие границам области. Фронты обрабатываются последовательно. Формирование четырехугольников начинается с фронта, описывающего внешнюю границу области. На всех ребрах выбранного фронта выполняется процедура построения четырехугольной ячейки. Обход ребер фронта осуществляется против часовой стрелки (область, подлежащая разбиению, всегда слева

от фронта). После этого выполняется переход к обработке следующего фронта. По мере построения четырехугольников фронт меняется (движется) и представляет собой границу, отделяющую уже построенные четырехугольные ячейки от области, содержащей треугольные ячейки. Фронты могут делиться и объединяться в процессе построения сетки. В конце, после смыкания всех фронтов, поверхность оказывается полностью заполненной четырехугольными элементами. Заметим, что получаемая сетка гарантированно будет полностью четырехугольной, если фронт содержит четное число ребер (условие четности фронта).

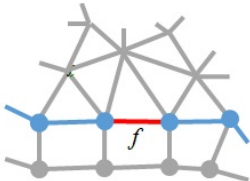
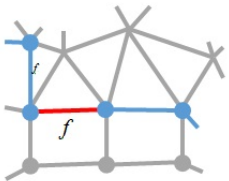
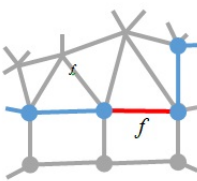
- 2. Классификация ребер фронта.** Будем считать, что общий узел двух соседних ребер фронта находится в состоянии 0 (состоянии 1), если угол между этими ребрами больше $3\pi/4$ (меньше $3\pi/4$). В свою очередь, в процессе создания четырехугольных элементов каждое ребро f фронта принимает одно из трех состояний (табл. 1) в зависимости от значений состояний своих узлов.

На начальном этапе построения четырехугольной сетки в процессе обработки фронта, описывающего внешнюю границу области, определяется *стартовое* ребро. Для его определения используются два признака: минимальная длина ребра и минимальный угол между ребрами.

- 3. Построение новой ячейки на ребре фронта.** Ребро фронта, на котором строится новая четырехугольная ячейка, называется *активным*. Для построения используются различные процедуры, основанные на классификации ребер фронта. Если новая ячейка сформирована, выполняется локальное сглаживание. После построения ячейки

Таблица 1

Характерные состояния ребер фронта

0 – 0	1 – 0 (0 – 1)	1 – 1
		

активное ребро фронта меняется — им становится следующее ребро текущего фронта.

4. *Проверка наличия ребер подвижного фронта.* Если существуют ребра подвижного фронта, выполняется переход на этап 2.
5. *Топологическая оптимизация (регуляризация сетки).* Данная процедура направлена на оптимизацию связности сетки.
6. *Финальное сглаживание.* Выполняется сглаживание всех ячеек области построения.

Далее рассмотрим подробнее изменения, внесенные авторами в базовый алгоритм из [7]. При этом обращаем внимание на тот факт, что в работе не преследуется цель описать все изменения. Остановимся только на тех процедурах, модификация которых позволила получить сетку лучшего качества при более стабильном выполнении алгоритмов.

3. Модификации базового алгоритма

Выбор направления вектора нормали. В процессе построения ячейки на активном ребре фронта из начальной треугольной сетки требуется выбрать ребра, которые будут являться боковыми сторонами формируемого четырехугольника. Подходящим считается ребро, инцидентное узлу активного ребра фронта (от которого строится боковая сторона) и имеющее наименьшее отклонение от нормали V_n (диапазон отклонения $\pm\pi/6$). Классически (см. [3, 6, 7]) нормаль в узле определяется как биссектриса угла, образованного соседними ребрами фронта. На рис. 1 показано определение направления нормального вектора, исходящего из узла B , где $\varphi = \angle(AB, BC) = \angle(AB, BC)$; AB — активное ребро фронта.

На рис. 2 приведен пример, когда применение классического подхода к определению вектора

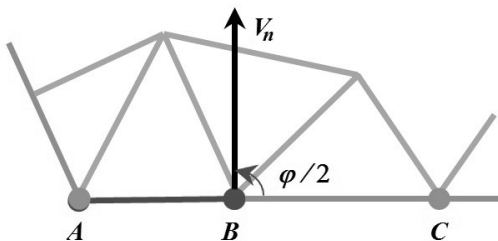


Рис. 1. Классическое определение направления нормали V_n между активным ребром AB и ребром BC

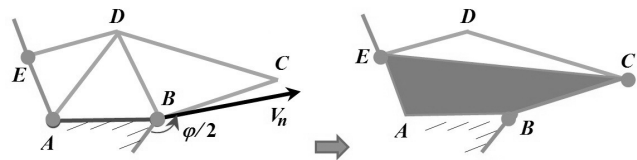


Рис. 2. Выбор ребра BC в качестве бокового ребра четырехугольника $ABCE$ с использованием классического подхода к определению вектора нормали V_n

нормали приводит к образованию четырехугольной ячейки низкого качества.

Для улучшения качества строящейся сетки был разработан и реализован следующий критерий выбора угла β между активным ребром AB и вектором нормали V_n :

$$\beta = \begin{cases} \frac{1}{4}\varphi, & \text{если } \varphi \geq 1,7\pi; \\ \frac{1}{3}\varphi, & \text{если } 1,2\pi \leq \varphi < 1,7\pi; \\ \frac{1}{2}\varphi, & \text{если } \varphi < 1,2\pi. \end{cases}$$

На рис. 3 схематично показан результат применения модифицированного алгоритма для определения направления вектора нормали V_n с последующим образованием ячейки $EABD$ удовлетворительного качества.

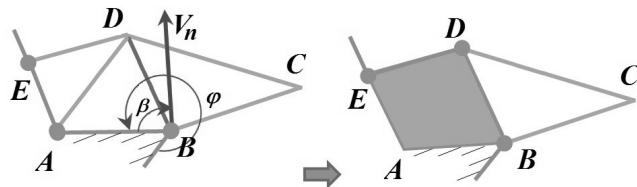


Рис. 3. Формирование четырехугольной ячейки по модифицированному алгоритму

Обработка узких фронтов. Будем называть фронт *узким*, если противолежащие ребра фронта разделены малым количеством треугольников (1–3 слоя). Признаком узкого фронта является то, что построенная из узла активного ребра нормаль V_n пересекает второе, третье или четвертое ребро фронта, считая от активного при обходе фронта по часовой стрелке. При этом из-за проведенных ранее трансформаций треугольники сильно деформированы по сравнению с первоначальной формой: площади треугольников существенно меньше первоначальных или отсутствуют треугольники, для

которых отношение площади к периметру стремится к нулю. Как правило, такая ситуация может возникнуть перед закрытием фронта, когда все входящие во фронт ребра находятся в состояниях 0-0 или 1-0 (0-1).

На рис. 4 схематично показано построение четырехугольника в узком фронте по алгоритму из работы [7]. В левой части рисунка видно, что образованный четырехугольник имеет малый размер (длина его ребер меньше характерной длины ребер фронта). Последующее применение операции сглаживания улучшит ситуацию, но треугольная сетка при этом сильно деформируется, что в дальнейшем скажется на качестве вновь формируемых ячеек. В правой части рисунка сформирован четырехугольник хорошего качества, но треугольная сетка сгущается. Также возможно заикливание по ребрам фронта (на каждом активном ребре фронта происходит отказ от построения по какой-либо причине).

Чтобы избежать подобных ситуаций, для обработки узкого фронта, когда активные ребра находятся в состояниях 0-0 или 1-0 (0-1), разработан другой алгоритм. Рассматривается узел активного ребра фронта, соответствующий состоянию 0 (для ребра фронта 0-0 берется левый узел). На рис. 5 это узел A . Далее определяется ребро фронта, которое пересекается нормалью, выходящей из рассматриваемого узла. Интересно его номер по отношению к активному ребру при прохождении фронта по часовой стрелке. Как правило, это третье или четвертое ребро фронта от активного (см. рис. 5). В других случаях четырехугольник либо образуется исполь-

зованием базового алгоритма [7], либо происходит отказ от его формирования на текущем ребре фронта.

На рис. 5 показано получение четырехугольных элементов по новому алгоритму, когда нормаль (обозначена стрелкой) из узла A пересекает третье (см. рис. 5, слева) или четвертое (см. рис. 5, справа) ребро фронта. Во втором случае формируемое ребро фронта AD заменяется на два ребра AP и PD . Тем самым сохраняется четность фронта, а вырожденность ячейки $APDE$ в дальнейшем устраняется операциями топологической оптимизации и сглаживания.

После формирования четырехугольников активным становится следующее ребро фронта BC . Процесс продолжается, пока весь узкий фронт не закроется.

Обработка шеститочечных фронтов. В работе [7] каких-либо рекомендаций по обработке шеститочечных фронтов предложено не было. Интуитивно понятно, что образуемый ребрами фронта шестиугольник можно трансформировать в два четырехугольника. Для этого требуется применить один из трех вариантов разбиения. В случае невыпуклых шестиугольников для выбора корректного разбиения (рис. 6, в) требуется каждый раз выполнять проверку получаемых четырехугольников на отсутствие пересекающихся ребер (рис. 6, а) и вывернутости (рис. 6, б).

На рис. 7, а показан пример, когда нельзя провести корректное разбиение на два четырехугольника. В этом случае шестиугольный фронт

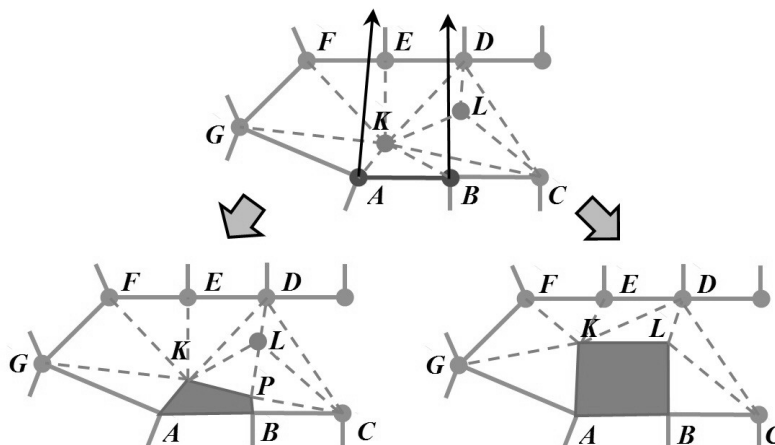


Рис. 4. Формирование четырехугольника в узком фронте по базовому алгоритму: добавлением точки P на пересекаемое ребро KC (слева); созданием ребра BL с помощью поворота ребра KC (справа)

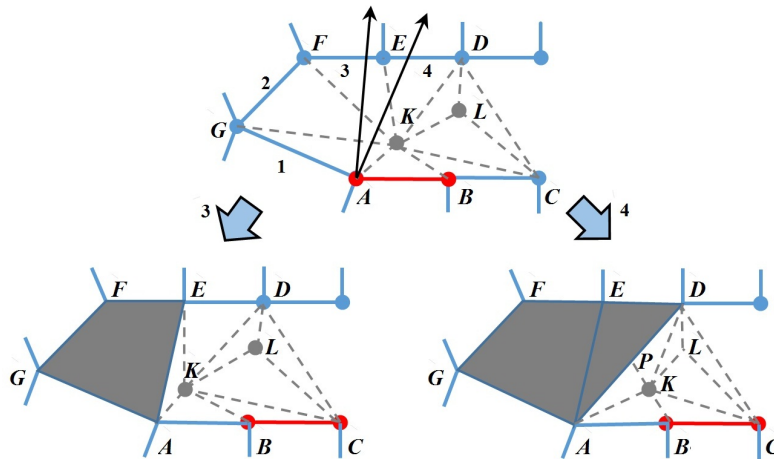


Рис. 5. Формирование из узла A активного ребра фронта AB одного четырехугольника (слева); двух четырехугольников с добавлением средней точки P ребра AD (справа)

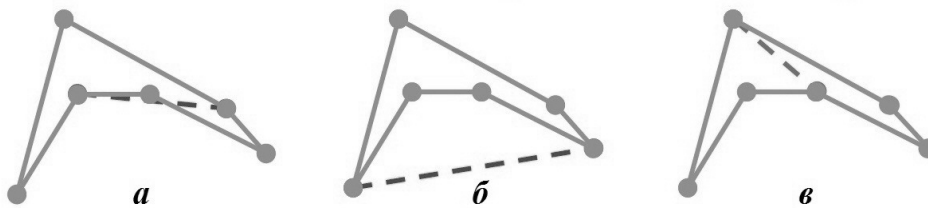


Рис. 6. Три варианта построения диагонали невыпуклого шестиугольника: a — с образованием четырехугольника с пересекающимися ребрами; $б$ — с образованием вывернутого четырехугольника; $в$ — с корректным разбиением

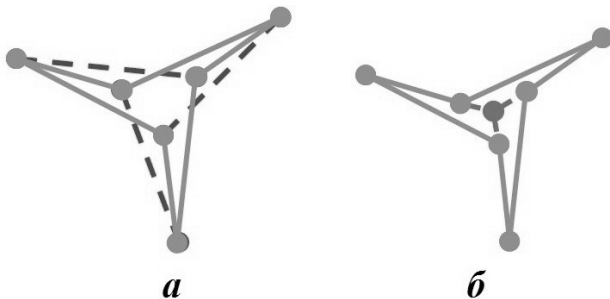


Рис. 7. Разбиение невыпуклого шестиугольника: a — на два четырехугольника (некорректное); $б$ — на три четырехугольника (корректное)

можно корректно разбить на три четырехугольника (рис. 7, $б$).

В итоге для разрешения шестиугольных фронтов был выбран алгоритм из работы [8], основанный на классификации шестиугольника по четырем шаблонам в зависимости от последовательности состояний фронтальных ребер. Предложенные в [8] шаблоны приведены в табл. 2. Числа в последовательности над каждым шаблоном

указывают количество исходных ребер, образующих одну сторону шаблонной фигуры. В зависимости от шаблона четырехугольник может быть разбит на 2, 3 или 4 четырехугольника.

Следует отметить, что алгоритм шеститочечных фронтов также потребовал доработки. Например, согласно базовому алгоритму для шестиугольника, изображенного на рис. 6, подходящим является шаблон 3-2-1 с результатом, показанным на рис 6, a . Но в этом случае образуется четырехугольник с пересекающимися ребрами. В связи с этим для нескольких шаблонов в алгоритм добавлены проверки на образование таких четырехугольников. Также был расширен список шаблонов: добавлен шаблон *пятиугольник*, показанный на рис. 8.

Восстановление ребра. Одним из этапов формирования ячейки является процедура восстановления ребра. Например, для восстановления ребра CD (рис. 9) необходимо определить набор ребер, пересекаемых CD , и затем путем последовательных поворотов совместить одно из ребер с точками C и D [7].

Таблица 2

Шаблоны, предложенные в [8] для классификации шестиугольника

Прямоугольник	Треугольник	Сегмент круга	Круг
2-1-2-1	2-2-2	3-3	1-1-1-1-1-1
3-1-1-1	3-2-1 (1-2-3)	4-2	6
2-2-1-1	4-1-1	5-1	

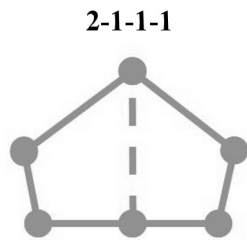


Рис. 8. Шаблон пятиугольник

Достаточно часто возникают ситуации, когда на пересекаемом ребре уже сформирована четырехугольная ячейка или оно принадлежит какому-либо контуру, ограничивающему область построения сетки. Поскольку в данных случаях восстановление ребра невозможно, выполняется переход к следующему шагу основного цикла,

где происходят смена активного ребра фронта и попытка сформировать на нем новую ячейку.

Поворот ребра разрешается, если содержащие его треугольники после поворота не являются вывернутыми. При нарушении этого условия осуществляется возврат сетки в состояние до вызова процедуры восстановления ребра.

Отметим еще один момент. Если при восстановлении ребра требуется повернуть более пяти ребер (данное значение получено апостериорно), то достигнуть хорошего качества результирующей четырехугольной сетки становится невозможным. В подобных ситуациях выполняется отказ от формирования ячейки на данном ребре фронта. Берется в рассмотрение следующее ребро, и делается новая попытка построения четырехугольника уже на нем. В случае закливания по фронту применяются специальные методы для его устранения.

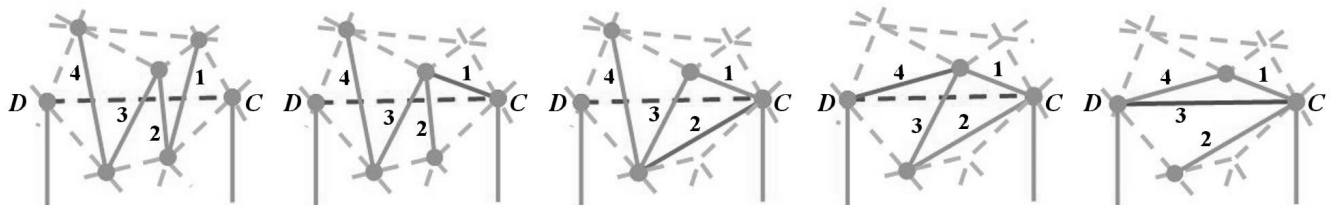


Рис. 9. Последовательность восстановления ребра CD

Объединение ребер фронта. Данная процедура применяется, если в процессе классификации ребер фронта обнаруживается, что угол между активным и предшествующим ему ребрами фронта меньше $\pi/6$, а соотношение их длин не больше 2. При этом объединение возможно только в случае, когда одно или оба рассматриваемых ребра фронта не принадлежат границе области (рис. 10, 11).

Теоретически объединение производится несложно, такая процедура описана в [7]. Однако практически всегда возникают ситуации, когда в область между объединяемыми ребрами попадают узлы и/или образуются перекресты. Заранее предусмотреть и обработать все возможные ситуации не представляется возможным. Для устранения лишних узлов и

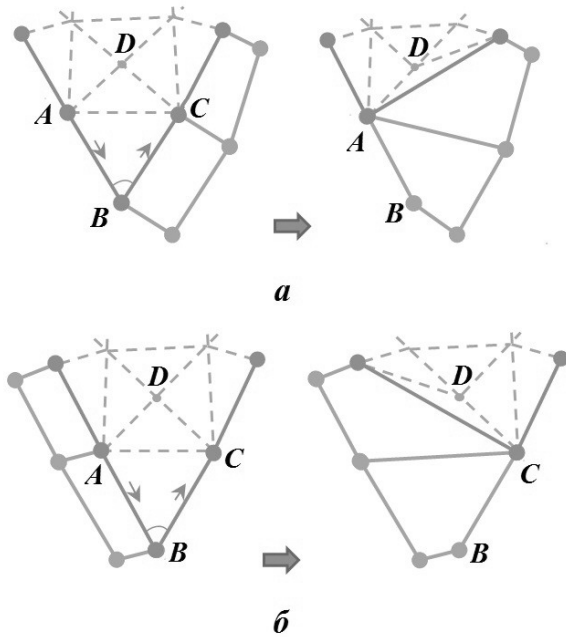


Рис. 10. Схема объединения ребер фронта, когда одно ребро принадлежит границе: *a* — со схлопыванием влево; *б* — со схлопыванием вправо

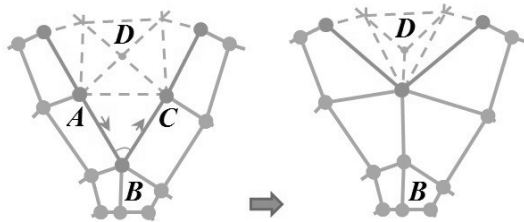


Рис. 11. Схема объединения не принадлежащих границе ребер фронта

перекрестов был реализован следующий алгоритм (см. рис. 10, 11 с объединяемыми ребрами *AB* и *BC*):

- 1) восстанавливаем ребро *AC* (если оно отсутствует);
- 2) находим $\triangle ACD$ — смежный с $\triangle ABC$ по ребру *AC*;
- 3) для четырехугольника *ABCD* проводим процедуру *схлопывания* по ребру *AC* влево ($C \rightarrow A$, см. рис. 10, *a*), вправо ($A \rightarrow C$, см. рис. 10, *б*) или в середину отрезка *AC* (см. рис. 11) в зависимости от ситуации.

Такой подход гарантирует, что никакая лишняя точка не попадет в область схлопывания. Однако предварительно необходимо сделать проверку на возможное появление перекрестов. Для этого производится оценка треугольников, непосредственно примыкающих к области схлопывания. Если какой-нибудь треугольник оказывается вывернутым, то процедура объединения ребер фронта не выполняется и восстанавливается исходная треугольная сетка. Далее в основном цикле в качестве активного берется следующее ребро фронта.

Локальное сглаживание. Сглаживание является одним из наиболее используемых инструментов в процессе построения сеток. Процедура сглаживания способствует улучшению качества элементов посредством придания ячейкам более правильной формы без изменения топологии сетки. Сглаживание осуществляется как в процессе построения ячеек, каждый раз после формирования нового четырехугольного элемента (локальное), так и уже на полностью построенной четырехугольной сетке (глобальное). В работе [7] при выполнении процедур локального и глобального сглаживания используется хорошо себя зарекомендовавший и простой в реализации метод Лапласа [9].

На практике выяснилось, что для областей сложной конфигурации локальный метод Лапласа не достаточен для получения сеток хорошего качества. Поэтому за основу процедуры локального сглаживания взят многоэтапный алгоритм, предложенный Т. Блэкером и М. Стивенсоном в [8]. Тестовые расчеты выявили ряд проблем и недостатков в используемом алгоритме многоэтапного локального сглаживания, которые, в свою очередь, потребовали изменения некоторых этапов алгоритма.

Первым важным изменением является введение очередности при обработке узлов сглаживаемой ячейки. На рис. 12 красным цветом выделена вновь построенная ячейка, а зеленым обведены обрабатываемые в этой ячейке узлы. Рассматриваются узлы двух типов в зависимости от количества четырехугольников, в которые эти узлы входят, — одного или двух.

Наивысшим приоритетом при обработке обладают узлы, входящие только в один четырехугольник, т. е. принадлежащие только обрабатываемой четырехугольной ячейке (узел 1 на рис. 12). Затем обрабатываются узлы, входящие в две четырехугольные ячейки со смежным ребром (узел 2 на рис. 12). Положение всех остальных узлов ячейки в процессе сглаживания не корректируется. Данный подход способствует улучшению сохранения формы и размера как уже построенных четырехугольных ячеек, так и вновь образованных.

Помимо изменения подхода к порядку обработки узлов, модификации подверглись также и некоторые этапы их обработки.

На рис. 13 показана корректировка положения узла первого типа (входящего только в одну четырехугольную ячейку). Сначала определяются оптимальные положения ребер обрабатываемой ячейки, содержащих рассматриваемый узел i . Ребра P_{ij} и P_{ik} поворачиваются в зависимости от фронтальных углов α и β , также производится корректировка их длин [8]. Для узла i вычисляются приращения координат $\Delta\vec{v}_{ij}$ и $\Delta\vec{v}_{ik}$ от ребер P_{ij} и P_{ik} соответственно. Итоговое приращение, определяющее искомое положение узла, вычисляется по формуле

$$\Delta\vec{v} = \frac{\Delta\vec{v}_{ij} + \Delta\vec{v}_{ik}}{2}.$$

Также внесены изменения в процедуру обработки узлов второго типа. На рис. 14 показано положение узла i после поворота и начальной

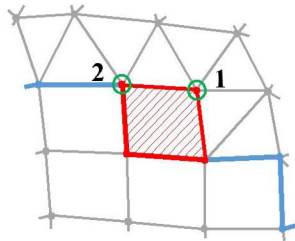


Рис. 12. Узлы новой ячейки, обрабатываемые в процедуре локального сглаживания

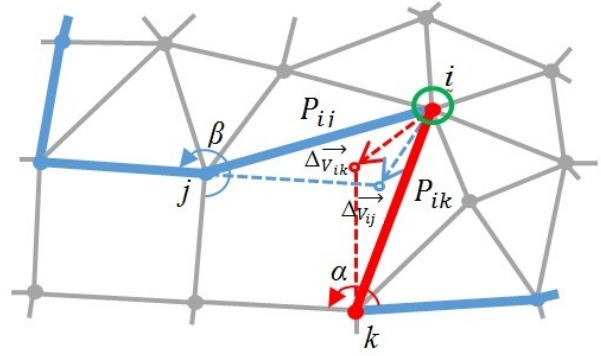


Рис. 13. Определение положения узла, принадлежащего только одной четырехугольной ячейке

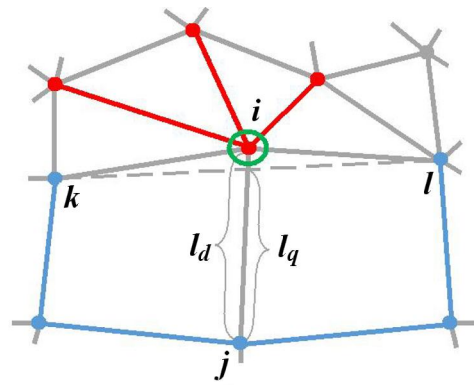


Рис. 14. Дополнительная корректировка длины ребра между узлами i и j

корректировки длины инцидентного ребра P_{ij} (между узлами i и j) [8]. Так называемая желаемая длина l_d вычисляется с помощью длин ребер четырехугольных и треугольных ячеек следующим образом:

$$l_d = \frac{l_{quad} + \sum_{i=1}^{N_{tr}} l_i}{4 + N_{tr}},$$

где l_{quad} — сумма длин ребер, не содержащих корректируемый узел и входящих в четырехугольные ячейки (выделены синим цветом); l_i и N_{tr} — соответственно длины ребер треугольных ячеек, содержащих узел i и не являющихся ребрами четырехугольников (выделены красным цветом), и их количество.

На конечном этапе обработки узла i происходит еще одна корректировка длины смежного ребра $|P_{ij}|$, которая выполняется по следующему правилу (см. рис. 14):

$$|P_{ij}| = \begin{cases} \frac{l_q + l_d}{2}, & \text{если } l_d > l_q; \\ l_d, & \text{если } l_d \leq l_q. \end{cases}$$

Здесь l_q — расстояние от узла j до точки пересечения отрезков, соединяющих узлы i и j , k и l . Данная корректировка способствует минимальному искажению фронта при построении новых элементов.

Топологическая оптимизация. Для реализации процедур топологической оптимизации были использованы результаты работ [7, 10–12]. Основная цель топологической оптимизации — уменьшить количество *нерегулярных* узлов и их концентрацию, особенно около границ области. Регулярность узла определяется его валентностью — количеством выходящих из узла ребер. Для четырехугольных сеток регулярным является четырехвалентный узел. Для граничных узлов валентность определяется с учетом внутреннего угла α между двумя инцидентными узлу граничными ребрами и числа исходящих ребер N_r .

Граничный узел считается регулярным, если его валентность V_r соответствует следующим значениям:

$$V_r = \begin{cases} 2, & \text{при } \alpha < \frac{2\pi}{3}; \\ 3, & \text{при } \frac{2\pi}{3} \leq \alpha < 1,206\pi; \\ 4, & \text{при } 1,206\pi \leq \alpha < 1,717\pi; \\ 5, & \text{при } 1,717\pi \leq \alpha. \end{cases}$$

Окончательно валентность граничного узла вычисляется по следующей формуле:

$$V = 4 + N_r - V_r.$$

Следуя [10], большинство операций топологической оптимизации сводится к последовательному применению базовых операций: схлопыванию/раскрытию четырехугольника, добавлению/удалению двухвалентного узла. Данные операции проиллюстрированы на рис. 15, 16.

На рис. 17, 18 показаны примеры комбинаций операций схлопывания для удаления нерегулярных узлов.

Операции оптимизации выполняются в соответствии с уменьшением приоритета устранения нерегулярности узлов в следующем порядке:

- 1) удаление двухвалентных узлов;

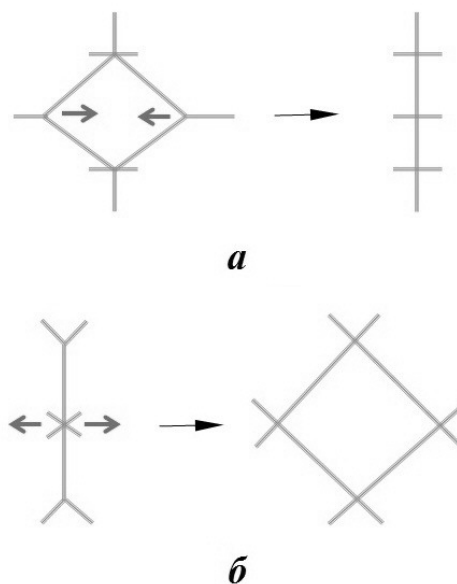


Рис. 15. Операции схлопывания (а) и раскрытия (б) четырехугольника

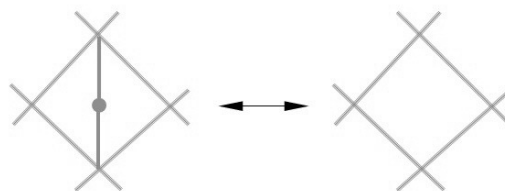


Рис. 16. Операции добавления/удаления двухвалентной вершины

- 2) удаление нерегулярных граничных узлов;
- 3) удаление трехвалентных узлов;
- 4) удаление шестивалентных узлов. На данном этапе выполняется операция раскрытия четырехугольника по ребрам, имеющим минимальную сумму валентностей узлов;
- 5) процедура релаксации, направленная на удаление комбинаций трех- и пятивалентных узлов [7].

После каждой операции на этапах 3–5 проверяется возникновение нерегулярных соседних узлов и при их наличии вызывается соответствующая процедура их удаления. Если в результате операции нарушается регулярность граничных узлов, то операция отменяется.

После ряда операций топологической оптимизации вызывается процедура сглаживания для затронутых узлов. По окончании всего цикла производится сглаживание всей полученной сетки.

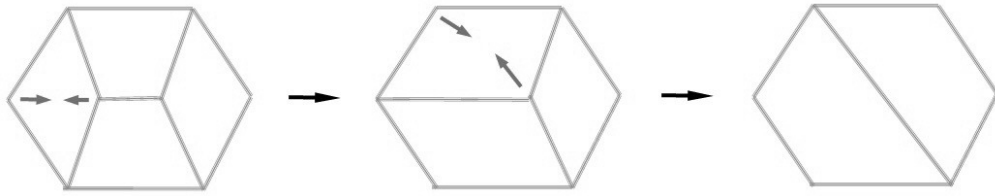


Рис. 17. Удаление двух соседних трехвалентных узлов [12]

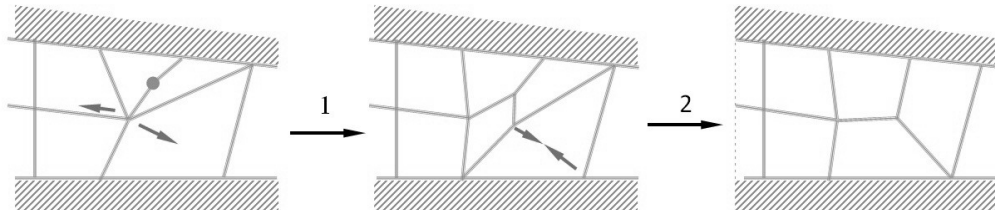


Рис. 18. Удаление двухвалентного узла около границы с помощью двойной операции раскрытия (1) и схлопывания (2) четырехугольника [10]

4. Примеры построения сеток

Рассмотрим примеры построенных сеток для нескольких геометрических моделей произвольной формы. На рис. 19 представлены результаты построения четырехугольных сеток с использованием генератора, реализующего базовые процедуры алгоритма [7], и модифицированного генератора. Видно, что внесенные в алгоритм изменения качественно улучшают получаемые

сетки. Наблюдается ортогональность ячеек с границей области, размер ячеек более равномерный, сетка имеет меньше измельчений и более регулярный характер.

На рис. 20 показаны результаты работы генератора поверхностных неструктурированных четырехугольных сеток в препроцессоре ПП "Логос", полученные для одной из граней пространственной модели на этапах построения 3–5 (см. разд. 1).

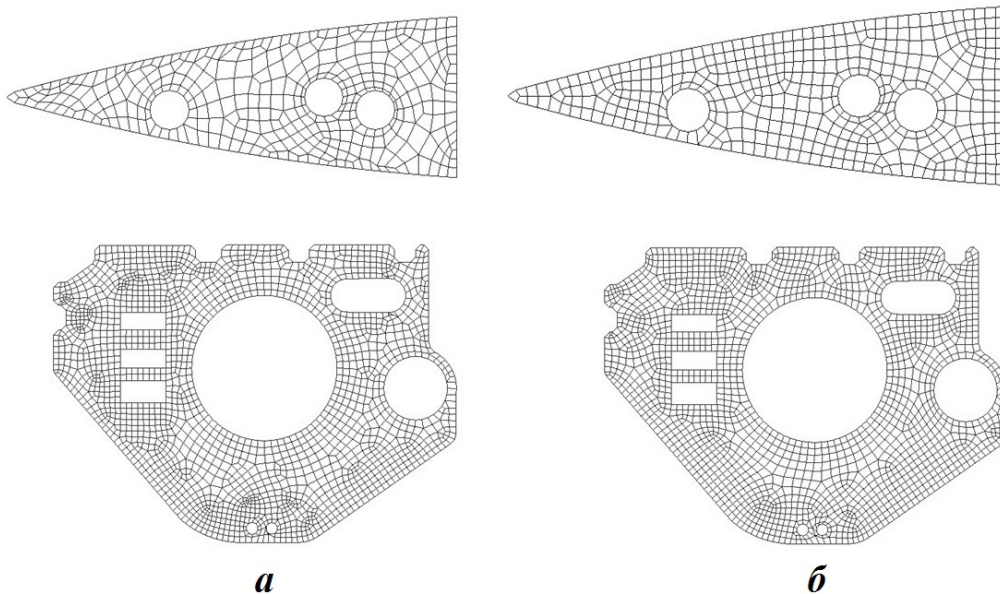


Рис. 19. Четырехугольные сетки, построенные по базовым процедурам алгоритма [7] (а) и по модифицированному алгоритму (б)

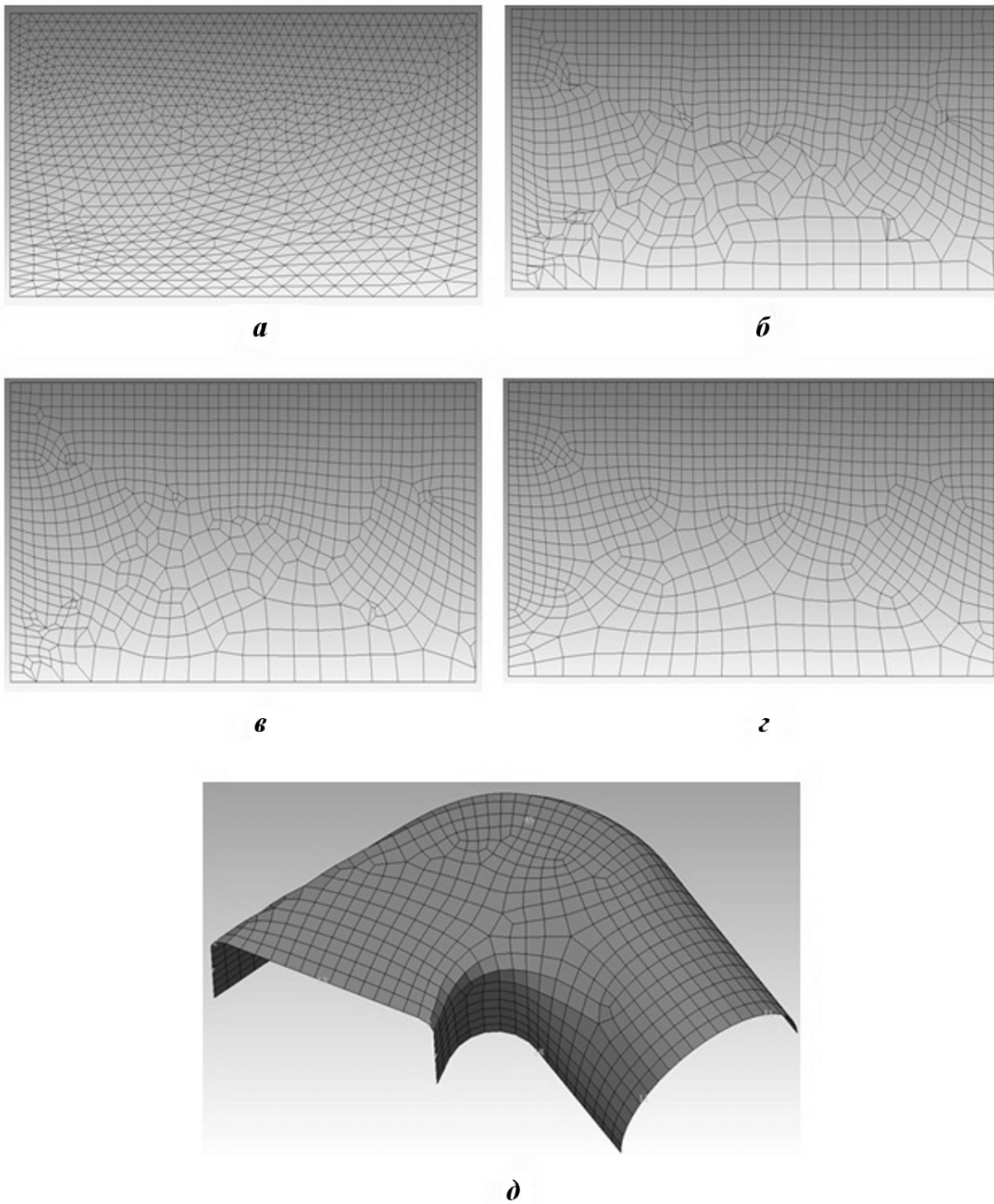


Рис. 20. Поэтапное построение поверхностной сетки на грани пространственной геометрической модели: *a* — треугольная сетка в параметрической плоскости на этапе 3; *б* — четырехугольная сетка в параметрической плоскости на этапе 4 после построения; *в* — четырехугольная сетка в параметрической плоскости после сглаживания на этапе 4; *г* — четырехугольная сетка в параметрической плоскости после топологической оптимизации на этапе 4; *д* — обратное отображение сетки из параметрической плоскости в пространство R^3 на этапе 5

Заключение

Приведено описание автоматического генератора поверхностных неструктурированных четырехугольных сеток в произвольных областях. Используется методика, позволяющая строить поверхностные сетки при помощи плоского генератора с учетом римановой метрики поверхности для контроля размера и формы четырехугольников в параметрическом пространстве. Данный генератор поверхностных сеток включен в состав препроцессора ПП "Логос".

Представлены значимые изменения, коснувшиеся алгоритмов формирования ячеек и обработки фронтов и позволившие улучшить качество отстраиваемых сеток.

Разработаны и реализованы новые алгоритмы локального сглаживания и оптимизации.

Дальнейшее развитие созданного средства генерации поверхностной сетки планируется в направлении улучшения качества имеющихся и реализации новых алгоритмов генерации и оптимизации сеток.

Список литературы

1. *Bate K.-Ю.* Методы конечных элементов. М.: Физматлит, 2010.
Bate K.-Yu. Metody konechnykh elementov. M.: Fizmatlit, 2010.
2. Пакет программ инженерного анализа и суперкомпьютерного моделирования. <http://logos.vniief.ru>.
Paket program inzhenerного analiza i superkompyuternого modelirovaniya. <http://logos.vniief.ru>.
3. *Owen S. J., Staten M. L., Cannan S. A., Saigal S.* Advancing front quadrilateral meshing using triangle transformations // Proc. 7th Int. Meshing Roundtable. Dearborn, Michigan, Sandia National Labs. October 26–28, 1998. P. 409–428.
4. *Tristano J. R., Owen S. J., Canann S. A.* Advancing Front Surface Mesh Generation in Parametric Space Using a Riemannian Surface Definition. ANSYS, Inc., 1999.
5. Официальный сайт ANSYS. <http://www.ansys.com>.
Ofitsialny sayt ANSYS. <http://www.ansys.com>.
6. *Сковпень А. В.* Реализация фронтального алгоритма построения нерегулярных четы-

рехугольных сеток // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2005. Вып. 1. С. 9–30.

Skovpen A. V. Realizatsiya frontalnogo algoritma postroeniya neregulyarnykh chetyrekhugolnykh setok // Voprosy atomnoy nauki i tekhniki. Ser. Matematicheskoe modelirovanie fizicheskikh protsessov. 2005. Vyp. 1. S. 9–30.

7. *Лысухин В. И.* Построение сетки четырехугольных элементов на произвольных областях // Прикладная геометрия (электронный журнал МАИ). 2010. Вып. 12, № 24. С. 1–29. <http://www.mai.ru>.
Lysukhin V. I. Postroenie setki chetyrekhugolnykh elementov na proizvolnykh oblastiakh // Prikladnaya geometriya (elektronny zhurnal MAI). 2010. Vyp. 12, № 24. S. 1–29. <http://www.mai.ru>.
8. *Blackler T. D., Stephenson M. B.* Paving: a new approach to automated quadrilateral mesh generation // Int. J. Num. Meth. Eng. 1991. Vol. 32. P. 811–847.
9. *Field D. A.* Laplacian smoothing and delaunay triangulation // Communications in Applied Numerical Methods. 1988. Vol. 4. P. 709–712.
10. *Canann S. A., Muthukrishnan S. N., Phillips R. K.* Topological improvement procedures for quadrilateral finite element meshes // Engineering with Computers. 1998. Vol. 14. P. 168–177.
11. *Kinney P.* CleanUp: Improving Quadrilateral Finite Element Meshes // Proc. 6th Int. Meshing Roundtable. Ford Motor Company, 1997. P. 449–467.
12. *Lo S. H.* Finite Element Mesh Generation. CRC Press, 2015.
13. *Беспалов А. П., Логинов Д. В., Соловьёв А. Н.* Метод автоматического построения поверхностных сеток на гранях геометрических моделей в параметрическом представлении в препроцессоре пакета программ "Логос" // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2020. Вып. 3. С. 23–39.
Bespalov A. P., Loginov D. V., Solovyev A. N. Metod avtomaticheskogo postroeniya poverkhnostnykh setok na granyakh geometricheskikh modeley v parametricheskom predstavlenii v

- preprotsessore paketa programm "Logos" //
Voprosy atomnoy nauki i tekhniki. Ser.
Matematicheskoe modelirovanie fizicheskikh
protsessov. 2020. Vyp. 3. S. 23–39.
14. Геометрическое ядро C3D. [https://
c3dlabs.com/ru/products/c3d-kernel/](https://c3dlabs.com/ru/products/c3d-kernel/).
Геометрическое ядро C3D. [https://
c3dlabs.com/ru/products/c3d-kernel/](https://c3dlabs.com/ru/products/c3d-kernel/).
15. Генератор сеток CM2 MeshTools. [https://
www.computing-objects.com/cm2-meshtools-
suite/](https://www.computing-objects.com/cm2-meshtools-suite/).
- www.computing-objects.com/cm2-meshtools-
suite/.
Generator setok CM2 Mesh Tools. [https://
www.computing-objects.com/cm2-meshtools-
suite/](https://www.computing-objects.com/cm2-meshtools-suite/).
- Статья поступила в редакцию 17.03.20.

AN UNSTRUCTURED QUADRILATERAL SURFACE grid GENERATOR IN THE
PREPROCESSOR OF THE "LOGOS" SOFTWARE PACKAGE / E. Yu. Arapova,
V. G. Kudelkin, E. A. Pavlov, S. Yu. Polyakova, A. V. Tikhonov (FSUE "RFNC-VNIIEF",
Sarov, N. Novgorod region).

The paper describes an automatic generator of unstructured quadrilateral surface grids for geometric models in the parametric representation. The grid generator is intended to construct grid models for strength simulations in the "Logos" software package. Steps of the surface grid generation procedure are considered. One of the steps, namely generation of quadrilateral grids in the parametric plane, is discussed in detail. Examples of test problems with different geometric models are given.

Keywords: "Logos" software package, quadrilateral surface grids, moving front, generation of quadrilateral cells, local smoothing, topological optimization.
