

УДК 519.6

DOI: 10.53403/9785951505309_2022_27_1_58

Метод автоматического построения поверхностных сеток на гранях геометрических моделей в параметрическом представлении в препроцессоре пакета программ ЛОГОС

Приводится описание метода автоматического построения поверхностных полностью и преимущественно четырехугольных сеток для геометрических моделей в параметрическом представлении. Метод реализован в рамках пакета программ ЛОГОС и используется при решении задач прочности. Рассматриваются ключевые аспекты и логика выполнения алгоритмов построения поверхностных сеток в препроцессоре ЛОГОС-Препост.

**А. П. Беспалов, Д. В. Логинов,
А. Н. Соловьев**

Введение

При проведении расчетно-теоретического обоснования проектных прочностных характеристик создаваемых трехмерных конструкций и их элементов в настоящее время повсеместно используются вычислительные методы компьютерного моделирования. Одним из широко распространенных является метод конечных элементов (МКЭ) [1]. Для расчета с использованием МКЭ необходимо выполнить декомпозицию области исследуемой сплошной среды на элементы конечного размера, задать параметры математической модели и запустить моделирующую программу. При моделировании методом МКЭ в трехмерном пространстве широко используются объемные, оболочечные и балочные элементы. Для обеспечения необходимой точности расчета к качеству элементов сеточной модели предъявляются определенные требования с точки зрения их формы

и размера. Зачастую инженеры предпочитают генерировать четырехугольные оболочечные элементы, по форме приближенные к квадрату, и шестигранные объемные, по форме приближенные к кубу, что позволяет обеспечить высокую точность расчета на основе МКЭ. Современные расчетные сетки задач прочности могут включать сотни тысяч и миллионы конечных элементов. Создание расчетных сеток таких размеров вручную не представляется возможным, поэтому в современных инженерных пакетах реализованы различные методы автоматического построения сеток.

В РФЯЦ-ВНИИЭФ с 2010 г. ведется разработка пакета программ (ПП) ЛОГОС [2] имитационного моделирования широкого класса инженерных задач, решаемых на предприятиях высокотехнологичных отраслей промышленности. Для подготовки расчетной модели и анализа результатов расчета в ПП ЛОГОС предназначена программа ЛОГОС-Препост. Начиная с 2016 г. ключевым направлением ее развития является разработка средств подготовки математических моделей для решения различных классов прочностных инженерных задач: статической и динамической прочности, модального анализа, анализа устойчивости, широкополосной случайной вибрации и т. п. [3–5].

Одним из типовых этапов подготовки исходных данных для проведения расчета является построение дискретной модели. Этап построения расчетных сеток для больших конструкторских сборок может занимать значительную долю общего времени подготовки/проведения расчетов и анализа их результатов. Для сокращения времени построения сеток в ЛОГОС-Препост развиваются соответствующие алгоритмы и методы, а также ведется работа по повышению удобства графического пользовательского интерфейса (ГПИ).

На текущий момент в области инженерного программного обеспечения (ПО) сложилось три основных подхода к построению поверхностных сеток [6–8]:

- построение сеток на гранях геометрических моделей;
- перестроение имеющейся поверхностной сетки (remesh), т. е. генерация на дискретных поверхностях;
- построение ячеек вручную (direct meshing), без использования геометрической модели или используя ее элементы (вершины, ребра или грани) в качестве вспомогательных.

Перечисленные подходы реализованы в ряде широко известных инженерных ПП: Simulia ABAQUS [6], ANSYS Workbench [7], Altair HyperMesh [8] и др.

В данной статье будут рассмотрены следующие ключевые аспекты реализации в ЛОГОС-Препост методов построения поверхностных сеток на гранях исходных геометрических моделей: представление геометрических моделей; определение и учет топологической формы граней; хранение информации об ассоциации узлов/ячеек сетки с элементами геометрических моделей; базовые алгоритмы генерации поверхностных сеток; учет контактных ребер; хранение данных генератора; скорость генерации. Вместе с тем будут приведены некоторые полученные результаты, которые покажут пригодность реализованных алгоритмов и методов для генерации поверхностных сеток при решении прикладных инженерных задач прочности.

Анализ подходов, реализованных в коммерческих инженерных ПП, показывает, что генерация поверхностных сеток на геометрических гранях в параметрическом представлении состоит из двух этапов. Это подготовка геометрии, включающая выполняемую инженером декомпозицию граней исходной геометрии задачи на конечное количество граней известной топологии, и непосредственно генерация сетки, выполняемая автоматически. При этом топология получаемых граней может соответствовать некоторому заданному набору шаблонов (треугольник, четырехугольник, круг и др.). Система автоматического построения сеток с учетом небольшого количества параметров, заданных пользователем, выполняет генерацию структурированных или блочно-структурированных сеток на гранях известной топологии, согласуя их *узел в узел* с другими сетками на границах (ребрах) соседних граней. На оставшихся гранях неизвестной топологии стро-

ится треугольная, преимущественно четырехугольная или четырехугольная сетка, например, методом движущегося фронта.

Указанная технология была взята за основу при реализации средств построения поверхностных сеток в ЛОГОС-Препост для решения инженерных задач прочности.

Требования к средствам построения поверхностных сеток

Анализ требований, предъявляемых к генерации поверхностных сеток на гранях исходных геометрических моделей, позволяет заключить, что в ЛОГОС-Препост должны быть обеспечены следующие возможности:

- генерация сетки на наборе граней в параметрическом представлении;
- генерация сеток, согласованных или несогласованных на границах смежных граней;
- поддержка следующих методов/алгоритмов генерации сетки на отдельной взятой грани:
 - отображения при получении блочно-структурированной сетки;
 - движущегося фронта для треугольной, преимущественно или полностью четырехугольной сетки;
 - получения LR треугольной сетки разбиением ячеек полностью четырехугольной блочно-структурированной сетки на треугольники;
- задание закона разбиения ребер на указанных гранях: равномерного, геометрического и двойного геометрического распределений;
- учет целевой длины ребра ячейки, общей для выбранного набора геометрических граней;
- автоматический выбор наиболее подходящего метода генерации с целью предоставления пользователю максимально структурированной качественной четырехугольной сетки. В случае невозможности генерации блочно-структурированной сетки – генерация сетки методом движущегося фронта;
- учет кривизны поверхности;
- учет «близости» ребер. В данном случае имеется в виду автоматическое уменьшение размера генерируемых ячеек в областях, где расстояние между ребрами, расположенными на «противоположных» сторонах грани, меньше, чем выбранный пользователем целевой размер ячейки;
- генерация согласованных сеток для сценариев неоднозначной (non-manifold) топологии, т. е. для граней, имеющих несколько контактных ребер;
- генерация несогласованных сеток для граней, имеющих «общие» и/или контактные ребра;
- поддержка режима *авторазбиения*: грань представляется (если это возможно) в виде набора четырехугольных областей, в каждой из которых строится структурированная сетка.

Общий алгоритм генерации сетки

Блок-схема общего алгоритма генерации поверхностных сеток на гранях геометрических моделей в параметрическом представлении, реализованного в ЛОГОС-Препост, показана на рис. 1.

На первом этапе пользователем вводятся данные, необходимые для генерации: набор граней, целевые размеры генерируемых элементов, тип алгоритма, используемого для генерации, параметры алгоритма.

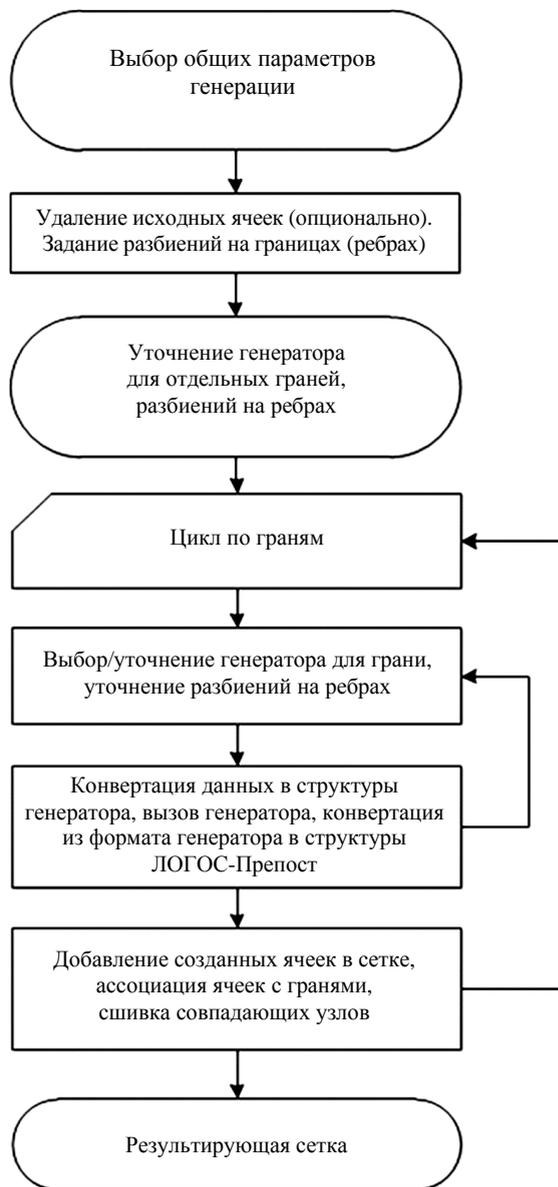


Рис. 1. Блок-схема алгоритма генерации поверхностных сеток

Далее удаляются оригинальные узлы и ячейки сетки, ассоциированные с гранями. Это делается только для граней, на которых сетка была построена ранее. Потом производится разбиение ребер: используется равномерный закон разбиения или закон разбиения с учетом близости ребер или кривизны граней. Этой информации достаточно для генерации поверхностной сетки.

На следующем этапе можно изменить параметры генерации сеток на отдельных ребрах (закон и параметры разбиения) и гранях (тип генератора). На этом ввод данных пользователем заканчивается, выполняется подготовка данных, необходимых для генерации сетки.

Далее независимо для каждой грани вызывается функция генерации сетки. При этом выбирается генератор, с помощью которого будет построена сетка на данной грани.

Тип генератора в явном виде может быть задан пользователем. При невозможности построить сетку этим генератором сетка на данной грани не строится. Если генератор в явном виде пользователем не задан, то осуществляется автоматический подбор подходящего генератора. Для этого выполняется проверка возможности автоматического разбиения грани на блоки, при которой учитываются заданные разбиения ребер и возможность менять эти разбиения. Если топология грани и заданные разбиения позволяют, сетка генерируется с использованием блочного генератора. В этом случае при необходимости грань может быть разбита на блоки и могут быть скорректированы разбиения ребер. В противном случае выполняется генерация методом движущегося фронта.

После окончания генерации поверхностной сетки выполняется конвертация данных в структуры сеточных данных ЛОГОС-Препост. Наконец, в структуру сеточных данных добавляются новые узлы и ячейки и каждая ячейка отмечается как ассоциированная с исходной гранью.

Представление геометрических моделей и расчетной сетки

Одним из ключевых элементов любой современной системы инженерного анализа является используемое геометрическое ядро. Это программный компонент, предоставляющий разработчикам структуры данных для представления элементов геометрических моделей и конструкторских сборок в памяти компьютера, а также набор математических алгоритмов, позволяющих выполнять операции редактирования элементов геометрий. В большинстве известных случаев геометрические ядра сторонней разработки, например, такие как Parasolid [9] (коммерческий продукт) или Open Cascade [10] (программный продукт с открытым кодом), хотя авторы некоторых программных продуктов используют геометрические ядра собственной разработки. Разработчиками ЛОГОС-Препост выбран лицензионный коммерческий продукт – геометрическое ядро C3D [11].

Геометрическая модель в ядре C3D описывается с помощью граничного представления (B-REP) [12]. Для описания геометрической формы используется набор граней, расположенных на границе между внутренним пространством моделируемого объекта и остальной частью пространства. Грани представляют собой параметрические поверхности, которые стыкуются друг с другом по своим границам, описываемым параметрическими кривыми. Формирование и стыковка граней выполняются во время построения модели. Это обеспечивают методы построения модели и организация данных в C3D. Тело в представлении B-REP схематично изображено на рис. 2.

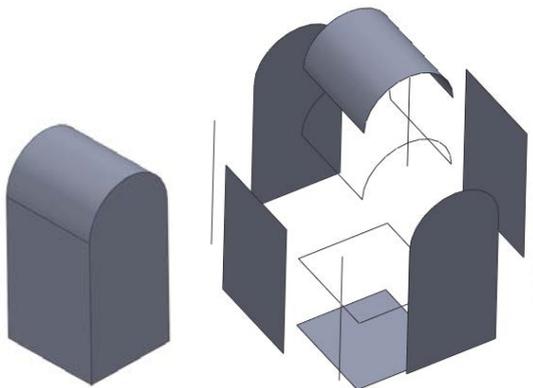


Рис. 2. Схематичное изображение тела в представлении B-REP

Для лучшего понимания реализации средств автоматического построения поверхностных сеток опишем в основных чертах представление геометрических моделей для пользователей ЛОГОС-Препост.

Геометрическая модель в ЛОГОС-Препост представлена в виде набора сплошных тел, *висячих граней*, т. е. граней, не входящих ни в одно сплошное тело, и *висячих ребер*, т. е. ребер, не входящих ни в одну из граней.

Хотя в ЛОГОС-Препост висячие грани и ребра выглядят как автономные составляющие геометрической модели, они входят в состав оболочечных (незамкнутых) или каркасных тел – именно в таком виде они хранятся в геометрическом ядре [11].

Отметим, что исходная конструкторская сборка может иметь в своем составе множество экземпляров одного тела или других объектов ядра в различных местах. Однако для упрощения работы с геометрическими моделями после загрузки исходной сборочной структуры выполняется ее *распластывание*, т. е. вставки тел заменяются на копии. Негативными сторонами данного подхода являются повышенный расход оперативной памяти и задержка, необходимая для выполнения алгоритма распастывания.

С3D выполняет 3D-моделирование с использованием подхода, не допускающего наличия многосвязности (non-manifold) в топологии элементов геометрической модели, т. е., проще говоря, ребро может разделять только две геометрические грани, а грань не может входить в два различных тела. Вопросы non-manifold-представления топологии геометрических моделей является предметом научного исследования (см., например, [13–15]). Примеры многосвязанных топологий приведены на рис. 3.

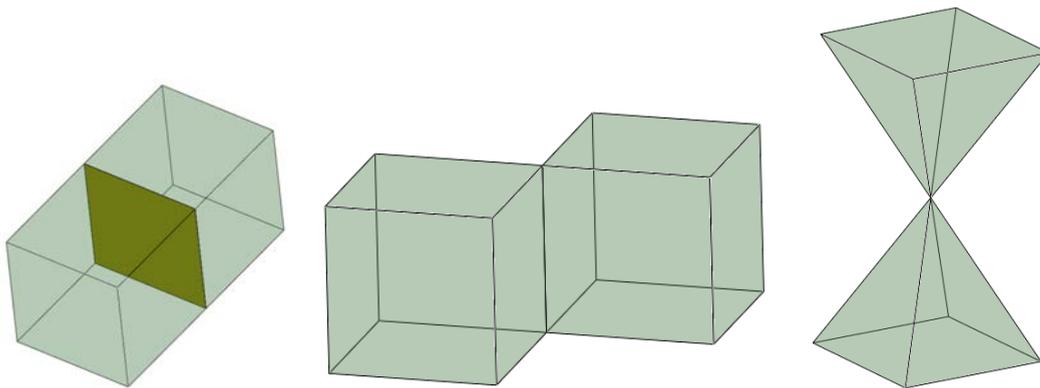


Рис. 3. Примеры неодносвязанных топологий

Однако в целях обеспечения эффективности построения 2D- и 3D-сеток в ЛОГОС-Препост на основе базовых алгоритмов геометрического ядра реализована частичная поддержка non-manifold-топологии для тел, а в средствах генерации поверхностных сеток автоматическими методами – для ребер. После выполнения пользователем операций разрезания тел запоминаются контактные грани, а при генерации поверхностных сеток определяются контакты граней по ребрам. После выполнения операций редактирования геометрических моделей информация о контактах обновляется.

К элементам сеточной модели в ЛОГОС-Препост относятся узлы, определяющие геометрию расчетной сетки, и ячейки сетки, определяющие ее связность. Могут использоваться ячейки с размерностью от 0D (частицы) до 3D (объемные ячейки). Поверхностные сетки состоят из ячеек размерностью 2D, т. е. треугольников и четырехугольников. Кроме линейных конечных элементов могут быть использованы квадратичные конечные элементы. Для хранения ячеек сетки используется ячеечно-узловая структура данных, хранящая списки ячеек, сгруппированных по типам.

При работе в ЛОГОС-Препост с геометрией и сеткой пользователь оперирует понятием *компонент*, которое является близким аналогом понятия *деталь* из области САПР. В отличие от

детали компонент может хранить в себе элементы сеточной модели. Однако, поскольку один и тот же узел сетки может входить в состав ячеек, расположенных в различных компонентах, узлы не ассоциируются с компонентами, они являются общими для всей расчетной модели.

Топологическая форма граней

Одна из целей при построении поверхностных сеток – чтобы система сама выбирала такой алгоритм генерации 2D-ячеек, которым обеспечивается наиболее качественная сетка. Для прочностных задач качественной обычно считается сетка, состоящая только из четырехугольников «хорошей» формы, т. е. приближенных к квадрату. Один из алгоритмов, гарантированно формирующих такие ячейки, – это алгоритм генерации блочно-структурированных сеток. Блочно-структурированные сетки можно генерировать для граней, чья топология соответствует одному из следующих шаблонов: треугольник, четырехугольник, пятиугольник, шестиугольник, полукруг, круг, кольцо.

Ниже приведены упрощенные правила распознавания того, что рассматриваемая грань соответствует какому-либо из указанных топологических шаблонов.

Правило распознавания топологической N -угольной грани: грань должна иметь N углов. В данном случае угол – это пара смежных ребер, угол между касательными к которым в общей вершине превышает пороговое значение.

Топологический полукруг – это фигура, включающая два топологических ребра: одно – прямое, второе состоит из гладко стыкующихся дуг/сплайнов. Топологическое ребро может быть сформировано набором гладко стыкующихся ребер так, что в вершинах, отличных от начальной и конечной, сходятся ровно два ребра.

Топологический круг в частном случае состоит из одного топологического ребра, включающего одно или более гладко стыкующихся ребер – дуг окружностей с одинаковыми параметрами окружностей.

Топологическое кольцо – это топологический круг с круглым же отверстием.

Примеры генерации сеток для граней, имеющих одну из приведенных топологических форм, показаны на рис. 4.

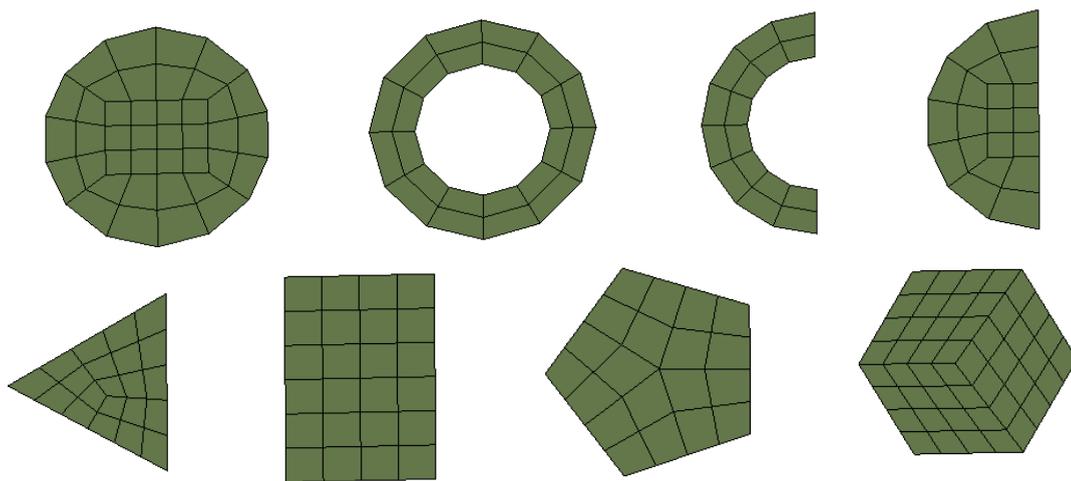


Рис. 4. Генерация блочно-структурированной сетки для граней различной топологии

Для того чтобы можно было использовать информацию о топологии граней при генерации поверхностных сеток, необходимо эту топологию распознать, т. е. найти ее соответствие одному из шаблонов, и далее в ходе выполнения операций редактирования геометрии поддерживать эту информацию в актуальном состоянии.

Если генерация блочно-структурированной сетки невозможна вследствие неподходящей топологии грани или из-за несогласованных разбиений на ребрах грани, система должна использовать резервный алгоритм, который не предполагает какой-либо топологии грани.

Одним из главных ограничений алгоритма генерации полностью четырехугольных сеток является требование к четному разбиению границы грани. Если это требование не выполняется, то такая сетка сгенерирована быть не может и необходимо использовать более общий алгоритм генерации. В качестве такого алгоритма можно взять алгоритм генерации преимущественно четырехугольных сеток.

Ассоциации узлов и ячеек

В ЛОГОС-Препост поддерживается возможность сохранения информации об ассоциации узлов расчетной сетки с вершинами/ребрами и гранями геометрической модели. Это позволяет при перемещении узла, ассоциированного с геометрическим элементом, оставлять его в пределах области этого элемента: в вершине (такие узлы запрещено перемещать), на ребре или в пределах области, занимаемой гранью.

В дополнение к информации об ассоциации узлов в ЛОГОС-Препост может сохраняться информация об ассоциации ячеек с геометрическими элементами – ребрами, гранями и телами. Это может быть полезным для быстрого отбора пользователем ячеек сетки, сгенерированных автоматическими методами для выбранных геометрических элементов.

При генерации поверхностных сеток реализованный механизм ассоциаций используется для хранения привязок *грань – набор ячеек*, сгенерированных автоматическим методом для грани.

Базовые алгоритмы генерации сеток на параметрических гранях

Общий подход. Для генерации в сжатые сроки поверхностных сеток на гранях исходных геометрических моделей в параметрическом представлении необходимо использование автоматических методов генерации с достаточной скоростью выполнения.

Решение типовой задачи генерации поверхностной сетки на геометрической грани в трехмерном пространстве сводится к генерации поверхностной сетки требуемого типа в области параметрического определения грани, т. е. в двумерном пространстве параметров поверхности грани с учетом ее 2D-границ. Схематично данный подход показан на рис. 5.

Алгоритм генерации сетки в двумерном случае управляется набором метрик, связанных с внутренними свойствами поверхности грани. Метрики могут оказывать влияние как на размер генерируемых ячеек, так и на их ориентацию. В итоге данные метрики позволяют контролировать целевую длину ребер создаваемых ячеек [16].

Для реализации предлагаемого метода построения поверхностных сеток были выбраны следующие базовые алгоритмы: блочно-структурированный, метод движущегося фронта, Q-Morph.

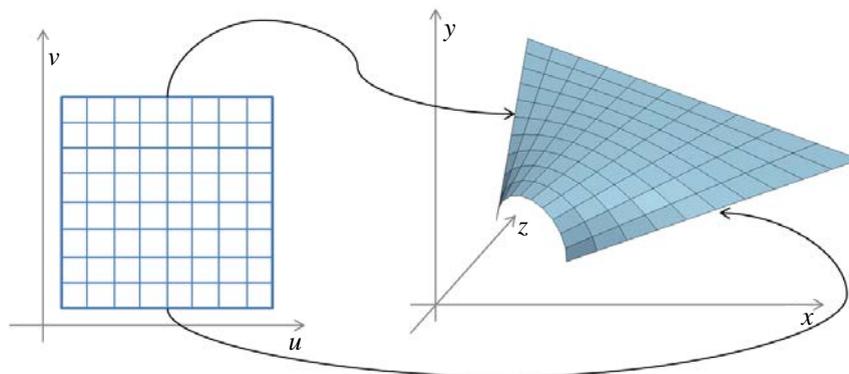


Рис. 5. Общая схема генерации сетки на грани

Блочнo-структурированный метод (автоматическое разбиение граней). Для генерации структурированных сеток используются алгебраические методы. Методы построения сеток, главным образом для топологически четырехугольных или шестигранных геометрий, и методы трансфинитной интерполяции, подходящие для простых форм, обобщены в работе [17]. В основе этих методов явно заданная функция используется для отображения сетки из логической области, имеющей простую форму (единичный квадрат, единичный треугольник и т. п.) в физическую область (см. рис. 5). Созданная таким образом сетка в логическом пространстве является структурированной. В общем случае отображающая функция является полиномиальной.

Однако алгебраические методы, используемые для генерации структурированных сеток, не подходят для построения сеток в сложных областях. Для таких областей может применяться блочно-структурированный метод. Основная идея этого метода состоит в том, чтобы разделить область на более простые, в которых возможна генерация сеток алгебраическими методами. В двумерном пространстве область разделяется на выпуклые (или не слишком деформированные) треугольники, четырехугольники или пятиугольники. Таким образом, ключевым моментом в многоблочном методе является получения подходящего разбиения начальной области. Для разделения области могут применяться как ручные, так и автоматические методы.

Метод движущегося фронта. Генерация сетки методом движущегося фронта исследуется в последние десятилетия, начиная с работы [18], в которой рассмотрены вопросы построения двумерных сеток. Классический метод движущегося фронта в его текущей форме был впервые представлен в работах [19, 20].

Этот метод стартует с дискретизации границы области, представленной в виде ребер в двумерном пространстве или набора треугольных граней в трехмерном. Стратегия данного метода состоит в построении сетки последовательно, ячейка за ячейкой, путем создания новых узлов и соединения их с предварительно созданными ячейками. Таким образом, происходит продвижение вглубь не заполненного ячейками сетки пространства и продвижение *фронта* далее от границы области. Фронтом называется набор ребер ячеек (или граней ячеек в трехмерном случае), отделяющих часть (части) области, заполненную сеткой, от другой части, остающейся ею не заполненной. Процесс останавливается, когда фронт оказывается пустым, т. е. вся область полностью заполнена сеткой.

Схематично процесс построения сетки с использованием алгоритма движущегося фронта представлен на рис. 6.

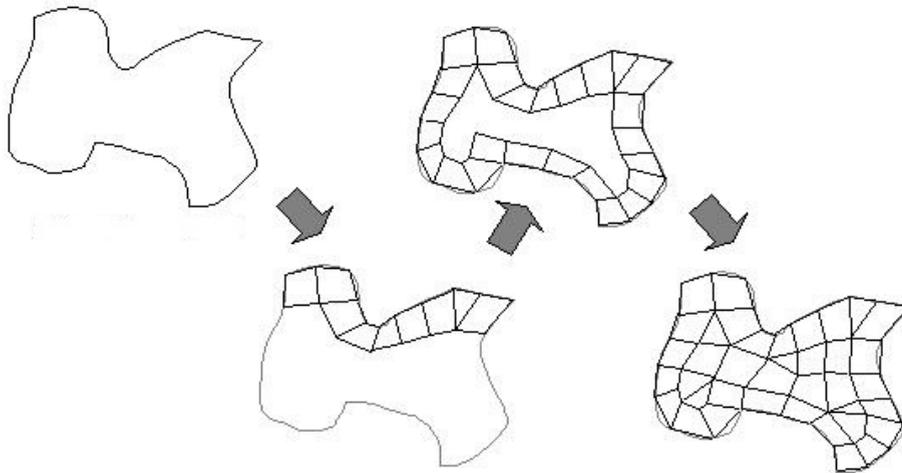


Рис. 6. Поэтапная генерация сетки методом движущегося фронта

Метод генерации объединением треугольников (Q-Morph). Алгоритм Q-Morph является непрямым методом генерации прямоугольной сетки из предварительно построенной треугольной сетки, называемой фоновой. Данный метод похож на другие не прямые алгоритмы тем, что он формирует прямоугольную сетку комбинированием треугольников фоновой сетки. Но его отличием является то, что в процессе построения прямоугольной сетки качество сетки улучшается благодаря использованию метода движущегося фронта для преобразования треугольников в четырехугольники. На рис. 7, взятом из [21], показаны этапы преобразования треугольной сетки в прямоугольную.

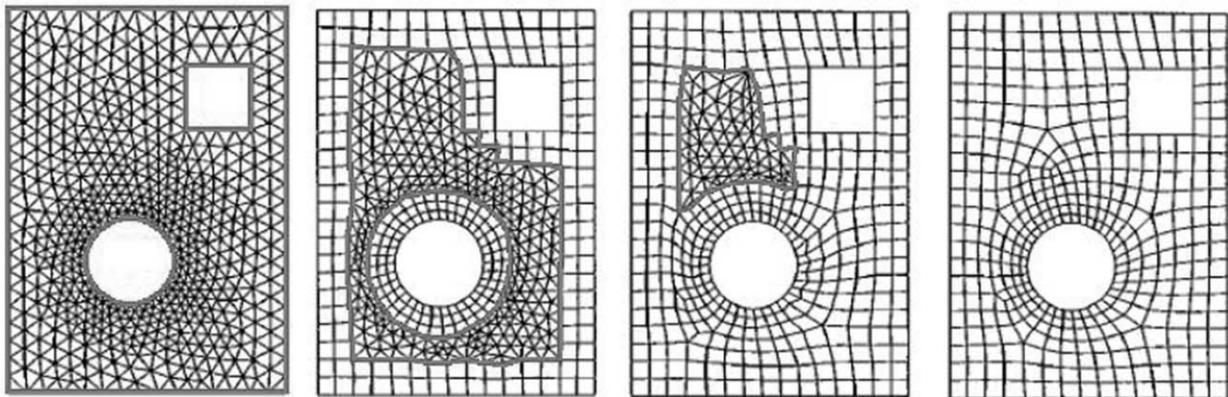


Рис. 7. Несколько стадий алгоритма Q-Morph

Классический алгоритм Q-Morph [21] – это алгоритм генерации сетки в замкнутых областях. В работе [22] рассматривается его модификация для незамкнутых областей, что востребовано при генерации сеток с ограничениями (линейные ограничения в результирующей сетке показаны на рис. 8).

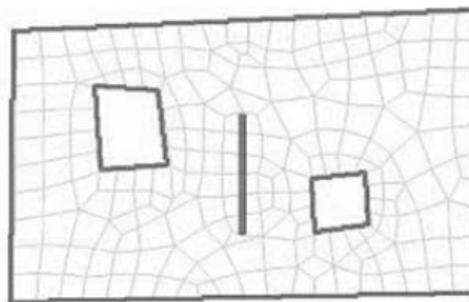


Рис. 8. Линейные ограничения в результирующей сетке

Другими ключевыми классами алгоритмов построения сеток являются алгоритмы оптимизации построенных сеток без изменения их топологии и алгоритмы улучшения (глобального или локального) сетки, которые заключаются в замене двух и более ячеек одной (или наоборот, дроблении ячейки на несколько частей) и удалении ячеек плохого качества.

При реализации метода автоматического построения поверхностных сеток вышеперечисленным базовым алгоритмам соответствуют отдельные программные компоненты.

Учет контактов граней по ребрам

При построении поверхностных сеток для геометрических моделей, подготовленных к решению задачи с использованием оболочечных элементов, особенно часто встречаются сценарии, когда необходимо построение согласованных узлов в узел сеток на гранях, имеющих полностью совпадающие или общие ребра. Такие сценарии демонстрируют классическую проблему неодносвязной топологии [23]. В упрощенном виде типичный пример неодносвязной топологии для четырех граней представлен на рис. 9.

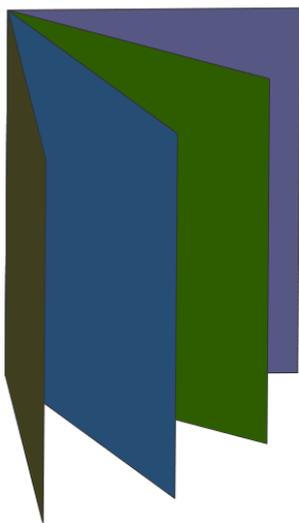


Рис. 9. Пример геометрии с T-соединением (неодносвязная топология)

В некоторых случаях при построении сетки необходимо учитывать наличие *контактных граней* и *контактных ребер*. Контактные грани – это грани, имеющие геометрическое отклонение друг от друга не более чем на заданную величину, которая при самом простом подходе может соответствовать точности сшивки модели. Контактные ребра – это ребра контактных граней, которые определяются аналогично. Также контактными считаются ребра, транзитивно связанные по цепочке контактов. Так, на рис. 10 ребра a и b являются контактными при наличии контакта по граням B, B' , ребра b и c – при наличии контакта по граням C, C' , а ребра a и c – по граням B, B' и C, C' .

Геометрическое ядро не обладает информацией о том, что в геометрической модели есть контактные ребра и грани, поэтому перед построением сетки эту информацию необходимо сформировать.

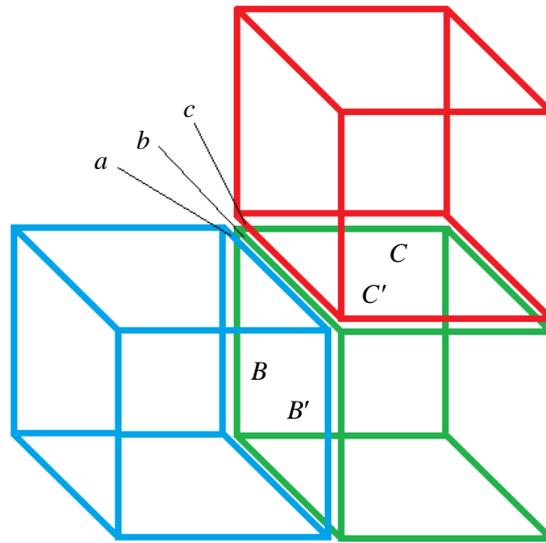


Рис. 10. Пример транзитивно контактных ребер

Хранение данных

В ходе генерации поверхностных сеток необходимо сохранять информацию, которая позволит пользователю продолжить построение сеток на тех гранях, на которых сетка еще не была построена, или перестраивать ранее построенные на гранях сетки.

Отметим, что на текущий момент к ключевым данным, которыми оперирует ЛОГОС-Препост, относятся:

- геометрические данные;
- сеточные данные;
- геометрические контакты тел по смежным граням;
- ассоциации узлов с геометрическими элементами;
- принадлежность ячеек и геометрических элементов компонентам.

Для обеспечения автоматического построения поверхностных сеток в ЛОГОС-Препост требуется дополнительно хранить разбиения на ребрах для сеток, построенных на заданных пользователем гранях.

Графический интерфейс пользователя

Для задания параметров генерации поверхностных сеток крайне важно реализовать удобный ГПИ. Не менее важно предоставить средства визуального отображения заданных пользователем параметров. Удобство здесь рассматривается в контексте объема усилий (перемещение мыши, клики, вызовы операций из контекстного меню и т. п.), которые необходимо затратить для достижения поставленной цели, т. е. для построения сетки с требуемыми характеристиками. Интегральным критерием удобства может служить время, которое пользователь тратит на взаимодействие с ГПИ для решения поставленных задач.

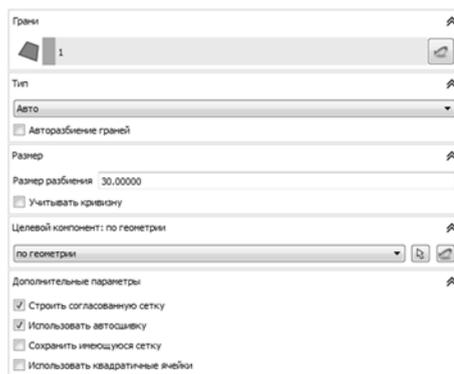
Наиболее подходящей, по мнению авторов, является реализация ГПИ, при которой ввод данных выполняется в два этапа. На первом этапе пользователь вводит общие параметры генерации, определяющие:

- набор граней, для которых будет построена поверхностная сетка;
- целевую длину ребра ячеек;
- алгоритм генерации сетки;
- включение режима авторазбиения геометрических граней;
- необходимость учета кривизны геометрических граней и минимальной длины ребра ячейки;
- включение режима учета близости геометрических ребер;
- целевой компонент, в котором будут размещены сгенерированные ячейки;
- включение режима построения сетки, согласованной по узлам с сетками на соседних гранях;
- объединение совпадающих узлов после генерации сетки;
- необходимость сохранения имеющейся сетки после перехода в режим интерактивного уточнения сгенерированной сетки;
- необходимость генерации квадратичных поверхностных элементов.

На втором этапе задания входных данных пользователь может уточнить тип сетки, генерируемой для отдельных граней из выбранного на первом этапе множества (т. е. базовый алгоритм), а также уточнить закон и параметры разбиения ребер, принадлежащих выбранным граням.

Для того чтобы пользователям ЛОГОС-Препост было легче сфокусироваться на задании параметров генерации для требуемого множества граней, остальные грани (не представляющие интереса для пользователя в текущем контексте) будут скрыты от него в ходе задания входных параметров и генерации сетки.

Внешний вид диалогов ввода данных, необходимых для построения поверхностных сеток, приведен на рис. 11.



а



б

Рис. 11. Внешний вид диалогов ввода данных: а – общих параметров генерации сетки; б – уточнения параметров построения сетки для отдельных граней

Использование ПО

При реализации предложенного метода автоматического построения поверхностных сеток использованы следующие готовые программные библиотеки:

- генерации блочно-структурированных сеток [24], разработанная специалистами ИТМФ РФЯЦ-ВНИИЭФ;
- генерации поверхностных сеток методом объединения треугольников (ИТМФ РФЯЦ-ВНИИЭФ);
- анализа топологии геометрических моделей (ННГУ, г. Нижний Новгород);
- генерации сеток методом движущегося фронта CM2 MeshTools [25].

Примеры построения сеток

Реализованные в ЛОГОС-Препост средства прошли верификацию и валидацию на ряде тестовых примеров, а также на практических задачах.

Ниже приводятся результаты исследований на четырех тестовых моделях. Сетка генерировалась с четырьмя целевыми длинами ребер ячеек: 1, 2, 5 и 10 мм. Дополнительно для двух моделей выполнялось построение сетки в режиме авторазбиения граней с одной из указанных длин ребер ячеек. В качестве результатов фиксировалось количество узлов и ячеек, процент треугольников от общего количества ячеек, время генерации, минимальное значение критерия качества *форма* библиотеки Verdict [26] для сгенерированных ячеек. Для трех тестовых моделей, за исключением первой, получены время построения и скорость генерации – количество сгенерированных ячеек в секунду. Тестирование проводилось на персональном компьютере с процессором Core i5-7200 и ОЗУ 16 Гб.

Получены результаты для следующих тестовых моделей:

1. *Сегмент колокола*. Исходная геометрия модели приведена на рис. 12. Модель содержит 1 тело, 11 граней, 29 ребер, 18 вершин. Результаты построения сетки приведены в табл. 1 и на рис. 13.



Рис. 12. Геометрия модели *Сегмент колокола*

Таблица 1

Результаты построения сетки для модели *Сегмент колокола*

Параметр/показатель	Целевая длина ребра ячейки				
	1 мм	1 мм, авторазбиение граней	2 мм	5 мм	10 мм
Количество узлов	4568	3962	1191	233	86
Количество ячеек	4473	3848	1140	210	86
Количество треугольников, %	1,76	0	3,07	6,19	6,97
Минимальное значение критерия качества <i>форма</i>	0,62	0,82	0,50	0,50	0,55
Количество граней с сетками, построенными блочно-структурированным методом, %	0	100	0	0	0

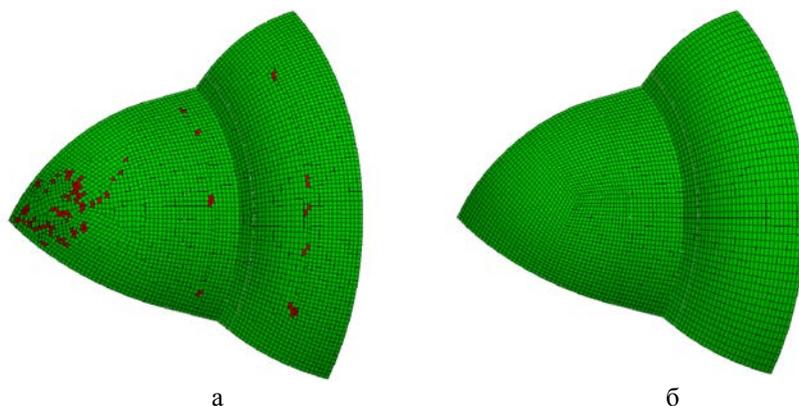


Рис. 13. Сетки, построенные для модели *Сегмент* колокола (длина ребра 1 мм):
а – методом движущегося фронта, б – блочно-структурированным методом

2. *Крепеж* (рис. 14). Содержит 2 тела, 57 граней, 157 ребер, 104 вершины. Результаты построения сетки представлены в табл. 2. На рис. 15 показана сетка, построенная с использованием метода движущегося фронта и блочно-структурированного метода.

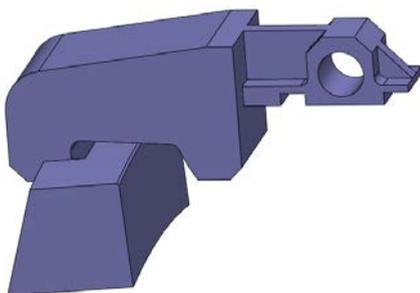


Рис. 14. Геометрия модели *Крепеж*

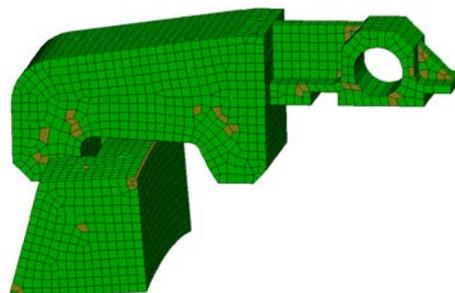


Рис. 15. Сетка для модели *Крепеж* (длина ребра 10 мм), построенная с использованием метода движущегося фронта и блочно-структурированного метода

Таблица 2

Результаты построения сетки для модели *Крепеж*

Параметр/показатель	Целевая длина ребра ячейки			
	1 мм	2 мм	5 мм	10 мм
Количество узлов	479043	121416	19852	5258
Количество ячеек	479922	121636	19916	5290
Время построения, с	54	12	3	1
Скорость генерации	8887	10136	6638	5290
Количество треугольников, %	0,367	0,365	0,662	1,285
Минимальное значения критерия качества <i>форма</i>	0,555	0,551	0,382	0,457
Количество граней с сетками, построенными блочно-структурированным методом, %	61	65	63	65

3. *Графическая карта* (рис. 16). В этой модели 20 тел, 137 граней, 303 ребра, 222 вершины. Результаты построения сетки представлены в табл. 3 и на рис. 17.

4. *Корпус* (рис. 18). В этой модели 1 тело, 27 граней, 52 ребра, 35 вершин. Результаты построения сетки представлены в табл. 4 и на рис. 19.

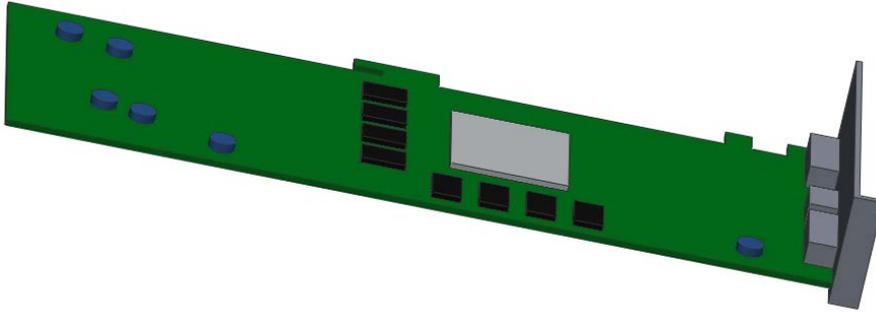


Рис. 16. Геометрия модели *Графическая карта*

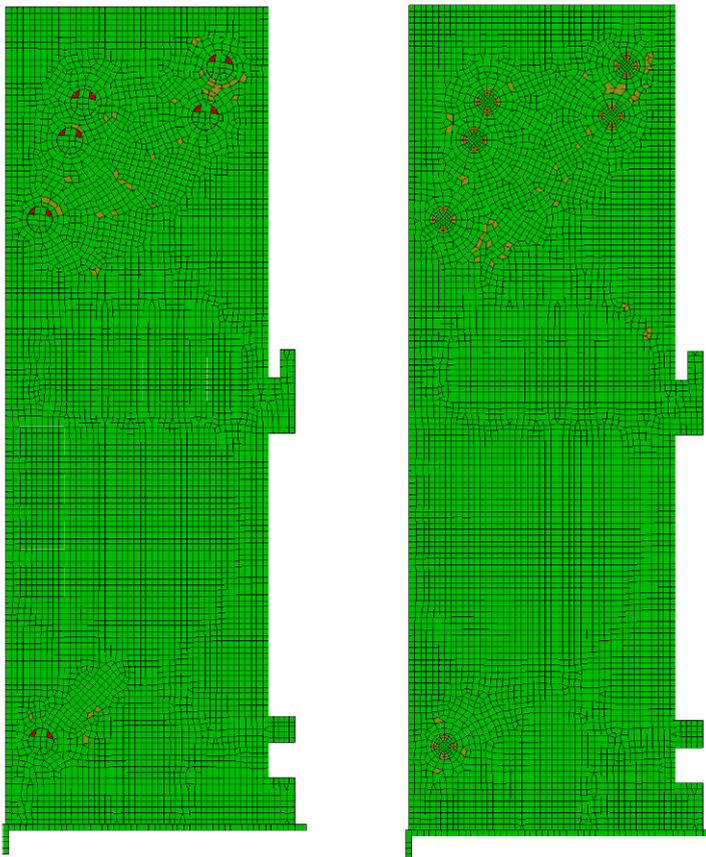


Рис. 17. Сетки, построенные для модели *Графическая карта* (длина 10 мм):
а – без режима авторазбиения граней; б – с режимом авторазбиения граней

Таблица 3

Результаты построения сетки для модели *Графическая карта*

Параметр/показатель	Целевая длина ребра ячейки				
	1 мм	2 мм	5 мм	10 мм	10 мм, авторызбиение граней
Количество узлов	1689497	425053	68505	17082	17674
Количество ячеек	1801138	453913	73407	18490	19343
Время построения, с	390	93	13	4	4
Скорость генерации	4618	4880	5646	4622	4835,75
Количество треугольников, %	0,601	0,651	0,574	1,114	0,744
Минимальные значения критерия качества <i>форма</i>	0,584	0,622	0,554	0,407	0,653
Количество граней с сетками, построенными блочно-структурированным методом, %	82	82	82	82	91

Таблица 4

Результаты построения сетки для модели *Корпус*

Параметр/показатель	Целевая длина ребра ячейки			
	1 мм	2 мм	5 мм	10 мм
Количество узлов	1083206	271696	43718	11770
Количество ячеек	1089409	273675	44170	11939
Время построения, с	239	55	9	2
Скорость генерации	4558,19	4975,90	4907,77	5969,50
Количество треугольников	12410	3962	908	442
Количество треугольников, %	1,139	1,448	2,056	3,702
Минимальное значение критерия качества <i>форма</i>	0,48	0,50	0,49	0,33
Количество граней с сетками, построенными блочно-структурированным методом, %	28	28	28	28

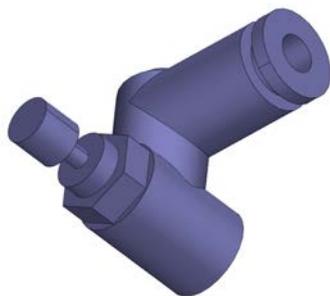


Рис. 18. Геометрия модели *Корпус*

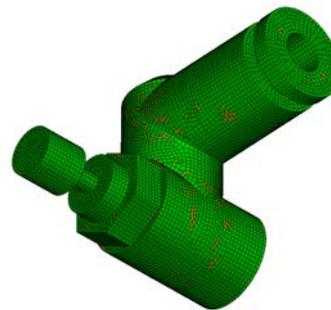


Рис. 19. Сетка, построенная для модели *Корпус* (длина ребра 10 мм)

Из приведенных результатов можно сделать следующие выводы.

1. Чем меньше целевой размер ребра, тем выше минимальное значение критерия качества элементов.

2. Чем более тщательно подготовлена модель к генерации сетки (как сама геометрия, так и параметры генерации), тем меньше время генерации и выше качество ячеек, поскольку основным становится алгоритм генерации блочно-структурированных сеток.

3. По результатам работы в режиме авторазбиения видно, что средства, повышающие «интеллектуальность» системы, снижают трудозатраты пользователя на задание параметров и повышают (в большинстве случаев) качество генерируемых ячеек.

Заключение

Авторами предложен и реализован метод построения поверхностных сеток для наборов геометрических граней в параметрическом представлении, который удовлетворяет требованиям, предъявляемым в препроцессоре ЛОГОС-Препост. Предложенный метод построения поверхностных сеток реализован в ЛОГОС-Препост.

Алгоритмы построения сетки используют возможности геометрического ядра С3D, но могут быть использованы и другие геометрические ядра, поддерживающие параметрическое представление граней.

Созданные средства позволяют строить качественную полностью четырехугольную сетку для геометрических граней, форма которых соответствует следующим шаблонам: треугольник, четырехугольник, пятиугольник, шестиугольник. Их применимость проверена на модельных и практических задачах.

Дальнейшее развитие созданных программных средств планируется в направлениях построения сеток по дискретным поверхностям, улучшения качества имеющихся и реализации новых алгоритмов генерации и оптимизации сеток.

Список литературы

1. Бате К. Ю. Методы конечных элементов. – М.: Физматлит, 2010.
2. Пакет программ инженерного анализа и суперкомпьютерного моделирования [Electronic resource]. – <http://logos.vniief.ru>.
3. Дьянов Д. Ю., Спиридонов В. Ф., Циберев К. В. и др. Пакет программ ЛОГОС. Модуль решения динамических задач прочности // Вопросы атомной науки и техники. Сер. Матем. моделирование физических процессов. 2018. Вып. 1. С. 3–14.
4. Барабанов Р. А., Дьянов Д. Ю., Каныгин И. И. и др. Пакет программ ЛОГОС. Метод решения задач статической прочности тонкостенных и стержневых конструкций на основе SOLID-SHELL-технологии // Там же. 2016. Вып. 3. С. 26–36.
5. Ерёмченко А. Ю., Косарим С. С. Пакет программ ЛОГОС. Методика расчета отклика конструкции при действии широкополосной случайной вибрации // Там же. 2020. Вып. 2. С. 71–85.

6. Simulia ABAQUS [Electronic resource]. – <http://www.3ds.com/ru/produkty-i-uslugi/simulia/produkty/abaqus>.
7. ANSYS Workbench [Electronic resource]. – <http://ansys.com/Products/Platform>.
8. Altair HyperMesh [Electronic resource]. – <http://www.altairhyperworks/product/HyperMesh>.
9. Геометрическое ядро Parasolid [Electronic resource]. – <http://www.plm.automation.siemens.com/global/products/plm-components/parasolid.html>.
10. Геометрическое ядро OpenCascade [Electronic resource]. – <http://www.opencascade.com>.
11. Геометрическое ядро C3D [Электронный ресурс]. – <https://c3dlabs.com/ru/products/c3d-kernel>.
12. Stroud I. Boundary representation modeling technique. – London: Springer-Verlag, 2006.
13. Masuda H., Shimada K., Numao M., Kawabe S. A mathematical theory and applications of non-manifold geometric modeling // Int. symp. on Advanced Geometric Modeling for Engineering Applications. – Amsterdam: F.-L. Krause and H. Jansen, 1990. P. 89–103.
14. Cazier D., Kraemer P. X-maps: an efficient model for non-manifold modeling // IEEE Int. Conf. on Shape Modeling and Applications (SMI). Provence, France, June 21–23, 2010.
15. Zenga L., Liu Y.-J., Lee S. H., Yuena M. M.-F. Q-Complex: Efficient non-manifold boundary representation with inclusion topology // Computer-Aided Design. 2012. Vol. 44. P. 1115–1126.
16. Frey P. J., George P. L. Mesh generation application to finite element. – Oxford, UK: HERMES Science Publishing, 2000.
17. Perronnet A. Interpolation transfinite sur le triangle, le tetraèdre et le pentaèdre. Application à la création de mailages et à la continuation Dirichlet // Comptes Rendus de l'Academie des Sciences. Serie I-matematique. 1998. T. 326. P. 117–122.
18. George J. A. Computer implementation of the finite element method : PhD thesis. Stanford Univ., Dept. of Computer Science, 1971.
19. Lo S. H. A new mesh generation scheme for arbitrary planar domains // Int. J. Numer. Meth. Engin. 1985. Vol. 21. P. 1403–1426.
20. Peraire J., Vahdati M., Morgan K., Zienkiewicz O. C. Adaptive remeshing for compressible flow computations // J. of Comput. Phys. 1987. Vol. 72. P. 449–466.
21. Owen S. J., Staten M. L., Canann S. A., Saigal S. Q-morph: an indirect approach to advancing front quad meshing. International // J. for Numer. Meth. Engin. 1999. Vol. 44(9). P. 1317–1340.
22. Lee K.-Y., Kim I.-I., Cho D.-Y., Kim T.-W. An algorithm for automatic 2D quadrilateral mesh generation with line constraints // Computer-Aided Design. 2003. Vol. 35. P. 1055–1068.
23. Rossignac J. R. Specification, representation, and construction of non-manifold geometric structures. – N.-Y.: IBM, T. J. Watson Research Center Yorktown Heights, 1996.
24. Лазарев В. В. Распараллеливание и оптимизация построения блочных расчетных сеток в препроцессоре пакета программ ЛЮГОС // Вопросы атомной науки и техники. Сер. Матем. моделирование физ. процессов. 2018. Вып. 1. С. 54–63.
25. Генератор сеток CM2 MeshTools [Electronic resource]. – <https://www.computing-objects.com/cm2-meshtools-suite>.
26. The Verdict Library Reference Manual. Elemental Technologies, Inc, 2007 [Electronic resource]. – https://vtk.org/Wiki/images68/VerdictManual_revA.pdf.

**A Method for Automatic Generation of Surface Grids
on Faces of Geometric Models in Parametric Representation
in the LOGOS Software Package Preprocessor**

A. P. Beshpalov, D. V. Loginov, A. N. Solovyev

The paper describes a method of automatically generating surface quadrangular (fully, or predominantly) grids for geometric models in parametric representation. The method has been implemented in the LOGOS software package and is used to solve the strength analysis problems. The key aspects and logic of the surface grid generating algorithms in the LOGOS-Prepost preprocessor are considered.