

АЛГОРИТМ УЧЕТА БЛИЗКИХ ПОВЕРХНОСТЕЙ ПРИ ПОСТРОЕНИИ ПОВЕРХНОСТНЫХ ТРЕУГОЛЬНЫХ СЕТОК ДЛЯ ЗАДАЧ АЭРОГИДРОДИНАМИКИ В ПРЕПОСТПРОЦЕССОРЕ ПАКЕТА ПРОГРАММ ЛОГОС

*Цалко Тарас Валентинович (TVTsalko@vniief.ru),
Панкратов Дмитрий Михайлович (DMPankratov@vniief.ru)*

ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской обл.

В работе рассматривается алгоритм учета близких поверхностей, используемых при построении поверхностной треугольной сетки. Алгоритм учета близких поверхностей позволяет решить проблему близких треугольников, имеющих большие различия в размерах. Из-за этого могут возникать особые случаи. При построении объемной сетки в таких областях будут сформированы ячейки низкого качества (схлопнутые, сильно вытянутые и др). Алгоритм основан на определении близких треугольников посредством построения kd-дерева. Следующим шагом алгоритма производится отбор пар близких треугольников относительно друг друга. Для этого применяется формула, разработанная авторами доклада. Формула также приведена в докладе. После этапа определения пар близких треугольников различных по размеру производится локальное перестроение сетки в областях, содержащих данные треугольники, чтобы минимизировать разницу в размерах между ними.

Ключевые слова: пакет программ ЛОГОС, препостпроцессор, генератор поверхностной треугольной сетки.

ALGORITHM DEVELOPMENT FOR SURFACE PROXIMITY CONTROL DURING SURFACE TRIANGULAR MESH GENERATION FOR AEROHYDRODYNAMICS MODELING IN LOGOS SOFTWARE PACKAGE PREPOSTPROCESSOR

*Tsalko Taras Valentinovich (TVTsalko@vniief.ru),
Pankratov Dmitry Mikhailovich (DMPankratov@vniief.ru)*

FSUE «RFNC-VNIIEF», Sarov Nizhny Novgorod region

The paper describes the surface proximity control algorithm used to generate surface triangular meshes. This algorithm allows solving the problem of near triangles with an exceeding size difference. Some special cases may occur. Low-quality cells (collapsed, much elongated, etc.) will be formed in the volume mesh generation process. The algorithm is based on finding neighboring triangles by means of the kd-tree construction. The next algorithm step is the selection of pairs of closely-spaced triangles. The formula developed by the authors is applied for this purpose and it is presented in the report. After the stage of finding pairs of closely-spaced triangles, local remeshing is performed in the regions containing such triangles to minimize the difference in their sizes.

Key words: LOGOS software package, pre-postprocessor, surface triangular mesh generator.

Введение

Пакет программ инженерного анализа ЛОГОС создан в рамках программы развития суперкомпью-

терных технологий [1]. В настоящее время счетные коды ЛОГОС используются для решения широкого класса задач [2]. Для задания начальных данных и генерации сеток, а также предоставления графиче-

ских интерфейсов подготовки и выполнения инженерного анализа по комплексу программ ЛОГОС ведется разработка препостпроцессора ЛОГОС [3].

Подготовка расчетной сеточной модели для задач аэрогидродинамики в препостпроцессоре ЛОГОС включает в себя следующие этапы:

- построение, импорт и исправление геометрической модели;
- построение поверхностной сеточной модели;
- построение объемной сеточной модели.

Построение поверхностной и объемной сеточной модели в произвольных трехмерных областях является сложной математической задачей.

Для решения этой задачи в препостпроцессоре ЛОГОС для задач аэрогидродинамики разрабатываются автоматические генераторы поверхностных сеток и объемных сеток с многогранными элементами методом отсечения.

Сквозная цепочка построения расчетной сетки предполагает последовательное построение в автоматическом режиме сначала поверхностной, а затем и объемной сетки. Входными данными для генератора поверхностных сеток является исходная геометрическая модель, а для генератора объемных сеток – построенная поверхностная сетка. Поскольку сквозная цепочка не предполагает пользовательского редактирования сеток в процессе построения, то актуальной становится задача подготовки поверхностной сетки высокого качества. Однако использование базовых алгоритмов генерации сетки не всегда позволяет получить сетку высокого качества, поэтому возникает необходимость разработки дополнительных алгоритмов повышения качества поверхностных ячеек.

В докладе представлено описание алгоритма учета близких поверхностей в модуле генерации поверхностных треугольных сеток. Данный алгоритм позволяет увеличить качество построенной поверхностной сетки за счет более плавного изменения размеров треугольников в зонах близких поверхностей. Он включает в себя обнаружение близких по расстоянию, несвязанных топологически между собой, треугольников, а также доработанное перестроение сетки по геометрической модели в фасеточном представлении для построения сетки с заданными параметрами на близких областях.

Постановка задачи

Одной из проблем, встречающихся на ряде геометрических моделей при подготовке сеток для производственных задач, является наличие близких треугольников, имеющих большие различия в размерах, как показано на рис. 1.

В качестве близких треугольников в данном докладе рассматриваются близко расположенные треугольники, имеющие значительные различия в размерах. Для оценки размера треугольника, используется значение длины самого маленького ребра.

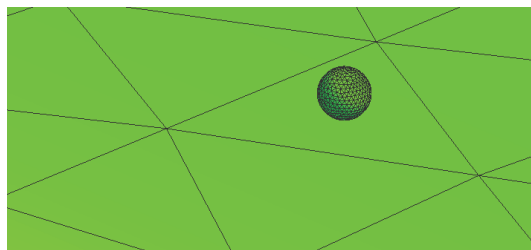


Рис. 1. Близко расположенные треугольники с большой разницей в размерах

Из-за большой разницы в размерах близких треугольников при построении объемной сетки могут возникать особые случаи. При построении объемной сетки в таких областях будут сформированы ячейки низкого качества (схлопнутые, сильно вытянутые и др.).

Основная проблема при поиске близких треугольников состоит в том, что геометрически близкие объекты могут быть топологически не связаны друг с другом, что исключает простую проверку соседства через ребро или вершину.

Для решения задачи поиска близких треугольников были рассмотрены два подхода, описанные в литературе. Первый подход заключается в использовании восьмеричного дерева для поиска областей и переноса размеров в трехмерном пространстве. Узлы дерева (узлами дерева являются ориентированные по осям координат в пространстве параллелепипеды (AABB)) дробятся в соответствии с размерами элементов, которые их пересекают. После этого производится балансировка дерева путем додрабливания соседних узлов, отличающихся больше чем на один уровень дробления. После этого в вершинах сетки записывается минимальный из размеров всех инцидентных ему треугольников и узла дерева, который его содержит. Последующее перестроение происходит с использованием найденных целевых размеров, а исходная сетка используется в качестве фоновой. Второй подход – это явный поиск ближайших треугольников, удовлетворяющих описанному далее критерию близости. Метод прямого перебора всех пар имеет квадратичную сложность и на сетках больших размерностей не применим. Для минимизации времени поиска пар близких треугольников применяется алгоритм, основанный на использовании структуры K-d дерева [4].

В этом подходе перенос размеров выполняется с использованием построения дополнительных ребер, соединяющих ближайшие по расстоянию точки найденных пар треугольников, после чего происходит сглаживание размеров.

Для реализации был выбран второй подход, т. к. он в большей степени подошел под реализованные в модуле генерации структуры данных и разработанную в модуле архитектуру. При этом в ходе тестирования показал приемлемые для решаемых задач результаты.

Поиск близких треугольников

Для определения близких треугольников используются узлы трехмерного K-d дерева, построенного по треугольной сетке. Узлы дерева имеют вид габаритных параллелепипедов, содержащих список треугольников, пересекающих грани этих параллелепипедов или расположенных внутри (рис. 2).

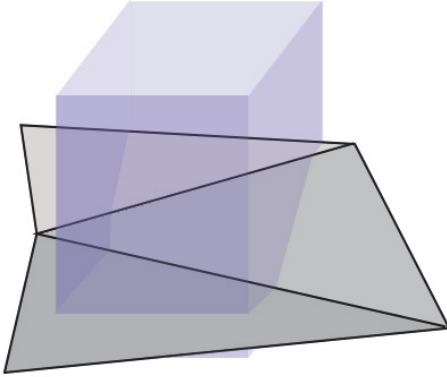


Рис. 2. Схематичное изображение узла дерева (AABB-блока) с входящими в него треугольниками

Отбор треугольников выполняется с помощью оценки расстояний между ближайшими вершинами треугольников.

Для этого рассматриваются пары треугольников. Из центра большего треугольника строится сфера радиуса, равного 1,5 длины наибольшего ребра треугольника, и оценивается попадание в нее геометрического центра меньшего треугольника. Если центр не попал внутрь сферы, то треугольники не считаются близкими.

Дополнительным критерием отбора является выполнение соотношения:

$$h_i = h_0 - L(1 - q), \quad (1)$$

где h_0 – размер наименьшего треугольника; L – расстояние между треугольниками, q – константа (коэффициент роста), h_i – желаемый размер противоположащего треугольника.

Результатом работы алгоритма является список пар близких треугольников, а также соответствующая пара ближайших точек в данных треугольниках (расстояние между которыми минимально).

Кроме того, в генераторе реализована дополнительная возможность опционального исключения пар треугольников, инцидентных через ребро или вершину. Для этого необходимо алгоритму программно задать опцию, определяющую угол (в градусах), используемый в качестве порогового значения для исключения треугольников. Если угол между нормальными пары треугольников меньше заданного порогового значения, то данная пара не рассматривается. Схематично данный критерий показан на рис. 3.

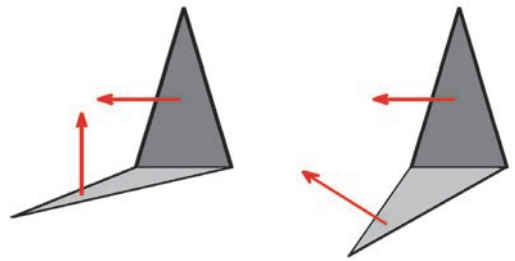


Рис. 3. Отбор соседних треугольников по углу между нормальными

Исходная сетка может не обладать достаточной детализацией для установки требуемого размера в ближайшей вершине треугольника сетки, что негативно скажется на распределении размеров после сглаживания. Для повышения эффективности сглаживания размеров, заданных в вершинах треугольной сетки области, перед выполнением перестроения сетки на области, производится предварительное додрабливание треугольников.

На этапе перестроения в сетку встраиваются точки, лежащие на ребрах и внутри треугольников. Исходными данными является список пар близких треугольников, список пар точек в данных треугольниках, расстояние между которыми минимально, а также размеры треугольников в этих точках. При вставке точки в треугольник производится его разбиение на 3 новых треугольника, как показано на рис. 4. Если точка с некоторой точностью совпадает с одной из вершин треугольника, то она в сетку не добавляется.

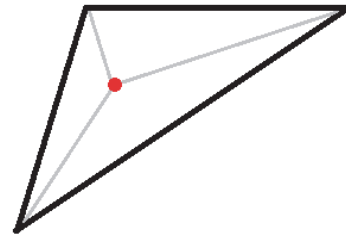


Рис. 4. Схематическое изображение разбиения треугольника точкой

Все добавленные точки проецируются на противоположащие ребра инцидентных треугольников (рис. 5).

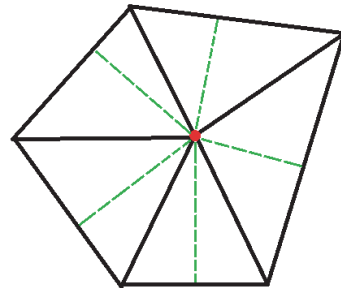
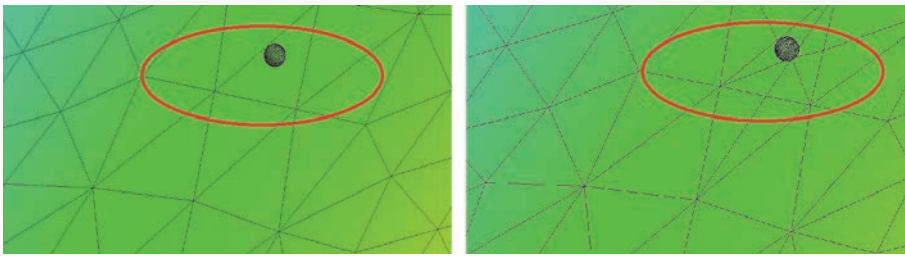


Рис. 5. Проекции вершины треугольника на ребра инцидентных треугольников



а

б

Рис. 6. Работа алгоритма додрабливания: а – исходная сетка, б – результат работы алгоритма додрабливания

Заключительным шагом этапа додрабливания сетки является разбиение ребер. Ребра разбиваются как соответствующими ближайшими точками, так и найденными ранее проекциями вершин (рис. 6).

Следующим этапом является добавление в сетку вспомогательных ребер. Ребра образуются на основе полученных ранее вершин и соответствующих им точек малых треугольников. Пример формирования вспомогательных ребер приведен на рис. 7.

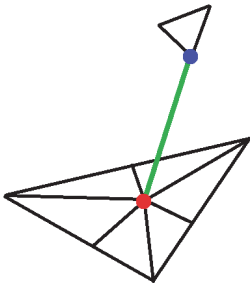


Рис. 7. Построение вспомогательных ребер по парам ближайших точек (линия между точками)

Требуемые размеры в точках вычисляются на основе размеров, заданных в вершинах исходной сетки, с помощью интерполяции барицентрическими координатами.

Далее производится сглаживание размеров в вершинах с использованием алгоритма сглаживания, реализованного в библиотеке генератора поверхностных сеток, после чего вспомогательные ребра удаляются из треугольной сетки.

Для этапа перестроения сетки производится формирование связанных областей, включающих треугольники, полученные на предыдущем этапе, а также треугольники, инцидентные каждой из их вершин.

Перестроение производится при помощи ранее разработанного в модуле генерации поверхностной сетки алгоритма локального перестроения сетки.

Результат перестроенной поверхностной сетки с рис. 6 приведен на рис. 8.

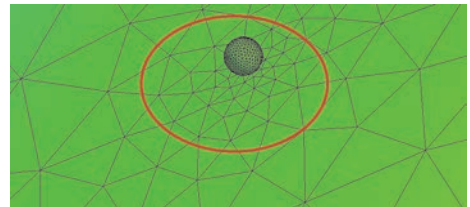
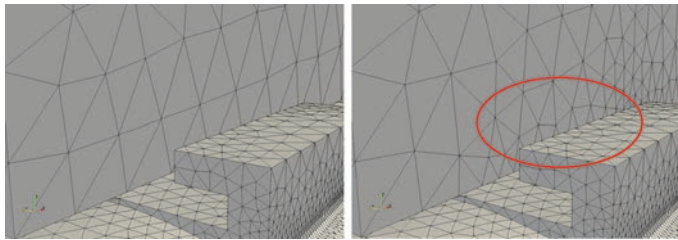


Рис. 8. Сетка с рис. 6 после перестроения

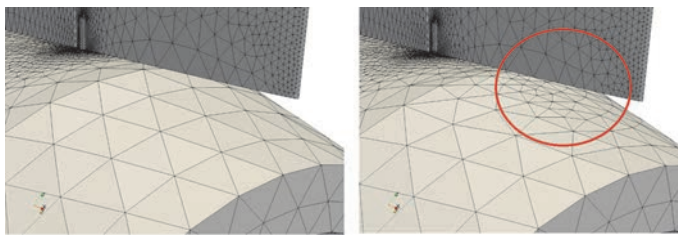
На рис. 9 и 10 представлены результаты работы алгоритмов учета близости поверхностей.



а

б

Рис. 9. Фрагмент поверхностной треугольной сетки: а – построенной без учета близости, б – построенной с учетом близости



а

б

Рис. 10. Фрагмент поверхностной треугольной сетки: а – построенной без учета близости, б – построенной с учетом близости

Реализация описанных алгоритмов проводилась для повышения качества объемной сетки и устойчивости сквозной цепочки построения расчетной сетки в целом.

На рис. 11–14 приведены примеры построения объемной сетки без учета близости поверхностей и с учетом близости поверхностей соответственно.

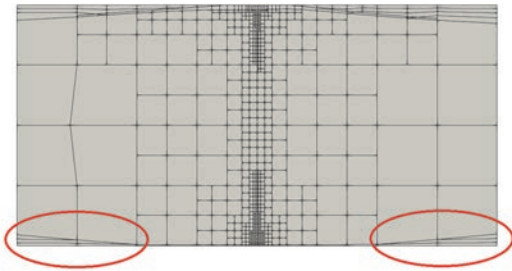


Рис. 11. Сечение объемной сетки, построенной методом отсечения, по поверхностной сетке без учета близости

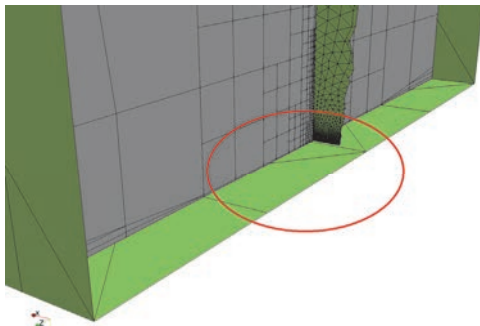


Рис. 12. Треугольники поверхностной сетки, построенной без учета близости

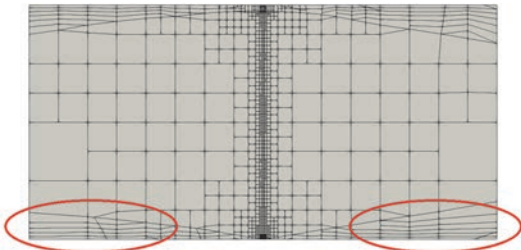


Рис. 13. Сечение объемной сетки, построенной методом отсечения, по поверхностной сетке с учетом близости

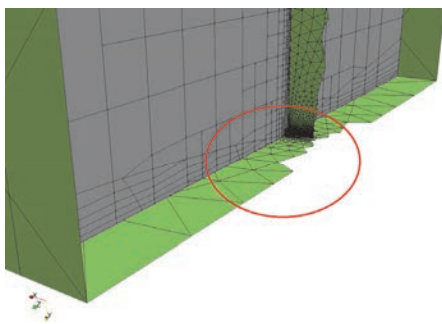


Рис. 14. Треугольники поверхностной сетки, построенной с учетом близости

Заключение

Использование алгоритма учета близких поверхностей в модуле улучшения качества поверхностных сеток позволило сократить требования к исходной геометрической модели, при этом повысив качество поверхностных сеток за счет более детального описания близких областей. Как следствие, это положительно повлияло на качество построения на ее основе расчетной сетки благодаря более точной аппроксимации близких областей.

В результате проведенных работ в генераторе поверхностных сеток были реализованы алгоритмы поиска и учета близких поверхностей на этапе улучшения качества поверхностных треугольных сеток. В качестве доказательства работоспособности алгоритмов приведены результаты их работы для некоторых моделей, а также продемонстрировано улучшение качества сеток на близких участках геометрических моделей.

Включение реализованных алгоритмов в сквозную цепочку подготовки сеточных моделей сократило требования к входной геометрической модели и позволило проводить более детальное построение поверхностных и объемных сеток.

Список литературы

1. Погосян М. А., Савельевских Е. П., Шагалиев Р. М., Козелков А. С., Стрелец Д. Ю., Рябов А. А., Корнев А. В., Дерюгин Ю. Н., Спиридонов В. Ф., Циберев К. В. Применение отечественных суперкомпьютерных технологий для создания перспективных образцов авиационной техники // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2013. Вып. 2. С. 3–17.
2. Сафронов А. В., Дерюгин Ю. Н., Жучков Р. Н., Зеленский Д. К., Саразов А. В., Козелков А. С., Кудимов Н. Ф., Липницкий Ю. М., Панасенко А. В. Результаты валидации многофункционального пакета программ ЛОГОС при решении задач аэродинамики старта и полета ракет-носителей // Математическое моделирование. 2014. Т. 26, № 9. С. 83–95.
3. Фархутдинов В. Ф., Тарасов В. И., Соловьев А. Н., Борисенко О. Н., Лазарев В. В., Логинов Д. В., Аверина Н. С., Марунин А. В., Гриднев А. И., Федякина М. Г., Кузнецов М. Г., Черенкова М. В., Фролова Е. А., Лукичев А. Н., Смолкина Д. Н., Купалова А. Г., Кузьменко М. В., Ховрин Н. А., Сергеева А. С., Попова Н. В., Баканова Т. Ю. Обзор возможностей по обработке геометрических и построению сеточных моделей в ЛОГОС-ПРЕПОСТ // Труды XIV Международной конференции «Супервычисления и математическое моделирование». Саров: ФГУП «РФЯЦ-ВНИИЭФ», 2013. С. 585–592.
4. Moore A. W.: An introductory tutorial on kd-trees // Technical Report No 209 Computer Laboratory, University of Cambridge, Cambridge, England, 1991.