

# Практические аспекты реализации алгоритма построения неструктурированной тетраэдральной сетки проекционным методом\*

С.К.Григорьев, М.В.Якобовский

Институт прикладной математики им. М.В.Келдыша РАН.

Рассматривается параллельный алгоритм и разработанная на его основе программа гарантированного разбиения трехмерных областей с заданной граничной триангуляцией на правильно согласованные тетраэдры. Алгоритм основан на построении треугольных призм, формируемых с помощью ортогонального проецирования треугольников исходной поверхности. Основными преимуществами предложенного алгоритма являются гарантированность заполнения области тетраэдрами и высокая степень внутреннего параллелизма. Рассматриваются возможности сокращения времени работы за счет введения иерархии дополнительных внутренних поверхностей. Основное назначение формируемых сеток: гарантированное выявление топологии трехмерных областей, на основе которой в дальнейшем возможно, например, построение расчетных сеток.

*Ключевые слова:* неструктурированные сетки, генератор расчетных сеток, параллельные алгоритмы, рациональные числа, MPI, триангуляция.

## 1. Введение

При численном моделировании физических процессов методами механики сплошной среды широко используются геометрические расчетные сетки. Вопрос их построения в трехмерных областях сложной формы, заданных поверхностными триангуляциями, является, на сегодняшний день, одним из наиболее сложных и затратных, с вычислительной точки зрения. В дальнейшем, говоря о разбиении пространства на тетраэдры будем, по аналогии с двумерным случаем, употреблять термин «триангуляция». Отметим ряд известных подходов к построению тетраэдральных сеток [1,2,3].

Методы граничной коррекции являются самыми быстрыми из итерационных методов, однако обладают существенными недостатками. Построение сеток в таких методах происходит в два этапа. На первом этапе производится триангуляция некоторой «супер-области», полностью включающей исходную область. На втором этапе все узлы полученной сетки, лежащие вне исходной области, удаляются. При этом узлы, лежащие у границы области, проецируются на саму границу области. Для компенсации геометрических искажений элементов вблизи границ применяют алгоритмы оптимизации сеток. Однако, данный метод плохо применим для областей с заданной триангуляцией границ, что является существенным недостатком [1].

Для триангуляции областей с заданной граничной триангуляцией применяют методы исчерпывания (в англоязычных источниках эти методы имеют название “advancing front”). Отправной точкой для этих методов является граничная триангуляция исходной области, причем не грубая триангуляция, а триангуляция, наиболее соответствующая требованиям разработчика, поскольку в дальнейшем никаких изменений этой триангуляции производиться не будет. На основе каждого из треугольников границы строится отдельный тетраэдр, причем четвертой вершиной тетраэдра служит либо другая уже существующая вершина «фронта», либо дополнительный узел, помещаемый внутрь заданной области. Однако, несмотря на кажущуюся простоту, подобные алгоритмы имеют ряд существенных недостатков. Во-первых, затратность проверки правильности построения каждого нового тетраэдра, так как необходимо удостовериться в отсутствии пересечений с уже существующими тетраэдрами. Во-вторых, и это основная проблема, как правило, в результате работы алгоритма «фронт» формирует области все более сложной формы, которые невозможно триангулировать без добавления дополнительных точек внутрь области. В результате, сходимость процесса оказывается под вопросом, и, более того,

---

\* Работа выполнена при поддержке проекта РФФИ 14-07-00712-а.

размер наименьшего элемента формируемой сетки неконтролируемо уменьшается, что крайне негативно отражается на возможности выполнения на такой сетке численного моделирования [2]. В-третьих, «фронт» может разбиться на несколько не связанных областей.

Так же существуют алгоритмы на основе критерия Делоне. Построение сеток такими алгоритмами производится на основе критерия Делоне для некоторого заданного набора точек. Их достоинством является возможность контроля размеров элементов сетки. Способов размещения узлов достаточно много. Некоторые из них, например метод с названием «упаковка пузырьков», позволяют получить априори высокое качество триангуляции. Недостатком таких методов являются крайне высокая чувствительность к точности машинных вычислений и сложность адаптации к границам области. Операции, используемые в этих методах (нахождение центра и радиуса описанной окружности и т.п.), дают на выходе иррациональные величины, что, в купе с постоянным использованием подобных операций, ведет к неизбежному накоплению ошибок округления [2].

В рамках данной работы рассматривается итерационный алгоритм построения триангуляции трехмерной области, отличный от всех вышеупомянутых. Преимущество рассматриваемого алгоритма состоит в его алгоритмических свойствах. Во-первых, метод является прямым, таким образом, число выполняемых в его рамках операций однозначно определяется сложностью исходной поверхности и может быть с хорошей точностью оценено априори. Во-вторых, метод обеспечивает возможность гарантированного построения трехмерной триангуляции на множестве вершин, координаты которых являются рациональными числами ограниченной разрядности. Важно отметить, что требуемая для проведения вычислений максимальная разрядность линейно зависит только от разрядности представления координат исходных вершин поверхностной триангуляции и не зависит от сложности исходной поверхности. Алгоритм отличается высокой степенью внутреннего параллелизма. Следует отметить два основных недостатка предлагаемого алгоритма. Первый заключается в том, что формирование триангуляции с вершинами, заданными в узлах рациональной решетки, не гарантирует существования топологически непротиворечивой триангуляции возникающей после отображения вершин на решетку, соответствующую машинному представлению вещественных чисел, поскольку разрешение второй решетки ниже, чем первой. Второй, более важный недостаток, заключается в формировании триангуляции низкого, с точки зрения дальнейшего моделирования, качества. Таким образом, основное назначение формируемой триангуляции - предварительное разбиение трехмерной области на тетраэдры, например, с целью выявления топологии области, или для упрощения этапа предварительного размещения узлов в областях сложной формы. Подчеркнем, что ни один из других доступных методов не гарантирует построение непротиворечивой объемной триангуляции за априори оцениваемое конечное время в общем случае.

В рамках работы рассматривается программная реализация алгоритма ортогонального проецирования, его параллельный вариант, а так же метод ускорения вычислений путем добавления иерархии дополнительных внутренних поверхностей.

## 2. Алгоритм ортогонального проецирования

Идея алгоритма состоит в генерации набора треугольных призм заполняющих объем трехмерного тела заданного поверхностной триангуляцией. Поверхность тела задается в виде набора согласованных (примыкающих друг к другу только сторонами или вершинами) ориентированных треугольников. Построение призм осуществляется за счет ортогонального проецирования треугольников исходной поверхности вдоль выбранного направления (рисунок 1.1, 1.2). Полученный в результате такого проецирования двумерный образ трехмерной поверхности триангулируется и образует набор верхних и нижних граней треугольных призм (рисунок 1.3). Боковые грани призм параллельны направлению проецирования, что обеспечивает их автоматическое формирование в результате такого построения. Следующим шагом является согласованная триангуляция боковых стенок построенных призм (рисунок 1.4). Таким образом, на выходе получается набор треугольных призм с согласованной триангуляцией боковых стенок (рисунок 1.5). Задача заполнения треугольной призмы тетраэдрами сложности не представляет. Она решается путем добавления в центр каждой призмы дополнительной вершины, соединении которой с треугольниками поверхности призмы и дает требуемые тетраэдры.

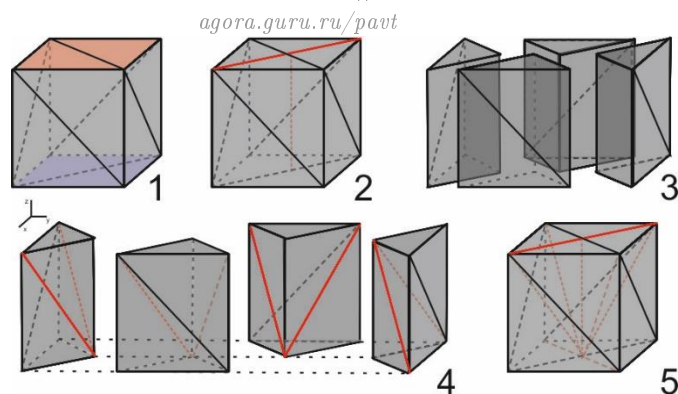


Рис. 1. Алгоритм ортогонального проецирования

В силу того, что для ортогонального проецирования не требуется производить никаких операций, дающих на выходе иррациональные числа, использование арифметики рациональных чисел произвольной разрядности позволяет добиться абсолютной точности в процессе построения сетки, а так же в процессе проверки полученного результата.

### 3. Параллельный алгоритм

При видимой простоте описанного алгоритма его распараллеливание сопряжено с рядом трудностей. Они обусловлены, главным образом, тем, что для получения согласованной сетки требуется обеспечить согласованность триангуляций боковых граней призм, касающихся друг друга целыми гранями или частями граней (здесь и далее такие призмы будут именоваться соседними). Это требование обуславливает необходимость алгоритмической гарантии соответствия триангуляций боковых граней соприкасающихся этими гранями призм. В последовательном случае подобные проблемы решаются за счет построения общей структуры, содержащей все элементы формируемого результата, что позволяет гарантировать согласованности триангуляции для каждой следующей общей боковой грани двух соседних призм. При разработке параллельного алгоритма возникает проблема ненулевой вероятности одновременной обработки одной и той же грани на двух процессорах, при отсутствии гарантии одинаковости результата на выходе алгоритма триангуляции. Одним из способов решения данной проблемы является привлечение алгоритма согласования работы процессов-обработчиков, предотвращающего параллельную обработку двух соседних призм.

В рамках работы реализована параллельная версия алгоритма для систем с распределенной памятью с использованием библиотеки MPI.

Проблемы согласования триангуляции боковых граней решается путем построения графа связей между призмами и запретом одновременной обработки соседних призм. Для предотвращения параллельной обработки боковых граней двух соседних призм используется система с процессом-монитором, который и обеспечивает диспетчеризацию работы процессов-обработчиков, предотвращая тем самым параллельную обработку соседних призм.

Такой подход приводит к тому, что все данные в конечном итоге передаются через один процесс-монитор, а это существенно снижает масштабируемость данной реализации алгоритма. Теоретический предел масштабируемости алгоритма при такой реализации, зависит не столько от сложности используемой сетки (такая зависимость обусловлена различием в объемах передаваемых данных между процессами), сколько от скорости передачи данных между процессом-обработчиком и процессом монитором.

Общая вычислительная сложность составляет в худшем случае  $O(n_{rib}^2)$ , где  $n_{rib}$  – число ребер результирующей сетки.

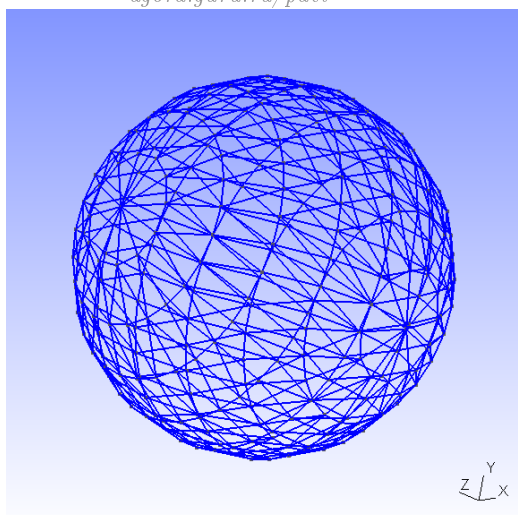


Рис. 2. Поверхность сферы

Эксперимент со сферой (рис. 2) показывает возможность хорошего распараллеливания исходного алгоритма. В случае со сферой хороший параллелизм обусловлен минимальными различиями в масштабах между треугольниками исходной поверхности. Однако известна проблема, не позволяющая осуществить хорошее распределение загрузки между процессами-обработчиками для некоторых исходных конфигураций. Эта проблема возникает на этапе построения ортогональной проекции и связана она с необходимостью построения проекции достаточно сложной двумерной области с наличием элементов существенно разных масштабов, что приводит в конечном итоге и к существенному снижению эффективности распараллеливания и к существенному увеличению времени работы алгоритма, и к существенному увеличению числа операций на обработку одного треугольника. В работе рассматривается один из способов ее решения. Проблема наглядно иллюстрируется конфигурацией представленной на рисунке 3. На этом рисунке показан самолет, представленный большим числом мелких треугольников, и внешние границы расчетной области, представленные небольшим числом треугольников большого размера. Алгоритм предполагает совместную обработку всех треугольников, лежащих под каждым из больших треугольников внешней границы области. Таким образом, в первом приближении, ресурс параллелизма определяется количеством треугольников внешней границы, а вычислительная сложность – количеством треугольников под каждым из внешних треугольников. Соответственно наилучшим условиям соответствует приближенное равенство размеров внешних и внутренних треугольников, а наихудшим (требующим наибольшего времени вычислений) – их существенная разномасштабность. На рисунке 3 как раз показана плохая ситуация, при которой под одним из больших треугольников расположены почти все треугольники описывающие самолет.

#### 4. Иерархия дополнительных поверхностей

Снижению разномасштабности способствует разработанный метод добавления набора пластинок нулевой толщины, вставляемых внутрь заполняемого тела. Пластинки добавляются в том случае, когда при построении очередного элемента проекции, количество покрываемых элементов превышает некоторую величину  $N_{el\_max}$ , значение которой можно варьировать в зависимости от исходных данных. Этот алгоритм целесообразно применять для экранирования тех областей, где расположено наибольшее количество треугольников.

Каждая пластика, представляющая собой четырехугольник или треугольник, располагается над группой из примерно  $N_{el\_max}$  объектов. Группы, над которыми устанавливаются пластины, выбираются из соображений наличия минимального числа связей с остальными элементами. Задача выявления таких групп решается при помощи алгоритма спектральной бисекции [4]. Бисекция применяется к спроецированному набору ребер. Для вычисления собственных значений и собственных векторов спектральной матрицы используется метод редукции Хаусхольдера для исходной матрицы к трехдиагональному виду, после чего уже для трехдиагональной матрицы вычисляются собственные значения и вектора. Для реализации этих алгоритмов при-

меняются процедуры `tred2` и `tqli`, взятые в книге [5]. При этом наиболее ресурсоемким процессом на данный момент является построение проекции, и время, затрачиваемое на построение существенно больше времени произведения матричных вычислений. Сами добавляемые пластинки лежат в плоскости, параллельной плоскости исходного проецируемого треугольника. Над каждой из областей, выделенной алгоритмом бисекции, размещается одна пластинка. Сама пластинка представляет собой уменьшенную копию исходного треугольника. Геометрический центр добавляемой пластинки совпадает с центром закрываемой области. Для определения высоты расположения пластинки ( $Z$ -координаты в нашем случае) в ходе построения проекции определяется точка, ближе всего расположенная к плоскости исходного треугольника. Это расстояние  $h_{max}$  задает область пространства, в которую можно поместить пластинку. На рисунке 3 добавленные пластинки отмечены стрелками.

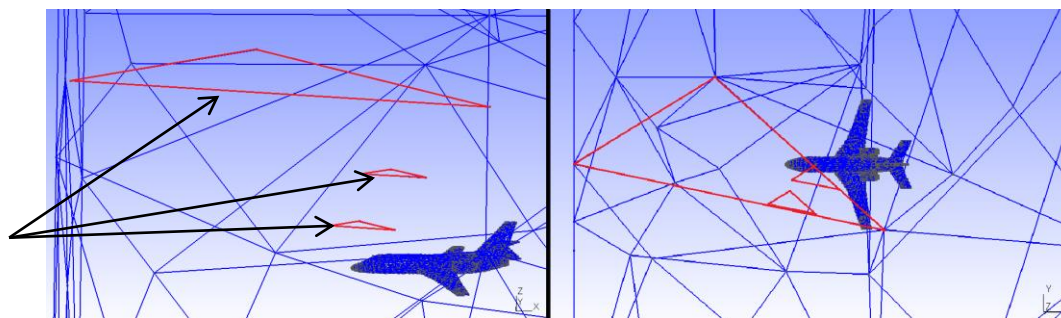


Рис. 3. Пример расположения пластинок над сложной областью

## 5. Выводы

Создание алгоритмов гарантированной генерации расчетной стеки является актуальной задачей. Реализованный в рамках работы параллельный алгоритм обладает высоким показателем внутреннего параллелизма. Использование арифметики рациональных чисел обеспечивает гарантированность получения ответа и значительно упрощает программный код, однако требует решения проблемы перехода от рациональных чисел к числам с плавающей запятой, не рассмотренной в настоящей статье. Автоматическое добавление иерархии пластин нулевой толщины позволяет значительно снизить вычислительную сложность алгоритма.

## Литература

1. Галанин М.П., Щеглов И.А. Разработка и реализация алгоритмов трехмерной триангуляции сложных пространственных областей: прямые методы. Препринт ИПМ.– М.: ИПМ. № 10, 2006.
2. Галанин М.П., Щеглов И.А. Разработка и реализация алгоритмов трехмерной триангуляции сложных пространственных областей: итерационные методы Препринт ИПМ.– М.: ИПМ. № 9, 2006.
3. Скворцов А.В. Триангуляция Делоне и ее применение.– Томск: Томский университет, 2002. 128с.
4. Якововский М.В. Введение в параллельные методы решения задач: Учебное пособие / Предисл.: В. А. Садовничий. – М.: Издательство Московского университета, 2013. – 328 с. – (Серия «Суперкомпьютерное образование»). ISBN 978-5-211-06382-2.
5. Numerical recipes in C : the art of scientific computing / William H. Press. . . [et al.]. – 2nd ed. Includes bibliographical references (p. ) and index. ISBN 0-521-43108-5.

## Practical aspects of realization of projection tetrahedral mesh generation method.

S.K. Grigorjev, M.V. Yakobovskiy

Institute of Applied Mathematics, the Russian Academy of Science

We consider parallel algorithm and developed on his base program of guaranteed discretization of three-dimensional areas on properly coordinated tetrahedron. Algorithm is based on building triangular prisms, formed by orthogonal projection of initial surface triangles. The main advantages of proposing algorithm are guaranteed filling area with tetrahedron and high degree of internal parallelism. We consider possibilities of work time reduction by inserting hierarchy of additional internal surfaces. The main function of farmed meshes consists in guaranteed detection of three-dimensional areas topology, from which it is than possible, for example, construction of computational grids.

*Keywords:* unstructured meshes, mesh generation, parallel algorithms, rational numbers, MPI, triangulation.

### References

1. M.P.Galanin, I.A.Sheglov Development and Implementation of Algorithms for Constrained Volume Triangulations: Direct Methods , Preprint, Inst. Appl. Math., the Russian Academy of Science, №10, 2006.
2. M.P.Galanin, I.A.Sheglov Development and Implementation of Algorithms for Constrained Volume Triangulations: Iterative Algorithms, Preprint, Inst. Appl. Math., the Russian Academy of Science , №9, 2006.
3. Skvorcov A.V. Trianguljacija Delone i ejo primenenie. - Tomsk: Tomsk State University, 2002. 128p.
4. Yakobovskiy M.V.. Vvedenie v parallel'nye metody reshenija zadach: Uchebnoe posobie / introduction: V.A. Sadovnichy. - Moscow: Publishing house of the Moscow University, 2013. - 328 p. - (Series "Supercomputer Education"). ISBN 978-5-211-06382-2.
5. Numerical recipes in C: the art of scientific computing / William H. Press. . . [et al.]. – 2nd ed. Includes bibliographical references (p. ) and index. ISBN 0-521-43108-5.