

ИСПОЛЬЗОВАНИЕ ВСПОМОГАТЕЛЬНОГО МАССИВА УЗЛОВ ПРИ ДИСКРЕТИЗАЦИИ СЛОЖНОЙ ТРЕХМЕРНОЙ ОБЛАСТИ МЕТОДОМ ИСЧЕРПЫВАНИЯ

Рассмотрен класс методов исчерпывания дискретизации трехмерных областей и основные проблемы, связанные с их реализацией. Предложен метод решения некоторых проблем путем использования вспомогательного массива узлов. Приведен пример алгоритма, основанного на этом приеме. Результат работы описанного алгоритма иллюстрирован примером.

Конечно-элементные методы нашли широкое применение при решении различных прикладных задач науки и техники. Использование этих методов предполагает предварительное построение так называемой “конечно-элементной сетки”, т.е. разбиение области, в которой решается задача, на множество непересекающихся конечных элементов заданной формы (тетраэдры, призмы, шестигранники и т.п.). Эта задача является нетривиальной, если область сколько-нибудь геометрически сложна.

Для построения тетраэдрических сеток в сложных областях, как правило, используются методы на основе критерия Делоне или методы исчерпывания [1, 2]. В данной работе рассмотрены методы исчерпывания и связанные с их реализацией проблемы, а также приведен вариант алгоритма метода с использованием массива вспомогательных узлов.

Методы исчерпывания. Впервые идея метода исчерпывания была предложена Рейнальдом Лонером (R. Lohner), а его трехмерный вариант разработал профессор Гонконгского университета С.Х. Ло [3, 4]. Алгоритм Ло вот уже многие годы успешно используется в программном комплексе ANSYS для дискретизации произвольных объемных областей.

Общая идея этого класса методов заключается в последовательном изыпании из заданной области фрагментов треугольной (тетраэдрической) формы до тех пор, пока вся область не окажется исчерпана. Впрочем, этот легко представимый в воображении процесс на самом деле имеет мало общего с практической реализацией. Английское название “advancing front” (что можно перевести как “продвижение фронта”), пожалуй, лучше отражает сущность алгоритмов.

Отправной точкой всех алгоритмов этого класса является некая триангуляция границы заданной области, причем не грубая, а наиболее полно соответствующая требованиям задачи, поскольку в дальнейшем никаких изменений этой триангуляции не будет. Отметим, что подобное требование к начальным данным может быть как положительной стороной метода, так и отрицательной. Если никаких ограничений на границу области не накладываемся, то это приводит к необходимости ее отдельной триангуляции, что само по себе является самостоятельной задачей. С другой стороны, если необходимо обеспечить согласование триангуляции в сложной области, либо же триангуляция поверхности задана изначально, то использование метода исчерпывания будет самым удачным выбором. (В двумерном случае под триангуляцией границы будем понимать разбиение линии границы на отрезки необходимой длины.)

Триангуляция границы является тем самым фронтом, упоминаемым в английском названии. Например, в трехмерном случае, используя какой-либо треугольник из фронта, можно на его основе построить тетраэдр, причем четвертой вершиной тетраэдра может быть либо другая вершина фронта, либо дополнительный узел, помещаемый внутрь заданной области. При изъятии полученного тетраэдра из фронта может быть удалено от одной (случай вставки дополнительного узла) до четырех граней и одновременно добавлено от одной до трех новых граней. Таким образом, текущий фронт дискретизации постепенно продвигается в пространстве — откуда и само название “advancing front”. Обновив данные о фронте, можно вновь изъять тетраэдр, снова обновить фронт и так далее, пока вся область не окажется исчерпана [5–10]. Несмотря на кажущуюся простоту идеи, реализация алгоритма исчерпывания таит в себе ряд трудностей (особенно в трехмерном случае). Рассмотрим пример простого алгоритма исчерпывания для триангуляции двумерных областей.

Пусть задана некоторая область и триангуляция ее границы (рис. 1). Тогда множество всех ребер границы L будет представлять собой фронт (это множество может быть и не односвязным, как в приведенном примере), а концы отрезков фронта — множество граничных узлов G . Алгоритм состоит из следующих шагов:

1. Поиск граничного узла g_1 с наименьшим значением внутреннего (по отношению к исчерпываемой области) угла.
2. Выбор L_i — длиннейшего из двух ребер, инцидентных g_1 . (Второй конец этого ребра обозначим g_2 .)
3. Формирование множества B граничных узлов, находящихся на расстоянии не более $\sqrt{3}\|L_i\|$ от узлов g_1 и g_2 . Для каждого узла $g_3 \in B$

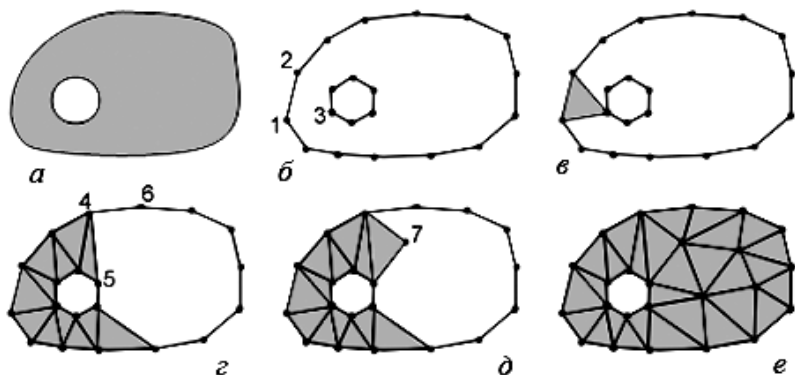


Рис. 1. Пример работы алгоритма исчерпывания в двумерном случае

рассматривается треугольник $T(g_1, g_2, g_3)$ и проверяются следующие условия:

- а) T лежит в исчерпываемой области;
- б) внутри и на границе T нет никаких других узлов триангуляции, кроме узлов, образующих этот треугольник;
- в) T не пересекается никакими ребрами существующей триангуляции;

г) площадь (либо иная размерная характеристика) T не превосходит максимально допустимого условиями задачи значения. Среди всех треугольников T , удовлетворяющих указанным условиям, выбирается вариант с наилучшими аппроксимационными свойствами¹ [1] и осуществляется переход к п. 5. Если же подходящих треугольников не нашлось, то осуществляется переход к п. 4.

4. Поиск в исчерпываемой области точки p , для которой треугольник $T(g_1, g_2, p)$ удовлетворяет всем условиям из п. 3 и при этом обладает лучшими аппроксимационными свойствами (т.е. в идеале на ребре L_i следует построить правильный треугольник). Для нахождения такой точки достаточно восстановить из середины ребра L_i перпендикуляр

(высоту) длиной $h = \min \left(\frac{2S_{\max}}{\|L_i\|}, \frac{\sqrt{3}}{2} \|L_i\| \right)$, где S_{\max} — максимально допустимая площадь элемента.

5. Обновление фронта по следующей схеме: рассматривается каждое ребро сформированного треугольника, и если ребро принадлежит фронту, то оно удаляется из фронта; если же ребро не принадлежит фронту, оно добавляется во фронт.

¹Аппроксимационные качества элемента зависят от его формы. В большинстве методов наилучшими свойствами обладают правильные симплексы (т.е. правильные треугольники и тетраэдры). Рассмотрение вариантов оценок этих свойств выходит за рамки данной статьи.

6. Если область еще не исчерпана, то выполняется п. 1, иначе процесс окончен.

В приведенном на рис. 1 примере на первой итерации алгоритма наименьшим углом обладает узел, обозначенный "1". Согласно алгоритму, выбирается более длинное из двух ребер (1–2), а затем проводится поиск среди граничных узлов, с которыми могут быть образованы подходящие элементы. Из них выбирается узел, позволяющий создать треугольник с наилучшими аппроксимационными свойствами (рис. 1, фигура в). Далее итерации продолжают, пока не реализуется ситуация, иллюстрированная на рис. 1, фигура г. Здесь из фронта выбирается ребро 4–5, однако с соседним узлом 6 элемент образовать нельзя, так как его площадь получится слишком большой, поэтому в область добавляется новый узел 7 (см. рис. 1, фигура д). Далее алгоритм продолжается до тех пор, пока вся область не оказывается исчерпанной (т.е. пока множество ребер фронта L не станет пустым, см. фиг. в на рис. 1).

Заметим, что самой ресурсоемкой частью любого алгоритма исчерпывания является проверка условий п. 3, так как необходимо удостовериться, что новый элемент не пересекается ни с какими уже существующими. Причем на каждой итерации алгоритма эта процедура с различными параметрами может вызываться от 5 до 20 раз (иногда и больше). От того, каким образом выполнена эта проверка, главным образом и зависит производительность алгоритма.

Реализация алгоритма исчерпывания в трехмерном случае таит в себе несколько дополнительных трудностей (помимо очевидно

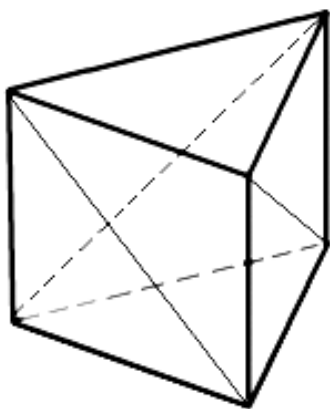


Рис. 2. Пример трехмерной области (пятигранная призма), которую невозможно разбить на тетраэдры без вставки дополнительного узла

большой ресурсоемкости). В первую очередь следует отметить краеугольную проблему трехмерной дискретизации: нередка ситуация, когда в результате работы алгоритма образуются области, которые невозможно дискретизировать, не используя дополнительных внутренних узлов (пример такой ситуации приведен на рис. 2). В двумерном случае этой проблемы нет, так как любой многоугольник на плоскости можно разбить на треугольники одними только внутренними ребрами. Вторая главная сложность связана с выбором текущей грани фронта для построения тетраэдра. Дело в том, что для этого весьма желательно использовать грань, вершинами которой являются узлы с наименьшими значениями

внутренних телесных углов. В то время как вычисление *планарных* и *двугранных* углов тривиально, вычисление *телесных* углов представляет собой куда более трудоемкую задачу (особенно если учесть, что каждый телесный угол в данном случае может быть сформирован различным числом граней). Отметим, однако, что эту проблему можно частично обойти, если на каждом этапе изымать не один тетраэдр, а сразу целый слой. Недостатком этого варианта является довольно существенное снижение качества тетраэдров [7].

Цель данной статьи — предложить достаточно простой и эффективный способ решения этих проблем. Предлагаемый метод основан на использовании вспомогательного массива узлов и иллюстрируется на примере разработанного алгоритма.

Вариант метода исчерпывания с использованием вспомогательного массива узлов. Рассмотрим вариант алгоритма исчерпывания с использованием вспомогательного массива узлов, предназначенный для построения сеток, близких к равномерным, в произвольных областях (предполагается, что граница области уже триангулирована).

Пусть необходимо построить сетку с желаемой средней длиной ребра, равной h (эту величину часто называют шагом триангуляции). В качестве исходных данных алгоритм требует два массива: массив, описывающий триангуляцию границы (фактически, начальный фронт, представляемый в виде списка треугольников-граней) и массив вспомогательных точек F . Этот массив формируется так: заданная область помещается в суперобласть (параллелепипед), которая равномерно заполняется вспомогательными точками F с шагом $h_F = h/n$, $n = 10 \dots 15$ (т.е. плотность их размещения должна быть на порядок больше плотности размещения узлов триангуляции). Отметим, что поскольку координаты этих узлов можно легко вычислять ($r_{ijk} = (ih_F, jh_F, kh_F)$), для этого массива не требуется большого объема. Фактически, множество F будет описываться трехмерным массивом однобитных булевых переменных (TRUE — точка существует, FALSE — точка удалена). Таким образом, хранение восьми миллионов элементов F потребует около одного мегабайта оперативной памяти. После формирования множества F из него удаляются все точки, лежащие вне заданной области. Оставшиеся точки будут представлять своего рода объемное растровое изображение заданной области. Именно с помощью множества F в этом алгоритме и решаются проблемы, рассмотренные выше.

Введем контрольную величину V_{\max} , обозначающую максимальный допустимый объем тетраэдра, и положим ее равной $0,15h^3$ (объем правильного тетраэдра с ребром h плюс допуск 25%). Для оценки аппроксимационных качеств будем использовать отношение объема

тетраэдра к наибольшему из произведений длин тройки ребер, выходящих из одного узла: чем это отношение больше, тем более тетраэдр близок к правильному и тем лучше его аппроксимационные качества.

Алгоритм состоит из следующей последовательности действий:

1. Формируется множество узлов G , состоящее из вершин фронта (т.е. изначально в него входят все существующие узлы сетки); для каждого узла G проводится оценка телесного угла. Эта оценка сводится к нахождению числа существующих (т.е. не удаленных) точек F , находящихся от данного узла на расстоянии не более, чем $\sqrt{2}h$ (что требует существенно меньших затрат ресурсов, чем прямое вычисление телесных углов).

2. Находится вершина $g_1 \in G$ с наименьшим оценочным значением телесного угла и выбирается такая грань (g_1, g_2, g_3) фронта, для которой сумма оценок телесных углов для (g_1, g_2, g_3) минимальна.

3. Формируется множество G_1 узлов G , находящихся от каждой из вершин выбранной грани на расстоянии не более чем $\sqrt{3}h$ (это множество может оказаться и пустым). Для каждого узла $g_4 \in G_1$ рассматривается тетраэдр (g_1, g_2, g_3, g_4) , причем тетраэдр отбрасывается, если не выполняется хотя бы одно из следующих условий:

– строго внутри этого тетраэдра нет ни одной удаленной точки F (существование таких точек на гранях допускается);

– внутри этого тетраэдра нет других существующих узлов сетки, за исключением самих вершин этого тетраэдра;

– этот тетраэдр не пересекается никаким существующим ребром сетки (считается, что тетраэдр пересекается ребром, не являющимся ребром этого тетраэдра, если у него с этим ребром есть хотя бы одна общая точка, не являющаяся вершиной этого тетраэдра)²;

– объем этого тетраэдра не больше максимально допустимого V_{\max} . Из всех тетраэдров (g_1, g_2, g_3, g_4) выбирают тетраэдр наилучшего качества [1] и переходят к п. 5; если же тетраэдров, удовлетворяющих указанным условиям, не оказалось, переходят к п. 4.

4. Находится точка p внутри еще не исчерпанной области, для которой тетраэдр (g_1, g_2, g_3, p) удовлетворяет всем условиям, указанным в п. 3 и в шаре $B(p, 2h/n)$ нет ни одной удаленной точки F (это предотвращает размещение узла p слишком близко от граней и вершин существующих тетраэдров). Вариант алгоритма поиска узла p рассмотрен ниже.

5. Удаляются все вершины F , попавшие внутрь (и на границы) сформированного тетраэдра. Затем фронт обновляется по следующей

²Может показаться, что из условия 3 следует условие 2, но на самом деле это не так. Например, существующий тетраэдр может оказаться целиком внутри проверяемого тетраэдра.

схеме: рассматривается каждая грань сформированного тетраэдра и если эта грань является гранью фронта, то она удаляется из фронта; если же эта грань не является гранью фронта, она добавляется в него.

6. Если остались еще не удаленные точки F или (что эквивалентно) фронт не пуст (т.е. область еще не исчерпана полностью), осуществляется переход к п. 1, иначе процесс окончен.

Таким образом, массив F используется сразу для нескольких целей: для оценки телесного угла, для контроля правильности построения и размещения новых узлов. Также массив F удобно использовать для индикации процесса выполнения. Отношение числа удаленных во время работы алгоритма точек F к начальному числу существующих точек F фактически показывает, какая часть области уже исчерпана.

Вернемся к вопросу нахождения координат нового узла для построения тетраэдра, поставленному в п. 4 описанного алгоритма. Пусть заданы три узла: g_1, g_2, g_3 .

1. На первом шаге необходимо найти \vec{r}_c — радиус-вектор барицентра треугольника (как среднее арифметическое координат его узлов) и единичную нормаль \vec{n} к плоскости грани (через нормированное векторное произведение).

2. Далее определяется первое приближение для расстояния от барицентра до искомой точки p : $H = \min(H_1, H_2)$, где H_1 определяется из максимально допустимого объема тетраэдра: $H_1 = 3V_{\max}/S$ (заметим, что площадь грани S фактически найдена на предыдущем шаге, ведь результат векторного произведения двух ребер как раз равен удвоенной площади грани), а H_2 находится из принципа максимальной приближенности формы тетраэдра к правильной: $H_2 = \sqrt{S}$.

3. Проверяется точка с радиус-вектором $\vec{p} = \vec{r}_c + \vec{n}H$, и если тетраэдр (g_1, g_2, g_3, p) удовлетворяет всем требованиям, переходят к п. 6, иначе — к п. 4.

4. Проверяется точка $\vec{p} = \vec{r}_c - \vec{n}H$, и если тетраэдр (g_1, g_2, g_3, p) удовлетворяет всем требованиям, переходят к п. 6, иначе — к п. 5.

5. Полагают $H := 0,75H$ и переходят к п. 3.

6. Искомый узел найден³.

Пример работы алгоритма. Проиллюстрируем описанный алгоритм примером. Пусть требуется построить тетраэдрическую сетку в области, представляющей собой куб, внутри которой расположено включение в виде произвольно ориентированного куба меньшего размера (данная область моделирует ячейку композитного материала). Возьмем размеры внешнего куба $8 \times 8 \times 8$, размеры внутреннего —

³Заметим, что в 99% случаев искомая точка находится во время 1-й или 2-й итерации данного алгоритма.

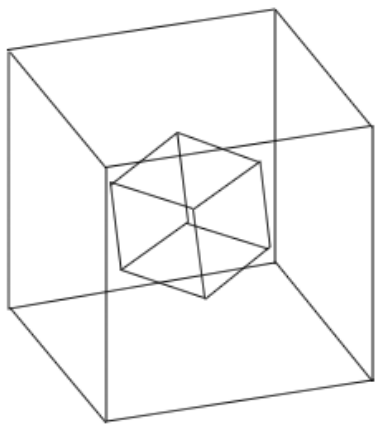


Рис. 3. Область триангуляции

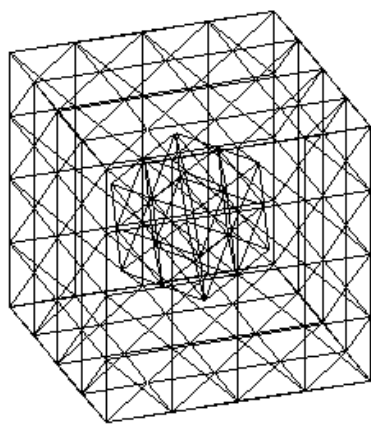


Рис. 4. Начальный фронт

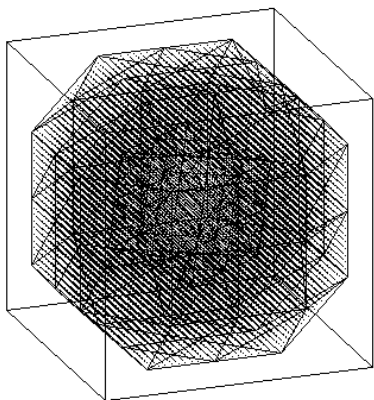


Рис. 5. 100-я итерация метода (показаны только еще не исчерпанная область и фронт)

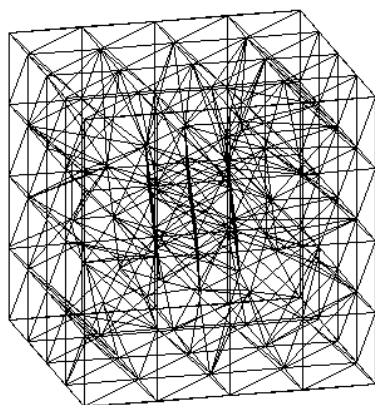


Рис. 6. Построенная сетка (297-я итерация)

$3 \times 3 \times 3$. В углах Эйлера ориентация внутреннего куба может быть описана как $(\pi/4, \pi/6, 0)$ (рис. 3). Требуемый шаг сетки 2. Построим триангуляцию границы области, для чего разобьем грани внешнего и внутреннего кубов на квадраты (со сторонами 2 и 1,5 соответственно), а затем каждый квадрат — на два треугольника (рис. 4). В результате получим начальный фронт (отметим, что он является несвязным). Заполнив пространство между двумя кубами вспомогательными узлами, получим растровое изображение области, которую необходимо триангулировать. На рис. 5 приведено подобное растровое изображение области, но уже на 100-й итерации, т.е. после изъятия 100 тетраэдров. На рис. 6 приведен конечный результат работы алгоритма — тетраэдрическая сетка из 297 тетраэдров.

Выводы. Использование вспомогательного массива узлов позволяет довольно эффективно решать проблемы технического характера, возникающие при реализации алгоритмов исчерпывания, в том числе

нахождение граней с наименьшими телесными углами, определение направления исчерпывания и проверку корректности строящихся тетраэдров. Вместе с тем затраты на реализацию этого вспомогательного массива не велики и его можно использовать в различных вариантах алгоритмов исчерпывания.

Работа выполнена при частичной финансовой поддержке РФФИ (проект № 06-01-00421).

СПИСОК ЛИТЕРАТУРЫ

1. Г а л а н и н М. П., Щ е г л о в И. А. Разработка и реализация алгоритмов трехмерной триангуляции сложных пространственных областей: прямые методы. Препринт ИПМ им. М.В. Келдыша РАН, № 10, 2006. – 32 с.
2. Г а л а н и н М. П., Щ е г л о в И. А. Разработка и реализация алгоритмов трехмерной триангуляции сложных пространственных областей: итерационные методы. Препринт ИПМ им. М.В. Келдыша РАН, № 9, 2006. – 32 с.
3. L o S. H. Volume Discretization into Tetrahedra-I. Verification and Orientation of Boundary Surfaces // Computers and Structures, Pergamon Press, Vol. 39, No 5, pp. 493–500, 1991.
4. L o S. H. Volume Discretization into Tetrahedra-II. 3D Triangulation by Advancing Front Approach // Computers and Structures, Pergamon, Vol. 39, No 5, pp. 501–511, 1991.
5. L o h n e r R. Generation Of Three-Dimensional Unstructured Grids by the Advancing Front Method // Proceedings of the 26th AIAA Aerospace Sciences Meeting, Reno, Nevada, 1988.
6. O w e n S. J. A Survey of Unstructured Mesh Generation Technology // Proceedings of 7th International Meshing Roundtable, pp. 239–269, Dearborn, MI, 1998.
7. P i r z a d e h S. Unstructured Viscous Grid Generation by Advancing-Layers Method // AIAA-93-3453-CP, AIAA, pp. 420-434, 1993.
8. R a s s i n e u x A. Generation and Optimization of Tetrahedral Meshes by Advancing Front Technique // International Journal for Numerical Methods in Engineering, Wiley, Vol. 41, pp. 651–674, 1998.
9. B e r n M., E p p s t e i n D. Mesh Generation and Optimal Triangulation // Computing in Euclidean Geometry, World Scientific Publishing Co., 1995, pp. 23–90.
10. F r e y P. J., B o r o u c h a k i H., G e o r g e P.-L. Delaunay Tetrahedralization Using an Advancing-Front Approach // Proceedings of 5th International Meshing Roundtable, Sandia National Laboratories, pp. 31–46, October 1996.

Статья поступила в редакцию 22.02.2007

Илья Александрович Щеглов родился в 1982 г., окончил МГТУ им. Н.Э. Баумана в 2005 г. Аспирант кафедры “Прикладная математика” МГТУ им. Н.Э. Баумана. Автор трех научных работ в области вычислительной математики.

I. A. Shcheglov (b. 1982) graduated from the Bauman Moscow State Technical University in 2005. Post-graduate of “Applied Mathematics” department of the Bauman Moscow State Technical University. Author of 3 publications in the field of computational mathematics.

