

## ОБ ОДНОМ АЛГОРИТМЕ ГЕНЕРАЦИИ ПОВЕРХНОСТНЫХ СЕТОК

## ABOUT ONE ALGORITHM FOR SURFACE MESH GENERATION

A. Dronov

*Summary.* The article discusses the methods of mesh generation in the context of creating a parallel algorithm for generating an unstructured computational mesh for tasks of modeling physical processes on supercomputers with distributed memory. The idea of an algorithm for the construction of surface meshes for such systems based on the advancing front method with space partitioning into segments and the use of common areas for communication between neighboring processes is proposed. This paper is considered as the first step in the design and implementation of a fully distributed tetrahedral mesh generation algorithm.

*Keywords:* mesh generation, distributed algorithms, distributed memory systems, surface mesh, surface triangulation, advancing front method, domain decomposition.

Дронов Антон Геннадьевич

аспирант, ФГБОУ ВО «МГТУ «СТАНКИН» (г. Москва)  
chesston1t@yandex.ru

*Аннотация.* В статье обсуждаются способы построения расчетных сеток в контексте создания параллельного алгоритма генерации неструктурированной расчетной сетки для задач моделирования физических процессов на суперкомпьютерах с распределенной памятью. Предложена идея алгоритма построения поверхностных сеток для таких систем на основе метода движущегося фронта с разбиением пространства на сегменты и использованием общих зон для коммуникации между соседними процессами. Данная работа рассматривается в качестве первого этапа проектирования и реализации полностью распределенного алгоритма построения тетраэдральных сеток.

*Ключевые слова:* построение сеток, распределенные алгоритмы, системы с распределенной памятью, поверхностная сетка, триангуляция поверхности, метод движущегося фронта, разбиение пространства.

## Введение

Моделирование является важным инструментом для исследования различных явлений и процессов. С его помощью можно изучить поведение системы или процесса в определенных заданных условиях, определить оптимальные параметры, а также проверить интересующие предположения и выдвинуть гипотезы.

Моделирование производится на замещающих реальные объекты прототипах, и в объяснение этому можно выделить несколько ключевых факторов. Во-первых, создание реальной физической модели может быть дорогостоящим и трудоемким процессом, а в некоторых случаях и вовсе невозможным. Во-вторых, они могут быть достаточно сложными и времязатратными для изучения с точки зрения системы; при использовании моделей мы можем абстрагироваться от определенных её компонентов, изучая только необходимые с определенного ракурса, затем при необходимости меняя фокус наблюдения. В-третьих, использование моделей позволяет изменять исходные параметры исследования и повторять вышеупомянутый процесс. Многие программы для моделирования используют геометрические модели, построенные с помощью систем автоматизированного проектирования (САПР).

В математическом моделировании широкое распространение получили расчеты на суперкомпьютерах для изучения математического описания реального объекта

для предсказания и анализа его поведения. В то же время вычислительные эксперименты над моделями зачастую опираются на численные методы, которые находят решения приблизительно с некоторой точностью. Первым этапом численного моделирования является дискретизация пространства, в частности, построение расчетной сетки. В трехмерном пространстве сетки состоят из узлов, ребер, граней и ячеек. От качества построенной сетки зависит решение поставленной задачи, при этом задача определения критериев качества и их оптимальности является сложной [1].

## Виды сеток

Обычно подразделяют сетки на структурированные (регулярные) и неструктурированные (нерегулярные) [2,3], иногда к этой классификации добавляют блочно-структурированные, или гибридные [4,5]. Структурированными называют сетки, где каждая точка уникально определяется индексами и соответствующими координатами в пространстве, что является их главным преимуществом, поскольку позволяет быстро находить соседние элементы добавлением или вычитанием единицы из какого-то из индексов [3]. Однако построение таких сеток есть трудоемкая задача в сложных областях, благодаря чему неструктурированные сетки являются более популярными, поскольку представляют большую гибкость за счет использования сущностей произвольной формы. В то же время, выбор такого подхода усложняет доступ к данным и вынуждает разработчика вводить дополнительные структуры для их хранения [6]. Блоч-

но-структурированные сетки формируют структурированные блоки в геометрии, соединяя их в общую нерегулярную структуру [4].

Также можно выделить разделение по наличию или отсутствию конформности — данное свойство тесно связано с понятием смежности элементов. В трехмерном пространстве сетка является конформной, если ни одна грань не имеет более двух смежных элементов, и не является в обратном случае.

Наконец, разделяют сетки по форме лежащих в основании фигур. В двумерном пространстве выделяют треугольные и четырехугольные сетки, а в трехмерном, несмотря на имеющееся многообразие (треугольные призмы, четырехугольные пирамиды и т.д.), в основном выделяют тетраэдральные и гексаэдральные сетки.

В работе был выбран подход с неструктурированными сетками. Основные методами построения треугольных неструктурированных сеток в двумерном пространстве являются триангуляция Делоне и метод движущегося фронта.

Триангуляция Делоне основана на одноименном критерии, в рамках которого внутри окружности, описанной вокруг любого построенного треугольника, не должны попадать другие задействованные в этой триангуляции точки. Сам критерий тесно связан с диаграммой Вороного, разбиением плоскости для заданного дискретного множества узлов таким образом, что в каждой области разбиения, соответствующей определенному узлу, любая точка была бы ближе к этому узлу, чем к любому другому; при соединении расположенных таким образом узлов соседних областей ребрами получается триангуляция Делоне.

В реальных задачах преимущественно входными данными служит не набор точек, а некая выпуклая область (или вовсе уже построенная триангуляция, не обязательно соответствующая условию Делоне); в таком случае дополнительную важность принимает операция добавления точек в исходную область для более подробного разбиения. Критерий Делоне для построения триангуляции впервые был использован в [7, 8]. Более подробную информацию о ней можно найти в [9].

Метод движущегося фронта оперируют понятиями начального и текущего фронтов. В качестве начального фронта для каждой грани исходного объекта берутся все отрезки из двух вершин, полученных дискретизацией каждого из её ребер; этот фронт в первоначальный момент времени является и текущим. Среди всех выбирается один из отрезков в соответствии с заранее выбранным критерием (например, наименьший по длине). После этого определяется положение третьей вер-

шины для формирования треугольника, которой будет либо новая, либо уже существующая точка, после чего текущий фронт обновляется, постепенно приходя к центральной части пространства от границ; операция продолжается до тех пор, пока текущий фронт не станет пустым — в таком случае можно сказать, что вся область покрыта триангуляцией.

Большинство из методов построения треугольных сеток расширяются на случай трехмерного пространства [5], при этом построение сеток тетраэдральных происходит в 2 этапа: на первом строится поверхностная сетка на границе области, а уже затем тетраэдральная сетка внутри области [10]. Существуют алгоритмы, где используется комбинация этих методов как в двумерном [11], так и трехмерном пространствах [10].

Методы построения четырехугольных сеток можно разделить на две категории: прямые, где поверхность сразу заполняется четырехугольными объектами, и непрямые, где они получаются трансформацией из сначала созданных других фигур, в большинстве случаев треугольников [5]. Прямые методы, как правило, дают более качественную сетку, однако, для обработки сложных областей зачастую требуется вмешательство пользователя [12]. Непрямые методы являются более универсальными, и поэтому получили широкое распространение [13], в частности, наиболее популярным является метод Q-Morph [14], та или иная его реализация легла в основу ряда работ [13, 15, 16].

При этом адаптация алгоритмов при переходе в трехмерное пространство для построения гексаэдров не является настолько тривиальной задачей, как в случае триангуляций. Основные подходы к их построению представлены в [5].

### Параллелизация процесса построения сеток

С учетом постоянного роста вычислительных мощностей встает вопрос о более эффективном их использовании, чтобы снизить временные рамки простоя оборудования и потери производительности. Использование параллельных алгоритмов является одним из способов достижения этой задачи.

В системах с общей памятью потоки и процессы могут читать и изменять данные в одной и той же области памяти, в то время как каждый узел систем с распределенной памятью имеет собственное адресное пространство и обменивается данными через операции обмена. Большую популярность набрали вычислительные кластеры и суперкомпьютеры, которые используют системы с распределенной памятью как один из способов горизонтального масштабирования.

В научно-исследовательских проектах зачастую для более точных исследований используются достаточно большие по размеру сетки; программный комплекс MARPLE [17], решающий задачи механики сплошных сред и в дополнение к которому разрабатывается данный проект, не является исключением: здесь предусмотрена возможность проведения расчетов в параллельном режиме, а храниться сетка может в нескольких файлах, не обязательно расположенных в памяти одного вычислительного узла. При этом, существует проблема построения таких сеток и их разбиения между процессами. Традиционным подходом к решению такой задачи является генерация сетки последовательным алгоритмом с последующим разбиением на сегменты с помощью программных средств, например, METIS [18]. В то же время, в случае действительно большой по размеру сетки существует проблема её хранения на одном вычислительном узле, поскольку она может не помещаться целиком в его память. Кроме того, такой подход приводит к принципиальной нераспараллеливаемости операций построения и разбиения сетки, что, согласно закону Амдала, ограничивает эффективность параллельных вычислений при решении задачи в целом.

Суммируя все вышесказанное, отметим, что напрашивается идея параллельной, или во всяком случае распределенной по памяти генерации сеток, которая на выходе дает некоторое количество частей для параллельной обработки. Такое первоначальное разбиение может оказаться далеким от оптимального, и может потребоваться дальнейшее переразбиение. В настоящей работе обсуждается идея параллельного алгоритма построения треугольных сеток с планируемым последующим выходом к тетраэдральной форме.

В основе существующих алгоритмов построения трехмерных дискретизаций преимущественно лежат заранее построенные либо поверхностные, либо грубые объемные сетки.

В статье [19] область исходной поверхностной сетки в начале разделяется в пространстве с помощью структуры октодеревя, после чего сегменты идентифицируются и строится сетка параллельно с использованием метода движущегося фронта и стандарта OpenMP. Впоследствии алгоритм был адаптирован [20] для использования на системах с распределенной памятью с использованием гибрида MPI + OpenMP.

Работа [21] также использует октодеревья для декомпозиции областей, однако используется один из методов триангуляции Делоне «разделяй и властвуй».

Алгоритм [22] для входных данных в виде треугольной поверхностной сетки формирует области для дальнейшего независимого параллельного построения объ-

емной сетки на MPI процессах с помощью программ TRIANGLE [23] и TetGen [24] для двумерной и трехмерной триангуляций соответственно.

Наконец, в статье [25] стартовая грубая объемная сетка разбивается на несколько сегментов с помощью METIS [18], которые затем обрабатываются отдельно друг от друга на MPI процессах с уточнением поверхностной сеток каждого сегмента, после чего строится объемные сетки, впоследствии объединяемые в единое целое.

В этой работе автор сосредоточился на проектировании алгоритма для систем с распределенной памятью, в качестве входных данных получающего представление геометрической модели без какой-либо построенной заранее сеточной структуры (поверхностной или объемной).

### Концепция алгоритма

Пусть геометрия расчетной области задана в виде граничного представления, т.е. в виде криволинейных граней и ребер, а также вершин. Также задана функция желаемого шага сетки вида  $h = h(x, y, z)$ . Требуется построить триангуляцию (тетраэдризацию) расчетной области таким образом, чтобы вершины граничных ребер и граней сетки лежали бы на соответствующих ребрах и гранях расчетной области, треугольники (тетраэдры) имели бы приемлемое качество, а длина ребра сетки приблизительно соответствовала бы заданному шагу  $h$  в этой точке. Предполагается, что характерные размеры области, коими являются радиусы кривизны криволинейных границ и расстояния между границами в узких местах, всегда много больше, чем шаг в соответствующем месте, а задающая функция  $h$  не меняется слишком быстро.

Также задается количество разбиений  $N$  с помощью плоскостей, параллельных  $XZ$  или  $YZ$ , в соответствии с которыми необходимо разделить объект на заданное количество сегментов. Соответственно, итоговое число разбиений задается количеством используемых в ходе выполнения алгоритма подпроцессов.

Первым шагом алгоритма является определения ограничивающего пространства сетки (bounding box), куда попадают все элементы геометрии исходного объекта. Данное действие выполняется строго главным подпроцессом программы. Затем необходимо разделить пространство на  $N$  частей, которые представляют из себя параллелепипеды, задающимися парами точек, одна из которых отражает положение левого нижнего угла области, а другая — правого верхнего угла. Для согласования построения триангуляций на границах разбиения между соседними сегментами вводится понятие общей зоны, которая формируется при помощи функции  $d = d(x, y, z)$ . Построение плоскостей с отступом  $d$  по обе

стороны от плоскости разбиения ограничивает геометрическую область, которая задает общую зону между соседними процессами. По сути, у каждого подпроцесса есть собственная область ответственности, где он может формировать объекты, но в случае, если в соответствии с его критериями следующий должен находиться в общей зоне, то необходимо сначала согласовать данный шаг с подпроцессом, делящим с ним эту совместную область. Таким образом, при генерации сущности в пределах общей зоны подпроцесс должен убедиться у соседа в том, что ее положение является удовлетворительным для дальнейшей корректной работы.

После этого, отступая от каждой оси разбиения в обе стороны на расстояние  $dh$ , определяются общие зоны между соседними подпроцессами. Данный этап также полностью выполняется корневым процессом, который распределяет полученные сегменты и общие зоны по процессам, забирая в свое распоряжение область с наименьшими координатами, а сегменту с наибольшим номером — область с наибольшими координатами соответственно. Пример разбиения представлен на Рис. 1.

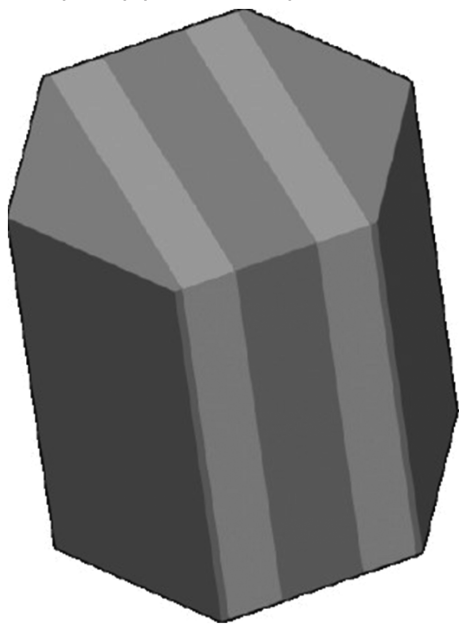


Рис. 1 Разбиение исходного пространства на 3 сегмента, соседние разделены общей зоной

На данный момент выбор разбиений ограничен одной плоскостью (или осью для двумерного пространства) для того, чтобы избежать ситуаций, когда подпроцесс может получить область для построения сетки, при этом ни одно из ребер модели не проходит сквозь нее напрямую.

На следующем шаге начинают работать уже все процессы, причем делают это параллельно, за исключением случаев, описанных выше. Каждый процесс умеет работать с файлом исходного представления геометрии (при

распределенных системах это означает, что на каждом вычислительном устройстве содержится копия входного файла), в то же время составляющие её части элементы различных размерности (узлы, грани, ребра) пронумерованы идентичным образом.

Алгоритм базируется на методе движущегося фронта. Первым делом процессы начинают с дискретизации вершин. Основная идея заключается в том, чтобы определить, принадлежит ли текущая вершина сегменту, выделенному процессу; в случае, если принадлежит, следует добавить ее в список узлов и текущий фронт. Действие повторяется для всех существующих вершин геометрической модели.

Затем происходит дискретизация ребер. Процессы, не имеющие узлов активного фронта, ожидают получения узлов от соседних процессов, пока те, в свою очередь, отступая на расстояние  $h$  относительно координат точки текущего фронта, формируют новый узел сетки и ребро, соединив эти узлы. Вновь добавленный узел становится фронтальным узлом, а прежний из фронта удаляется.

Как только следующей узел попадает в общую зону с одним из соседних процессов, происходит сеанс обмена, в котором процесс сообщает его координаты, а также номер ребра и значение параметра, определяющее узел на нем, после чего исключает его из фронта. После передачи данных процесс прекращает дискретизацию текущего ребра, отдавая дальнейшую работу в этом направлении на откуп своему соседу. Операция также продолжается до тех пор, пока множество фронтальных узлов у всех процессов не станет пустым и не будут обработаны все ребра.

Наконец, завершает цикл процесс дискретизации граней, которые являются представлением поверхностей. Все записанные на предыдущем этапе ребра записываются в активный фронт, на каждом ребре строится треугольник в соответствии с выбранным критерием (например, строится равнобедренный треугольник с известной длиной боковой стороны), новые ребра, сформированные треугольником, добавляются в активный фронт, в то время как основание треугольника из него исключается.

Важным моментом является то, что при выборе положения третьей вершины треугольника может оказаться, что на самом деле она является близкой по расположению к уже существующей точке (например, попадает в область окружности с центром в этой самой точке). В таком случае в качестве вершины выбирается самая близкая точка; в случае, если сторона получившегося треугольника находится в активном фронте, она отсюда исключается.



Если же новая точка, формирующая третью вершину, попадет в область общей зоны, то необходимо сообщить об этом расположении соседу и получить его одобрение на ее размещение в данном месте. В случае, если соседний процесс не имеет никаких противоречий, он направляет подтверждение в качестве ответа, и добавляет данную вершину в свой набор сущностей.

При ситуации, когда противоречия все же имеются, необходимо, помимо отрицательного ответа, предложить самую близкую вершину, имеющуюся в его пространстве. Поскольку такая точка может оказаться даже не в общей зоне, а в области ответственности только другого процесса, автором рекомендуется аккуратно задавать параметры построения сеток, в частности, чтобы ширина общей зоны, как минимум, превышала длину шага сетки.

Работа с модифицированным фронтом продолжается в цикле до тех пор, пока множество фронтальных ребер не станет пустым у всех процессов.

Большую значимость на всех этапах имеет коммуникация между процессами. Помимо необходимости синхронизации процессов после выполнения каждого этапа дискретизации, важно, чтобы во время этих процессов сеансы обмена происходили как можно чаще. В частности, в случае попадания узла в общую зону ожидается подтверждение от процесса-соседа, и долгое ожидание приведет к существенной потере производительности.

Автором предлагается подход, связанный с созданием на каждом процессе потоков, использующих общую память. В соответствии с описанной архитектурой, их

количество должно равняться количеству соседей у текущего процесса, по потоку на взаимодействие с конкретным соседом. Таким образом, если исходная модель разбивается на 4 сегмента, то 1 и 4 процесс выделяют по одному потоку для коммуникации с процессами 2 и 3 соответственно, в то время, как процессы 2 и 3 выделяют по 2 потока (для 1–3 и 2–4 соответственно). Для того, чтобы информация, передаваемые через эти потоки была актуальной, возможно использование различных примитивов синхронизации, таких, как семафор, мьютекс, критические секции и т.д.

В качестве заключительного этапа можно рассматривать процесс оптимизации расположения узлов, например, эллиптическую оптимизацию, когда при сохранении структуры сетки новые координаты узлов находятся в результате минимизации некоторого функционала.

### Заключение

Рассмотрены вопросы построения расчетных сеток, их классификация. Обсуждены способы увеличения производительности программ и использования мощностей современных программных комплексов с помощью параллельных вычислений. Предложена идея алгоритма построения поверхностных сеток для систем с распределенной памятью на основе метода движущегося фронта с разбиением пространства на сегменты и использованием общих зон для коммуникации между соседними процессами. Учитывая, что построение тетраэдральных этапов состоит из двух этапов, данная работа рассматривается в качестве первого этапа проектирования и реализации полностью распределенного алгоритма построения тетраэдральных сеток.

### ЛИТЕРАТУРА

- Ильин В.П. Математическое моделирование. Часть 1. Непрерывные и дискретные модели / В. П. Ильин. — Новосибирск: Изд-во Сиб. Отделения РАН, 2017. 2017. — 428 с.
- Berg M. Computational Geometry. / M. Berg, O. Cheong, M. Kreveld, M. Overmars — Springer, 2008. — 386 p.
- Blazek J. Computational Fluid Dynamics: Principles and Applications (Third Edition). — Butterworth-Heinemann, 2015. — 447 p.
- Bern M., Plassmann P. Mesh generation // In J.-R. Sack and J. Urrutia (Eds.), Handbook of Computational Geometry. — Elsevier Science, 1999. — P. 291–332.
- Owen S.J. A survey of unstructured mesh generation technology // In Proc. of International Meshing Roundtable, 1998. — P. 239–267.
- Маврин А.С. Типы расчетных сеток и способы хранения информации о них [Электронный ресурс] / А.С. Маврин // Научное обозрение: электрон. журн. — 2018. — № 3. — 1 электрон. опт. диск (CD-ROM). — Систем. требования: Pentium III, процессор с тактовой частотой 800 МГц; 128 Мб; 10 Мб; Windows XP/Vista/7/8/10; Acrobat 6 x.
- Lowson C.L. Software for C1 Surface Interpolation // Mathematical Software III (J.R. Rice ed.). — New York: Academic Press, 1977. — P. 161–194.
- Watson D.F. Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes // The Computer Journal. — 1981. — Vol. 24. — No. 2. — P. 167–172.
- Скворцов А.В., Мирза Н.С. Алгоритмы построения и анализа триангуляции. — Томск: Изд-во Том. ун-та, 2006. — 168 с.
- Данилов А.А. Технология построения неструктурированных сеток и монотонная дискретизация уравнения диффузии: диссертация кандидата физико-математических наук: 05.13.18 — Москва, 2010.
- An Improved Advancing-front-Delaunay Method for Triangular Mesh Generation. / Y. Guo, X. Huang, Z. Ma, Y. Hai, R. Zhao, K. Sun // In: Advances in Computer Graphics: 38th Computer Graphics International Conference. — Springer, Cham, 2021. — P. 477–487.
- Araújo C., Celes W. Quadrilateral Mesh Generation with Deferred Constraint Insertion. // Procedia Engineering. — 2014. — Vol. 82. — P. 88–100.

13. Генератор поверхностных неструктурированных четырехугольных сеток в препроцессоре пакета программ «Логос» / Е.Ю. Арапова, В.Г. Куделькин, Е.А. Павлов, С.Ю. Полякова, А.В. Тихонов // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. — 2020. — Вып. 4. — С. 72–85.
14. Advancing front quadrilateral meshing using triangle transformations. / S.J. Owen, M.L. Staten, S.A. Cannan, S. Saigal // Proc. 7th Int. Meshing Roundtable. — Dearborn, Michigan, Sandia National Labs, 1998. — P. 409–428.
15. Беспалов А.П., Логинов Д.В., Соловьёв А.Н. Метод автоматического построения поверхностных сеток на гранях геометрических моделей в параметрическом представлении в препроцессоре пакета программ «Логос» // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. — 2020. — Вып. 3. — С. 23–39.
16. Сковпень А.В. Усовершенствованный алгоритм построения нерегулярных четырехугольных сеток // Ж. вычисл. матем. и матем. физ. — 2005. — Т.45, №8. — С. 1506–1528.
17. MARPLE: программное обеспечение для мультифизического моделирования в задачах сплошных сред / В.А. Гасилов [и др.] // Препринты ИПМ им. М.В. Келдыша. 2023. № 37. 40 с. <https://doi.org/10.20948/prepr-2023-37>, <https://library.keldysh.ru/preprint.asp?id=2023-37>.
18. Karypis, G., Kumar V. METIS — Unstructured Graph Partitioning and Sparse Matrix System, Version 2.0. — Technical Report, Department of Computer Science, University of Minnesota. — 1995.
19. Löhner R. A parallel advancing front grid generation scheme. // Int. J. Numer. Meth. Eng. — 2001. — Vol. 51 (6). — P. 663–678.
20. Löhner, R. A 2nd Generation Parallel Advancing Front Grid Generator. // In: Jiao, X., Weill, J.C. (eds) Proceedings of the 21st International Meshing Roundtable. — Springer Berlin Heidelberg, 2013. — P. 457–474.
21. De Cougny H.L., Shephard M.S. Parallel Volume Meshing Using Face Removals and Hierarchical Repartitioning. // Comp. Meth. Appl. Mech. Eng. — 1999. — Vol. 174 (3–4). — P. 275–298.
22. Автоматическое параллельное построение тетраэдральных сеток с помощью декомпозиции расчетной области / Х. Андрэ, О.Н. Глущенко, Е.Г. Иванов, А.Н. Кудрявцев // Ж. вычисл. матем. и матем. физ. — 2008. — Т.48, №8. — С. 1448–1457.
23. Shewchuk J.R. Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator. // Proceedings first workshop on Applied Computational Geometry, 1996. — P. 124–133.
24. Si H., Gaertner K. Meshing Piecewise Linear Complexes by Constrained Delaunay Tetrahedralizations. // Proceeding of the 14th International Meshing Roundtable. — 2005. — P. 147–163.
25. Parallel unstructured mesh generation by an advancing front method. / Y. Ito, A. Shih, A. Erukala et al. // Math Comput Simul. — 2007. — Vol. 75 (5–6). — P. 200–209.

© Дронов Антон Геннадьевич (chesston1t@yandex.ru)

Журнал «Современная наука: актуальные проблемы теории и практики»