

УДК: 004.925.8

Метод построения неструктурированных шестигранных сеток из объемных данных

А. С. Караваев^а, С. П. Копысов^б

Институт механики УрО РАН,
Россия, 426067, г. Ижевск, ул. Т. Барамзиной, 34

E-mail: ^а karavaev-alexander@yandex.ru, ^б s.kopysov@gmail.com

Получено 22 января 2013 г.

Разработан метод преобразования поверхностного представления расчетной области в формат воксельных данных. Предложен алгоритм генерации расчетной шестигранной сетки на основе объемного формата данных.

Ключевые слова: неструктурированная шестигранная сетка, объемные данные, воксель, знакоопределенные поля расстояний, двойственное контурирование

The method of unstructured hexahedral mesh generation from volumetric data

A. S. Karavaev, S. P. Kopysov

Institute of Mechanics UB RAS 34 ul. T. Baramzinoy, Izhevsk, Russia 426067

Abstract. — The method of conversion of surface representation to the voxel data is developed. Voxel data based algorithm of hexahedral mesh generation is proposed.

Keywords: hexahedral mesh, volumetric data, voxel, signed distance fields, dual contouring

Citation: *Computer Research and Modeling*, 2013, vol. 5, no. 1, pp. 11–24 (Russian).

Работа выполнена при поддержке РФФИ (проект № 12-07-31114)

Введение

Существует несколько способов представления объектов в 3D-пространстве. Все их можно разделить на две группы: поверхностное, определяющее тело путем задания ограничивающей его поверхности, и объемное, содержащее всю информацию о внутренних точках объекта в пространстве.

В настоящее время поверхностное представление является распространенным способом задания геометрии расчетной области. Однако в приложениях, где рассматриваются структуры сложной формы (например моделирование в биомеханике, механике композиционных материалов) и когда возникает необходимость создания модели с внутренней структурой, использование воксельного представления данных является более предпочтительным. Также такой вид представления используется, когда геометрическая модель имеет неявное представление границы (фронт пламени, разделы фаз и т. д.). Для воксельных моделей (например медицинских данных со сканера магнитно-резонансной томографии) реализуется вывод любого сечения модели, что позволяет изучать любой срез данных.

Во многом алгоритмы построения расчетной сетки зависят от того, какой способ представления исходных данных они используют. Этим определяется граница применимости метода, точность и скорость работы, сложность практической реализации.

Разработано достаточное количество методов построения неструктурированных шестигранных сеток, большинство из них обладают небольшой степенью надежности, ограничением, накладываемым на применимость к определенному виду областей. Данные алгоритмы можно разделить на две группы: прямые методы — когда построение шестигранной сетки идет непосредственно внутри заданной границы; непрямые методы — когда чаще всего используется предварительно построенная тетраэдральная сетка с последующим преобразованием тетраэдральных ячеек в шестигранные либо осуществляется преобразование другого исходного дискретного представления области. Как правило, при построении расчетной сетки в качестве исходного описания области используют поверхностное представление (CAD-модель).

Среди прямых методов можно выделить методы регулярной сетки или методы суперпозиции, основанные на генерировании регулярной шестигранной сетки внутри объема с последующим созданием переходных ячеек, соединяющих регулярную сетку с четырехугольниками на границе [Schneiders et al., 1996]. Для этих методов часто возникает ситуация наличия вырожденных ячеек или ячеек плохого качества вдоль границы. При всей практичности такого подхода он является весьма трудоемким для реализации [Thompson et al., 1999].

Вторую группу составляют так называемые фронтальные алгоритмы, когда построение начинается от граничной четырехугольной сетки [Staten et al., 2005]. На каждой итерации проекцией к центру объема создается новый слой шестигранных ячеек. Такой способ обладает тем же недостатком, что и его двумерный аналог — сложным поиском соединения граней и ребер ячеек в случае пересечения двух и более фронтов.

Методы декомпозиции и блочной декомпозиции основаны на идее (ручной или полуавтоматической) декомпозиции тела на простые геометрические примитивы (параллелепипед, шар, цилиндр и др.), для каждого из которых построение расчетной сетки является простой задачей, с последующим объединением этих частей в исходный объект. Одним из способов автоматизации процесса декомпозиции на блоки является использование срединных поверхностей [Price, Armstrong, 1995]. Достоинством данной группы методов является их высокая скорость и качество получаемой сетки, однако они имеют высокую трудоемкость их использования для дискретизации сложных нестандартных областей.

Простейшим непрямым методом построения является алгоритм Эппштейна — пошаговое преобразование каждой тетраэдральной ячейки путем разбиения исходной на четыре новых шестигранника [Eppstein, 1996]. В этом случае модифицируются граничные ребра и получаемая

сетка полностью состоит из шестигранных элементов. Отметим низкое качество получаемых сеток, а также невозможность получения результирующей сетки, соразмерной с исходной, что делает его редко применяемым в практике вычислений.

Среди фронтальных непрямых методов можно выделить алгоритм H-Morph [Owen, Saigal, 2000], в котором трансформация ячеек тетраэдров в шестигранники начинается от так называемых граней подвижного фронта — замкнутой совокупности граней, отделяющей построенные шестигранные ячейки сетки от тетраэдральных. Далее в процессе работы фронт постоянно обновляется в соответствии с текущим состоянием сетки. Примеры работы данного алгоритма для двухмерного случая показывают хорошие результаты [Owen et al., 1998; Сковпень, 1998; Караваев и др., 2012], однако нужно отметить сложную практическую реализацию метода в 3D, где затраты построения сетки возрастают.

Выделим еще группу методов поэлементного преобразования тетраэдров на основе шаблонов, которые обеспечивают невырожденность ячеек. Однако, часто в результирующей сетке остаются тетраэдральные ячейки [Копысов, Пономарев, 2009]. Заслуживают внимания методы, использующие двойственное представление шестигранный сетки через ортогональные плоскости, определяющие связность ее ячеек [Murdoch et al., 1997]. Преимуществом данной группы методов является отсутствие вырожденных элементов, но накладываются требования построения качественной начальной дискретизации, что само по себе является трудоемкой задачей.

В данной работе представляется метод построения шестигранный сетки из объемных данных, основанный на алгоритме двойственного контурирования, используемого для построения изображения моделируемых тел.

Объемное представление данных

Объемные данные \mathcal{V} представляются в виде набора точек (вокселей) $v \in \mathcal{V}$, каждая из которых имеет некоторое скалярное значение, являющееся атрибутом данной точки в пространстве (плотность, концентрация, температура и т. д.). Объемные (воксельные) данные есть выборка значений функции на декартовой сетке координат, которая может быть записана как $\mathcal{V} = \{F(i, j, k) | i, j, k, \text{ где } i, j, k — \text{индексы в } x, y, z \text{ декартовых координат сетки}\}$. Изоповерхность (граница объекта, неявная поверхность), соответствующая изозначению α , определяется как $I_F(\alpha) = \{(x, y, z) | F(x, y, z) = \alpha\}$.

В работе рассматривается представление воксельных данных для случая, когда сохраняется наименьшее расстояние от каждого вокселя до границы рассматриваемого тела с учетом знака. Это представление данных получило название «знакоопределенные поля расстояний». В данной статье описывается метод преобразования поверхностного представления расчетной области в формат знакоопределенных полей расстояния. На основе полученных данных строится шестигранный расчетная сетка с использованием алгоритма двойственного контурирования. Адаптация построенной сетки в областях, требующих дополнительного уточнения границы, производилась шаблонами «перестроения на 27 ячеек» Шнейдера.

Метод преобразования поверхностного представления расчетной области в формат воксельных данных

В качестве исходного и широко представленного на практике формата поверхностного представления использовался формат .STL (описание тела с помощью поверхностной треугольной сетки). Кроме того, в некоторых случаях такой формат является единственным, например, для данных, полученных путем сканирования. После преобразования создается файл формата .rawiv, содержащий знакоопределенные поля расстояний. В отличие от стандартного представления, где значение вокселя F не содержит знак (плотность, температура), расстояние является

знакоопределенным ($F > 0$, если воксель внутри тела, $F < 0$, если снаружи, $F = 0$ в случае расположения на границе). Функция $SD_{\mathcal{T}}(p)$, определяющая такое расстояние, получила название «знакоопределенная функция расстояния». Пусть $\Omega \subset R^3$ замкнутая область, ограниченная поверхностной треугольной сеткой \mathcal{T} , тогда для произвольной точки $p \in R^3$ данную функцию можно записать в виде

$$SD_{\mathcal{T}}(p) = S_{\mathcal{T}}(p) \cdot D_{\mathcal{T}}(p), \quad D_{\mathcal{T}}(p) = \inf_{n \in \mathcal{N}} \|n - p\|, \quad S_{\mathcal{T}}(p) = \begin{cases} +1, & \text{если } p \in \Omega; \\ -1, & \text{если } p \notin \Omega. \end{cases}$$

где \mathcal{N} — множество узлов сетки \mathcal{T} , n — любой узел \mathcal{T} , принадлежащий \mathcal{N} . Как видно, знакоопределенная функция расстояния состоит из произведения функции знака $S_{\mathcal{T}}(p)$ и функции наименьшего расстояния $D_{\mathcal{T}}(p)$. При построении метода преобразования поверхностной треугольной сетки использовался следующий алгоритм: в цикле обходится массив вокселей декартовой сетки и для каждого вокселя v находится наименьшее расстояние до поверхности объекта с учетом знака [Fuhrmann, 2007].

Функция $SD_{\mathcal{T}}(p)$ вычисляется в два этапа: нахождение наименьшего расстояния; определение знака вокселя. Для нахождения наименьшего расстояния был использован алгоритм проекции исходной точки на поверхностную треугольную сетку, описывающую объект. Основная идея его состоит в проекции исходной точку p_0 на плоскость, образованную треугольником Δ , состоящим из вершин p_1, p_2, p_3 (рис. 1). Если проекция p' попадает внутрь треугольника, то длина отрезка $[p_1, p']$ является искомой величиной. В противном случае ищется ближайшая вершина или ребро треугольника Δ до точки p' и вычисляется расстояние до этого элемента. Если предположить, что ребро e , образованное точками p_1 и p_3 , является ближайшим к точке p' , и p'' есть проекция точки p' на e , то расстояние до треугольника Δ определяется как $\sqrt{|p'p''|^2 + |p'p_0|^2}$.

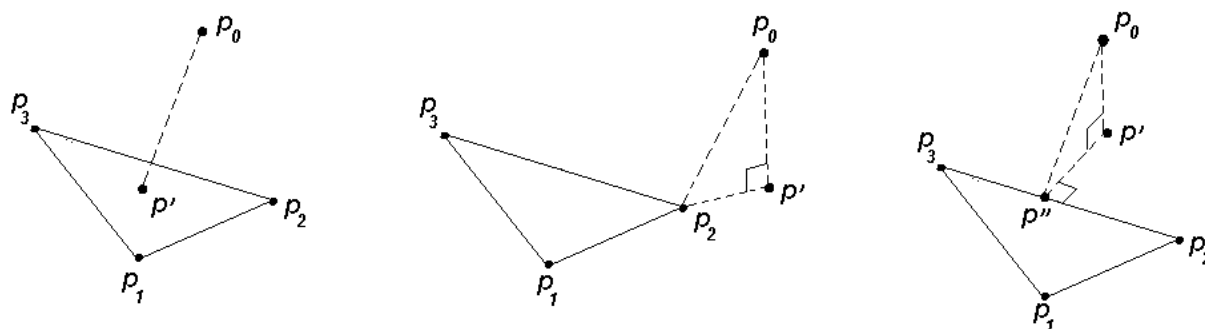


Рис. 1. Нахождение расстояния от точки p_0 до $\Delta p_1 p_2 p_3$

Положение произвольной точки p относительно тела определится с помощью нормалей заданных на его поверхности. Для формата .STL на поверхностной сетке заданы нормали треугольников. Нормаль ребра сетки вычисляется как нормализованная сумма нормалей прилежащих к нему треугольников. Пусть к ребру прилегают треугольники Δ_1 и Δ_2 , тогда нормаль ребра $\mathbf{n} = (\mathbf{n}_{\Delta_1} + \mathbf{n}_{\Delta_2}) / (\|\mathbf{n}_{\Delta_1} + \mathbf{n}_{\Delta_2}\|)$. Для вычисления нормали вершины сетки использовалась псевдонормаль, взвешенная по углам $\mathbf{n} = \sum_i \mathbf{n}_i \cdot \beta_i / \|\sum_i \mathbf{n}_i \cdot \beta_i\|$, где \mathbf{n}_i — нормали прилежащих к вершине треугольников, β_i — угол треугольника в рассматриваемой вершине.

Алгоритмы построения объемного изображения из воксельных данных

Самым известным и простым в реализации методом изображения объекта из объемных данных является алгоритм «марширующие кубы», предложенный Лоренсом [Lorensen, Cline,

1987]. В данном алгоритме осуществляется обход скалярного поля с выбором восьми соседних вокселей на каждой итерации. Далее определяются полигоны, необходимые для представления части изоповерхности, проходящей через данный куб. Совокупность данных полигонов и составляет поверхность объекта.

Наряду со своими преимуществами, такими как скорость и простота реализации, алгоритм имеет один существенный недостаток. Он плохо описывает острые углы рассматриваемой модели, малые углы пропускаются алгоритмом, вместо этого создаются полигоны, соединяющие точки пересечения с кубом декартовой решетки. Такой метод достаточно хорош для задания поверхности сферы, но дает плохую аппроксимацию параллелепипеда. Вторая особенность заключается в том, что сетка, генерируемая «марширующими кубами», является однородной, что приводит к неоправданным затратам памяти. Модификации алгоритма типа «расширенные марширующие кубы» требуют специальных вычислений при определении острого угла поверхности, а также дополнительный шаг переключения ребер для завершения аппроксимации в случае острых ребер границы.

Другую группу методов составляют так называемые двойственные методы. Алгоритм двойственного контурирования [Schaefer, Warren, 2002], создавался как метод построения изображения в компьютерной анимации, но благодаря своей простоте и эффективности получил широкое распространения в других областях компьютерного моделирования. Двойственное контурирование сохраняет острые углы объекта, а также генерирует поверхностную четырехугольную сетку. Основные особенности алгоритма:

- Использование структуры октодеревя для представления прямоугольной сетки. Каждый узел октодеревя содержит восемь вокселей, составляющих его вершины.
- Наряду с плотностью вокселя учитывается его нормаль к неявной поверхности. Нормаль точек пересечения с контуром поверхности определяется линейной интерполяцией соседних вокселей. Вершина сетки находится путем минимизации квадратичной функции погрешности $E(\mathbf{x}) = \sum_i (\mathbf{n}_i \cdot (\mathbf{x} - \mathbf{x}_i))^2$, где \mathbf{x}_i — вектор, содержащий координаты точки пересечения, а \mathbf{n}_i — ее единичная нормаль.
- Вычисление квадратичной функции погрешности для каждого куба декартовой сетки вокселей, пересекающего контур неявной поверхности вне зависимости от того, содержит он острый угол или нет. Благодаря этому вершины поверхности всегда располагаются внутри ячеек сетки. На рисунке 2, а белыми и черными квадратами изображены воксели расположенные внутри и снаружи тела соответственно, также показаны точки пересечения ребер решетки с контуром изоповерхности и их единичные нормали. На рисунках 2, б, в, г показаны варианты работы алгоритмов. Алгоритм «марширующие кубы» всегда располагает вершины на ребрах сетки, «расширенные марширующие кубы» генерируют вершину внутри ячейки для случая описания острого угла модели, двойственное контурирование всегда располагает вершины внутри ячеек сетки.

Исходя из возможности построения поверхностной четырехугольной сетки и использования структуры октодеревя, допускающего адаптацию сетки, метод был взят за основу при создании поверхностной однородной сетки предлагаемого алгоритма построения шестигранных ячеек.

Построение шестигранной сетки из объемного изображения

Предлагаемый алгоритм в качестве исходных данных использует знакоопределенные поля расстояний. Для хранения точек в пространстве и уровней детализации объекта используется структура октодеревя. Построения поверхностной четырехугольной сетки, а также выбор

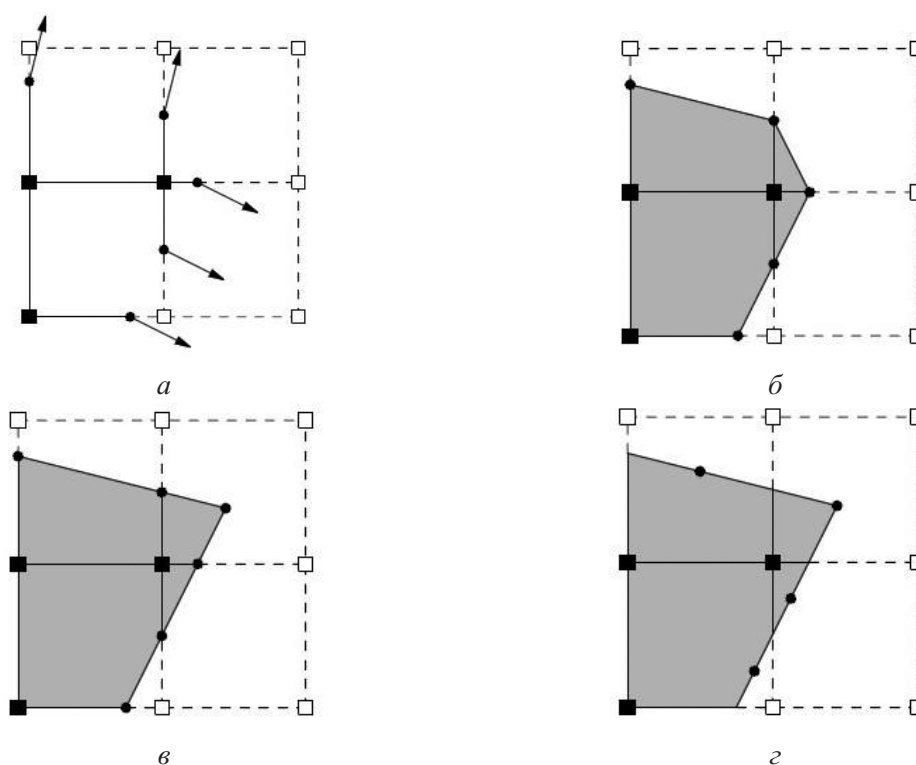


Рис. 2. Декартова сетка вокселей с нормальными к контуру поверхности (а). Поверхность построенная «марширующими кубами» (б). Поверхность, сгенерированная алгоритмом «расширенные марширующие кубы» (в). «Двойственное контурирование» (г)

начального уровня октодеревя выполняется на основе алгоритма двойственного контурирования, который позволяет сохранить острые углы и грани объекта. Основные этапы алгоритма следующие:

- Построение уровней октодеревя.
- Генерация однородной шестигранной сетки.
- Улучшение качества построенных ячеек.

Построение уровней октодеревя

Октодеревя — тип древовидной структуры данных, в которой у каждого внутреннего узла есть до восьми потомков. Как и в данном случае, деревья октантов чаще всего используются для разделения трехмерного пространства, рекурсивно разделяя его на восемь октантов. В алгоритме двойственного контурирования узел октодеревя хранит информацию о восьми вокселях, составляющих куб декартовой сетки. В каждом вокселе с индексами декартовой сетки (x_i, y_j, z_k) вычисляется градиент скалярного поля и нормаль равная нормализованному градиенту со знаком:

$$\text{grad}F(x_i, y_j, z_k) = \frac{1}{2} \begin{pmatrix} F(x_{i+1}, y_j, z_k) - F(x_{i-1}, y_j, z_k) \\ F(x_i, y_{j+1}, z_k) - F(x_i, y_{j-1}, z_k) \\ F(x_i, y_j, z_{k+1}) - F(x_i, y_j, z_{k-1}) \end{pmatrix}, \quad \mathbf{n}(x, y, z) = -\frac{\text{grad}F(x, y, z)}{|\text{grad}F(x, y, z)|},$$

где $F(x_i, y_j, z_k)$ — значение вокселя (x_i, y_j, z_k) .

В алгоритме используется структура октодера с размерностью данных $dim \times dim \times dim$, которая удовлетворяет условию $2^m = dim - 1$, $m \in \mathbb{N}$ и обеспечивает разделение подпространства строго кубической формы. Задаваемое число уровней октодера зависит от размерности исходных данных. В этом случае используется восходящий алгоритм, начинающий построение с самого нижнего уровня. При исходной размерности, равной $(2^k + 1)^3$, число уровней октодера равно k .

От выбранного уровня октодера зависит степень детализации изображаемого объекта, а следовательно точность аппроксимации границы физической модели. На рисунках 3, 4, 5, изображены примеры генерации изображения из файла объемных данных размерности $dim \times dim \times dim = 65 \times 65 \times 65$, при этом слева показаны узлы октодера, у которых хотя бы один воксель имеет значения больше нуля, именно в данных узлах располагаются вершины шестигранной сетки. Как видно, степень детализации изображения напрямую зависит от выбранного уровня октодера L , также от данного параметра зависит число узлов и ячеек построенной шестигранной сетки. Для уровня $L = 4$ число узлов сетки $N = 195$, а ячеек шестигранной формы — $M = 100$, при построении сетки на максимальном шестом уровне октодера число узлов возрастает до 57 980, а число шестигранников становится равным 52 074.

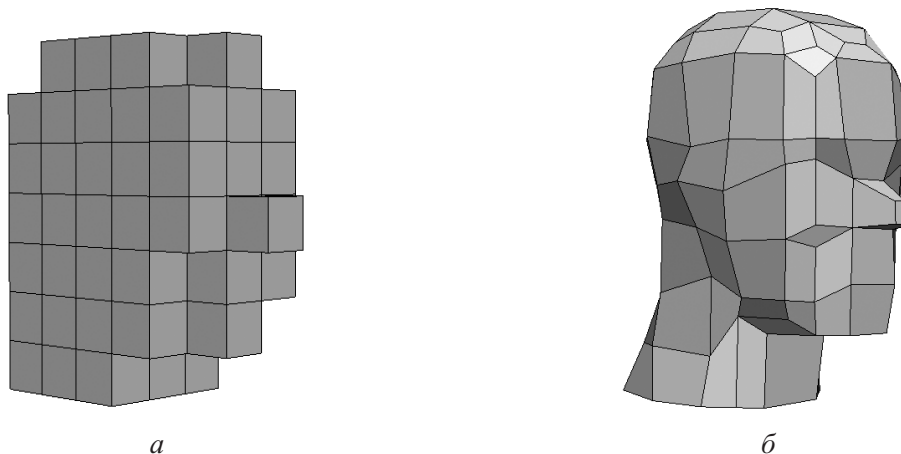


Рис. 3. $dim \times dim \times dim = 65 \times 65 \times 65$, уровень $L = 3$, число узлов сетки $N = 195$, ячеек — $M = 100$

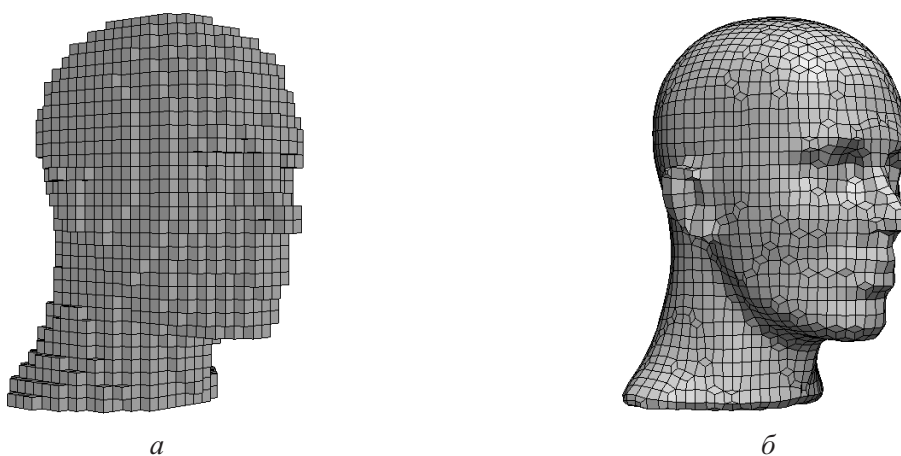


Рис. 4. $dim \times dim \times dim = 65 \times 65 \times 65$, уровень $L = 5$, число узлов сетки $N = 8073$, ячеек $M = 6561$

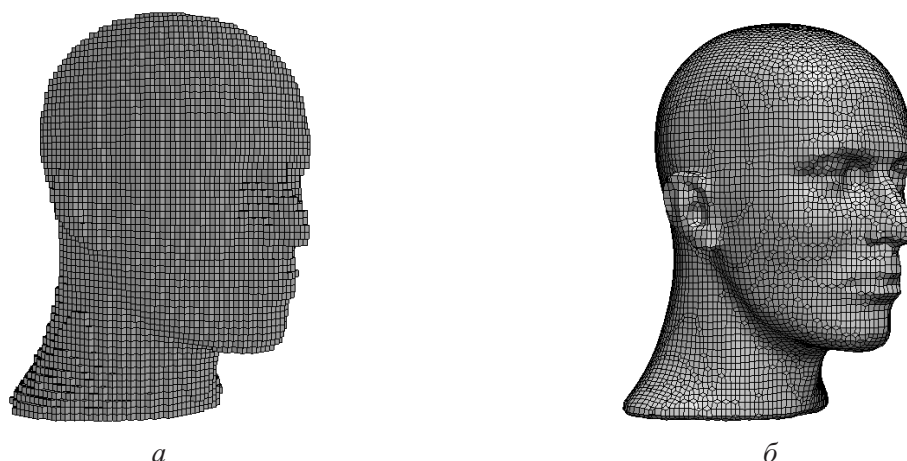


Рис. 5. $\dim \times \dim \times \dim = 65 \times 65 \times 65$, уровень $L = 6$, число узлов сетки $N = 57980$, а ячеек $M = 52074$

Построение неструктурированной шестигранной сетки

Построение шестигранной сетки начинается с определения начального уровня октодеревя, в узлах которого будут вычисляться ее вершины. От выбора начального уровня октодеревя зависит степень детализации рассматриваемого объекта, а также число ячеек сетки. После этого для каждого узла октодеревя, у которого один и более вокселей имеет значение $F \geq 0$, вычисляется соответствующая вершина сетки. Далее найденные вершины восьми соседних узлов октодеревя соединяются, образуя шестигранник. Таким образом, механизм формирования шестигранных ячеек выглядит достаточно простым, однако остается задача нахождения координат вершин сетки.

В предлагаемом методе все вершины (узлы) сетки можно разделить на две группы: граничные — вершина находится внутри узла, имеющего воксели разного знака, и внутренние — вершина располагается в узле октодеревя, находящегося внутри рассматриваемого объекта. Внутренняя вершина сетки располагается в центре масс узла октодеревя и вычисляется как $p = (p_1 + p_7)/2$, где p_1, p_7 — координаты первого и седьмого вокселя рассматриваемого узла.

Для нахождения граничной вершины используется алгоритм двойственного контурирования. Данный метод для каждого граничного куба октодеревя генерирует вершину, удовлетворяющую минимуму квадратичной функции $E(\mathbf{x}) = \sum_i (\mathbf{n}_i \cdot (\mathbf{x} - \mathbf{x}_{p_i}))^2$, где пары $\mathbf{x}_{p_i}, \mathbf{n}_i$ — соответствующие точки пересечения ребер куба с изоповерхностью и единичные нормали данных точек. Функция $E(\mathbf{x})$ определяется в матричной форме $E(\mathbf{x}) = \mathbf{x}^T \mathbf{A}^T \mathbf{A} \mathbf{x} - 2\mathbf{x}^T \mathbf{A}^T \mathbf{b} + \mathbf{b}^T \mathbf{b}$, где \mathbf{A} — матрица, составленная из нормалей точек пересечения \mathbf{n}_i , \mathbf{b} — вектор, элементы которого представлены в виде скалярного произведения $\mathbf{n}_i \cdot \mathbf{x}_{p_i}$. Минимум квадратичной функции находится решением переопределенной системы линейных уравнений $\mathbf{A} \mathbf{x} = \mathbf{b}$ алгоритмом Качмажа [Kaczmarz, 1993]. В случае если имеется множество решений минимума функции $E(\mathbf{x})$, она доопределяется так, чтобы вектор \mathbf{x} дополнительно минимизировал расстояние до центра масс точек с пересечения шестигранника с изоповерхностью. В таком случае выражение имеет вид $E'(\mathbf{x}) = E(\mathbf{x}) + (\mathbf{x} - \mathbf{x}_c)^2$. При этом к матрице добавляется одна строка, содержащая единицы, а к вектору \mathbf{b} добавляется сумма компонент центра масс.

Точки пересечения куба декартовой решетки с изоповерхностью определяются линейной интерполяцией по значениям соседних вокселей разного знака. Для координат вокселей \mathbf{x}_{p_1} и \mathbf{x}_{p_2} со скалярными значениями F_1, F_2 координаты точки пересечения \mathbf{x}_p вычисляются следующим

образом: $\mathbf{x}_p = \mathbf{x}_{p_1} - F_1 \cdot (\mathbf{x}_{p_2} - \mathbf{x}_{p_1}) / (F_2 - F_1)$. Нормаль в точке пересечения также вычисляется линейной интерполяцией по значениям нормалей вокселей на концах ребра, которому принадлежит данная точка.

Улучшение качества построенных ячеек

Одним из основных требований к расчетной сетке является невырожденность каждой ее ячейки. В данной работе условие невырожденности проверяется через положительность объемов десяти угловых тетраэдров, вписанных в шестигранник [Азаренок, 2008].

Первые пять тетраэдров изображены на рисунке 6, б — это четыре угловых тетраэдра при вершине ячейки, T_{1245} , T_{3427} , T_{6572} , T_{8754} , и внутренний — T_{2457} . Вторые пять состоят из четырех угловых тетраэдров T_{2316} , T_{4138} , T_{5861} , T_{7683} и одного внутреннего T_{1386} (рис. 6, в). Построенная

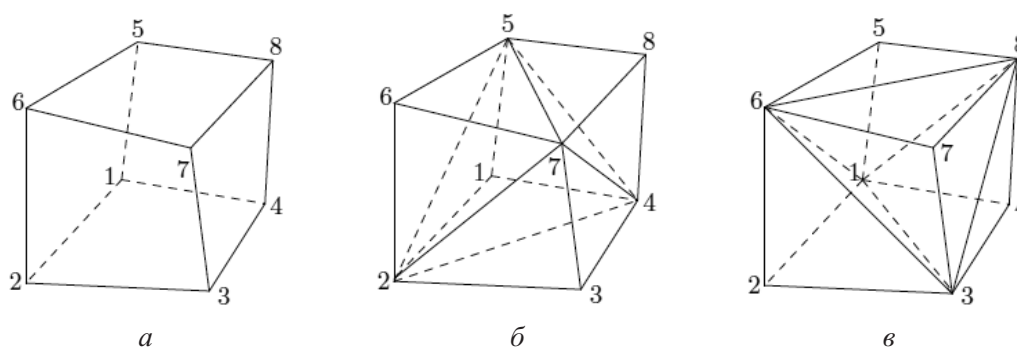


Рис. 6. Проверка невырожденности шестигранника

однородная шестигрannая сетка может содержать вывернутые ячейки и в этом случае требуются дополнительные операции по ее оптимизации.

Улучшение качества построенных шестигранников осуществляется с помощью процедуры сглаживания по Лапласу [Canann et al., 1998], а также вставки дополнительного слоя граничных ячеек. Во время процедуры сглаживания координаты внутреннего узла \mathbf{x}_n сетки вычисляются следующим образом: $\mathbf{x}_n = \mathbf{x}_n + 1/k \sum_{i=1}^k (\mathbf{x}_{n_i} - \mathbf{x}_n)$, где \mathbf{x}_{n_i} — координаты соседних k узлов (с каждым из них сглаживаемый узел связан ребром). Новое положение граничного узла определяется движением по касательной плоскости без изменения формы граничного контура сетки.

При построении сетки могут возникать ситуации, когда ячейка содержит два и более четырехугольника на границе. В этом случае простая процедура сглаживания является малоэффективной. В работе был использован метод вставки дополнительного граничного слоя ячеек. В этом случае от каждого граничного четырехугольника создается новый шестигранник, что позволяет значительно улучшить форму граничных ячеек (рис. 7).

Адаптивное перестроение шестигрannой сетки

При решении задач сеточными методами часто возникает необходимость уточнения полученного решения на выбранной подобласти или границ области. Это достигается алгоритмами перестроения начальной расчетной сетки. Необходимым условием такой операции является конформность сетки (согласованность узлов в соседних ячейках) и сохранение невырожденности ячеек после перестроения. В случае неструктурированной сетки особое внимание необходимо уделять выбору точек внутри перестраиваемых ячеек и обеспечению согласованного перехода между выбранными для перестроения ячейками и остальным множеством шестигранников.

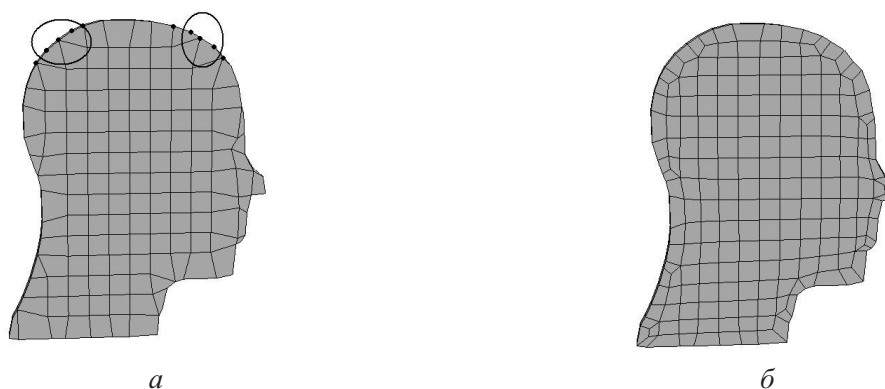


Рис. 7. Вставка буферного слоя ячеек

Основным способом перестроения шестигранной сетки являются шаблоны перестроения на 27 ячеек Шнейдера [Schneider, 1996]. Существенным преимуществом данного метода является независимость способа разбиения ячейки от ее соседей. В этом случае любой шестигранник, помеченный для перестроения, разбивается на 27 новых шестигранников, любой четырехугольник — на девять новых четырехугольников, любое ребро — на три новых ребра. На рисунке 8 показаны варианты разбиения шестигранника в зависимости от количества «помеченных» узлов, которые он имеет. Узел называется помеченным, если хотя бы один прилежащий к нему шестигранник должен быть перестроен.

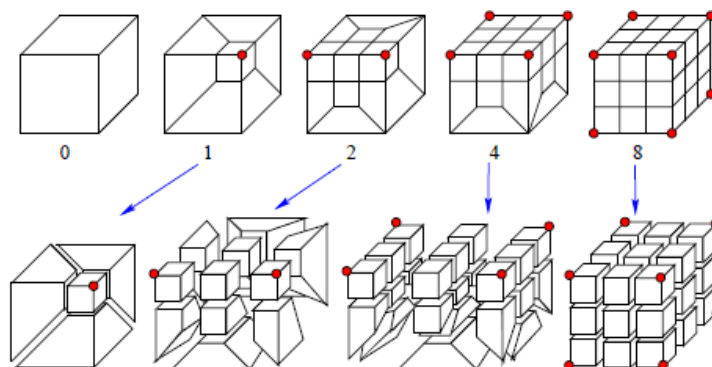


Рис. 8. Шаблоны перестроений

Рассматриваются четыре варианта расположения помеченных узлов для разбиения шестигранника, все остальные случаи конвертируются к рассматриваемым четырем шаблонам.

Уточнение границы осуществляется путем движения граничных узлов к точке пересечения с изоповерхностью. Точки пересечения более точно описывают границу тела, чем исходная вершина, полученная путем минимизации квадратичной функции. Движение граничных вершин осуществляется вдоль их нормалей, вычисленных трилинейной интерполяцией по значениям нормалей восьми окружающих вокселей (рис. 9).

Примеры построения расчетных сеток

В данном разделе представлены результаты работы алгоритма генерации расчетной шестигранной сетки из объемных (воксельных) данных. Исходные объемные данные получа-

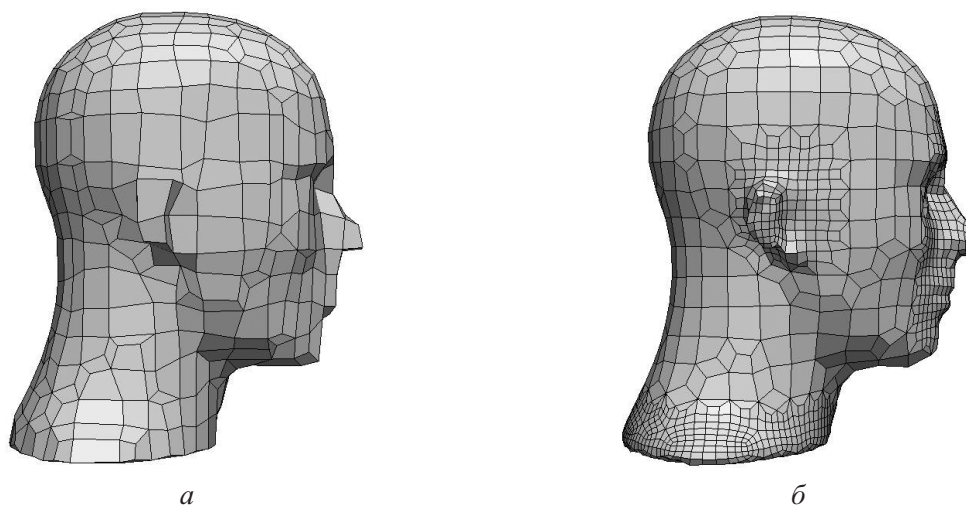
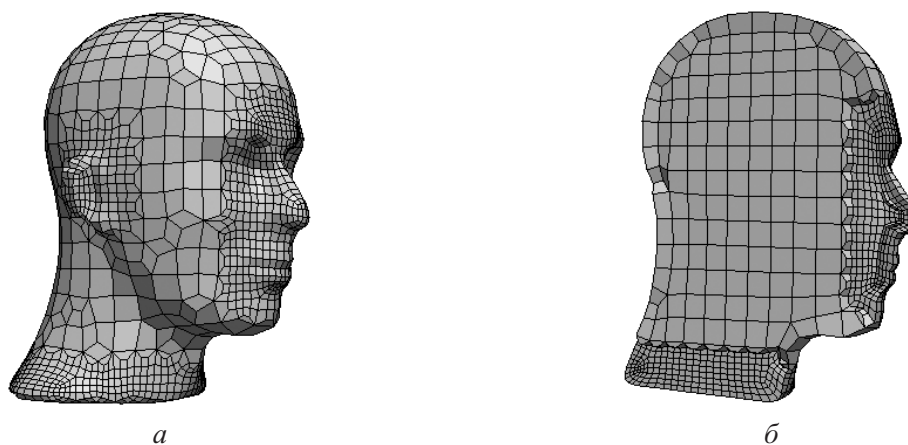


Рис. 9. Шестигранная сетка до (а) и после уточнение границы (б)

лись непосредственно из воксельного представления, а также путем конвертации начальной поверхностной сетки, описывающей геометрическую модель.

На рисунке 10 приведен пример сгенерированной шестигранной сетки. Размерность начальных составляла $65 \times 65 \times 65$, уровень октодеревя, выбранный для извлечения изоповерхности, а также генерации структурированной шестигранной сетки равнялся четырем. Сетка содержала $M = 814$ шестигранников и $N = 1\,209$ узлов, после адаптивного перестроения границы число шестигранников возросло до 11 980, число узлов стало равным 13 672. Каждая уточняемая ячейка сетки была разделена на 27 новых шестигранников. В таблице 1 приведены характеристики начальной однородной сетки, сетки после операций вставки граничного слоя ячеек и сглаживания, а также сетки после локального перестроения области.

Рис. 10. Область А. Шестигранная сетка построенная из файла объемных данных размерности $65 \times 65 \times 65$

На рисунке 11 представлен пример построения шестигранной сетки из данных, полученных путем преобразования поверхностной треугольной сетки, содержащей 255 треугольных ячеек. Объемные данные после преобразования содержали 65 вокселей вдоль каждой из осей, при этом для построения сетки был выбран пятый уровень октодеревя. Характеристики шестигранной сетки приведены в таблице 2. Все сетки, представленные в данной таблице, не содержат вырожденных ячеек.



Рис. 11. Область Б. Сетка из шестигранных ячеек построена из поверхностной сетки, содержащей 225 треугольных ячеек

На рисунке 12 приведен пример сетки на области «В». Невырожденность ячеек сетки (см. таблицу 2) была обеспечена вставкой дополнительного слоя шестигранников вдоль границы цилиндрического выреза.

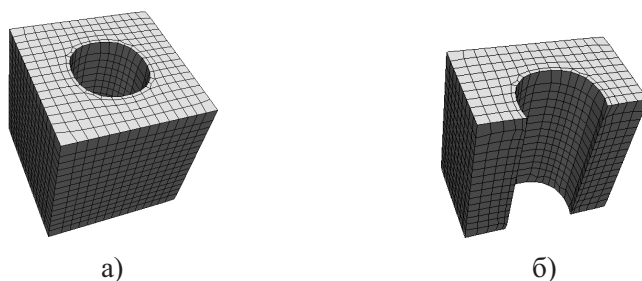


Рис. 12. Область В. Сетка построена из конвертированного файла формата .STL, содержащего 168 треугольных ячеек

Шестигранная сетка для области «Г» (Рис. 13) построена для размерности данных $65 \times 65 \times 65$, уровне октодеревы $L = 6$. В случае необходимости уточнения граничного контура потребуются исходные данные более высокого разрешения $129 \times 129 \times 129$, $257 \times 257 \times 257$ и т. д.

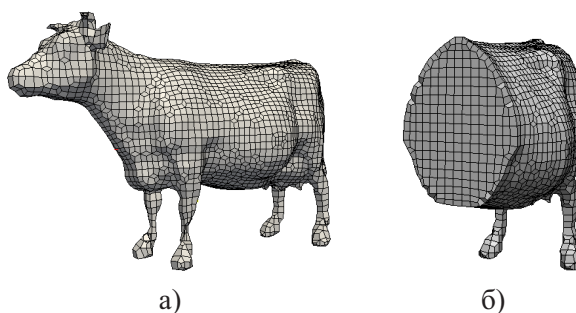


Рис. 13. Область Г. Сетка построена из конвертированного файла формата .STL, содержащего 5804 треугольных ячеек

Благодаря воксельной структуре данных метод построения сетки является достаточно простым в реализации и обладает небольшими вычислительными затратами. Программная реализация алгоритма осуществляется на облегченной структуре данных, что обеспечивает высокую интеграцию программы с приложениями в пакете FESstudio [Копысов и др., 2008].

Таблица 1. Характеристики шестигранных сеток, построенных на области А

Характеристика	Однородная сетка	Сетка после улучшения качества	Перестроенная сетка
Число узлов	1 209	1 915	13 672
Число ячеек	814	1 517	11 980
Макс. ребро	7.4	5.6	5.5
Мин. ребро	0.8	0.8	0.25
Аспектное соотношение	6.8	4.6	49.2
Макс. угол искажения четырехугольников	172.8°	85.6°	85.6°
Количество вырожденных шестигранников	55	0	0
с-валентность	8(501)	4(113), 6(150), 8(736), 10(208)	4(98), 6(1419), 8(7524), 10(516)

Таблица 2. Характеристики шестигранных сеток, построенных на областях Б, В и Г

Характеристика	Область Б	Область В	Область Г
Число узлов	4 720	4 042	14 774
Число ячеек	3 864	3 128	12 224
Макс. ребро	2.4	3.3	2.2
Мин. ребро	0.5	0.5	0.04
Аспектное соотношение	3.7	5.2	26.1
Макс. угол искажения четырехугольников	39.5°	76.8°	130.2°
с-валентность	4(216), 6(256), 8(2208), 10(448)	4(2), 6(60), 8(2128), 10(112)	4(538), 6(894), 8(7517), 10(1244)

Список литературы

- Азаренок Б. Н. Вариационный метод построения гексаэдральных сеток с управляющей метрикой // Матем. моделирование. — 2008. — Т. 20, № 9. — С. 3–22.
- Караваев А. С., Копысов С. П., Пономарев А. Б. Алгоритмы построения и перестроения неструктурированных четырехугольных сеток // Вычислительная механика сплошных сред. — 2012. — Т. 5. — С. 144–150.
- Копысов С. П., Новиков А. К., Пономарев А. Б., Рычков В. Н., Сагдеева Ю. А. Программная среда построения расчетных моделей метода конечных элементов для параллельных распределенных вычислений // Информационные технологии. — 2008. — № 3. — С. 75–82.
- Копысов С. П., Пономарев А. Б. Непрямой метод построения неструктурированных шестигранных сеток с преобладанием шестигранных элементов // В сб. «Актуальные проблемы, математики, механики, информатики». — Екатеринбург: ИММ УрО РАН, 2009. — С. 65–71.

- Сковпень А. В. Реализация фронтального алгоритма построения нерегулярных четырехугольных сеток // ВАИТ. Серия: Математическое моделирование физических процессов. — 2005. — Вып. 1. — С. 9–30.
- Canann S. A., Tristano J. R. and Staten M. L. An Approach to Combined Laplacian and Optimization-Based Smoothing for Triangular, Quadrilateral, and Quad-Dominant Meshes // 7th International Meshing Roundtable, Sandia National Labs. — 1998 — P. 479–494.
- Eppstein D. Linear complexity hexahedral mesh generation // In Symposium on Computational Geometry. — 1996. — P. 58–67.
- Fuhrmann S. Volume data generation from triangle meshes using the signed distance function // Darmstadt University of Technology. — Germany, 2007.
- Kaczmarz S. Approximate solution of systems of linear equations // Internat. J. Control. — 1993. — Vol. 57, No. 6. — P. 1269–1271.
- Lorensen W., Cline H. Marching Cubes: a high resolution 3D surface construction algorithm // Computer Graphics (SIGGRAPH 87 Proceedings). — 1987. — P. 163–169.
- Murdoch P., Benzley S. E., Blacker T. D., Mitchel S. A. The spatial twist continuum: A connectivity based method for representing all-hexahedral finite element meshes // Finite Elements in Analysis and Design. — 1997. — No. 28. — P. 137–149.
- Owen S. J., Saigal S. H-Morph: An Indirect Approach to Advancing Front Hex Meshing // International Journal for Numerical Methods in Engineering. — 2000. — P. 289–312.
- Owen S. J., Steven J., Staten M. L., Cannan S. A. and Saigal S. Advancing Front Quadrilateral Meshing Using Triangle Transformations // 7th International Meshing Roundtable, Sandia National Labs. — 1998. — P. 409–428.
- Price M. A., Armstrong C. G. Hexahedral mesh generation by medial surface subdivision: Part 1 // International Journal for Numerical Methods in Engineering. — 1995.
- Schneiders R., Schindler R., Weiler F. Octree-based Generation of Hexahedral Element Meshes // 5th Annual International Meshing Roundtable. — 1996. — P. 205–216.
- Schaefer S. and Warren J. Dual Contouring: The Secret Sauce // Technical Report 02-408, Department of Computer Science, Rice University. — 2002. — 6 p.
- Schneider R. Refining quadrilateral and hexahedral meshes // Proc. 5th Int Conference on Numerical. Grid Generation In Computational Field. — 1996. — P. 679–688.
- Staten M. L., Owen S. J., Blacker T. D. Unconstrained Paving and Plastering: A New Idea for All Hexahedral Mesh Generation // Proceedings, 14th International Meshing Roundtable. — Sandia National Laboratories:Springer-Verlag, 2005. — P. 399–416.
- Thompson J. F., Bharat K. S., Nigel P. W. Hand book of grid generation. — New York: CRC Press, 1999. — 1200 p.