

УДК 004.925.8

© А. С. Караваяев, С. П. Копысов

## ПОСТРОЕНИЕ АДАПТИВНЫХ ШЕСТИГРАННЫХ СЕТОК ИЗ ПОВЕРХНОСТНОЙ И ВОКСЕЛЬНОЙ ГЕОМЕТРИЧЕСКИХ МОДЕЛЕЙ

Представлена модификация разработанного генератора шестигранных сеток из воксельных данных, касающаяся возможности построения адаптивных расчетных сеток. Область перестроения определяется исходя из геометрических особенностей описываемой модели с выраженными участками относительно малых размеров. Предложен универсальный критерий перестроения ячеек как в случае объемного (воксельного), так и поверхностного (STL) представления геометрии модели. Описаны используемые шаблоны перестроения шестигранных ячеек, обеспечивающие конформное замыкание сетки.

*Ключевые слова:* генерация и перестроение шестигранных сеток, объемные данные, STL-геометрия.

DOI: [10.35634/vm230310](https://doi.org/10.35634/vm230310)

### Введение

Применение в рамках конечно-элементного анализа шестигранных сеток позволяет повысить точность и численную эффективность моделирования ряда физических процессов [1]. В отличие от тетраэдральных сеток, не решена в полной мере задача автоматической генерации шестигранных ячеек заданного качества, обеспечивающих точное описание границы произвольной модели [2].

В большинстве случаев разработка сеточных генераторов осуществляется для поверхностного описания исходной геометрии модели (STL, B-rep, B-splines), которое широко используется в существующих CAD-системах и является основным способом применяемым в инженерных расчетах. Среди основных сложностей в данном случае является точное восстановление острых углов и ребер модели ячейками сетки.

При этом для некоторых приложений таких, как биомоделирование, более естественным является вариант использования объемных (воксельных) данных (DICOM, NIFTI, SDF), полученных со сканеров магнитно-резонансной или компьютерной томографии. Точное восстановление граничного контура не является критичным при построении сеток биологических тканей и органов. Вместе с тем возникает необходимость решения трудоемкой задачи фильтрации искусственных погрешностей (артефактов), возникающих в процессе сканирования реальных физических объектов.

Разработанные к настоящему времени алгоритмы построения шестигранных сеток в большинстве случаев относятся к одной из следующих групп: продвигаемого фронта («paving», «plastering») [3], отображения («mapping», «sweeping») [4], декомпозиции («medial surface transformation», «midpoint subdivision») [5, 6], конвертации тетраэдров в шестигранники [1, 2], двойственного представления («whisker weaving») [7], регулярной сетки и октодеревьев [8–11]. Каждый из указанных методов в сравнении с другими обладает преимуществами и недостатками относительно производительности, робастности, степени автоматизированности, качества ячеек, способности описывать сложные геометрии, заданные в различных форматах (поверхностное и объемное представления), а также возможности построения полностью шестигранной сетки [12, 13].

Использование адаптивных сеток позволяет добиться оптимального баланса между вычислительными затратами на решение задачи и точностью получаемого результата. Проведено значительное число исследований в области адаптивного перестроения исходной шестигранной сетки и разработан ряд способов по достижению оптимальной плотности распределения ячеек, описывающих модель [9].

Область перестроения может задаваться вручную, а также определяться автоматически на основании критериев, оценивающих степень кривизны поверхности и наличие геометрических особенностей в виде участков относительно малых размеров.

Самым распространенным способом создания адаптивных сеток являются методы октодеревьев и регулярной сетки, при этом большинство из указанных работ относится к случаю, когда исходная геометрия модели представляется в виде ограничивающих ее поверхностей.

В работе [9] предложен подход, включающий поиск участков перестроения по общим вышеперечисленным критериям. Авторы публикации [14] формируют сгущение сетки на участках поверхности, имеющих высокую степень кривизны. В статье [15] предложена специальная трехмерная метрика, позволяющая увеличить степень анизотропности получаемой адаптивной шестигранной сетки. Работа [16] описывает подход генерации адаптивных шестигранных сеток для моделей, состоящих из многокомпонентных материалов. Отметим также исследования по перестроению шестигранных сеток из объемных данных из [10].

Генерация адаптивной сетки осуществляется на основе шаблонов перестроения, впервые предложенных в работах Р. Шнейдера [8]. В настоящее время разработано несколько вариантов таких шаблонов (по «двум» или «трем» ячейкам), обеспечивающих конформное замыкание перестраиваемой области [14, 17, 18] и отличающихся по степени качества и сгущения получаемых шестигранников, а также «разрастания» области согласующего перестроения.

В данной работе рассматривается модификация ранее разработанного воксельного сеточного генератора [12, 13] для построения адаптивных шестигранных сеток. Применяется новый критерий перестроения, обеспечивающий более точное описание границ модели на участках относительно малых размеров.

Рассматриваемый сеточный генератор относится к семейству методов на основе регулярной сетки и состоит из следующих основных этапов: построение однородной ортогональной сетки; перестроение ячеек в заданных областях; вставка буферного слоя вдоль границы области; проекция граничных узлов на поверхность модели; улучшение качества построенных ячеек.

Алгоритм применим для построения шестигранных сеток как из поверхностного, так и объемного формата описания геометрии.

## § 1. Представление данных геометрии модели

Воксельные данные  $\mathcal{V}$  есть выборка скалярных значений функции на декартовой сетке координат  $\mathcal{V} = \mathcal{F}(i, j, k)$ , где  $i, j, k$  — индексы в  $x$ -,  $y$ -,  $z$ -координатах декартовой сетки. Отдельная точка  $\nu^{(i,j,k)}$  декартовой сетки определяет воксель. Изоповерхность (функция уровня, граница тела), соответствующая значению  $\alpha$ , есть множество точек в которых функция  $\mathcal{F}$  равняется указанной величине  $I_{\mathcal{F}}(\alpha) = \{(x, y, z) \mid \mathcal{F}(x, y, z) = \alpha\}$ .

Объемные данные могут быть представлены посредством задания изоповерхностей и их изображения в виде полигональных сеток, либо напрямую через рендеринг объема как блока данных.

Разрабатываемый алгоритм использует стандартизированный формат томографических данных DICOM для построения сеток биологических тканей и органов. В случае CAD-моделей применяется поверхностная треугольная сетка (STL) в комбинации с файлом формата

«знакоопределенные поля расстояний» (SDF). Каждый воксель SDF файла  $\nu$  сохраняет расстояние до поверхности тела  $\Omega$  с учетом знака ( $\mathcal{F}(\nu) \geq 0$ , если воксель внутри тела  $\Omega$ , и  $\mathcal{F}(\nu) < 0$  в противном случае).

В случае геометрии, заданной в формате STL, для представления всех вершин и ребер геометрической модели дополнительно строится множество  $\Lambda$  «характерных» ребер и узлов модели. Множество  $\Lambda$  можно рассматривать как «каркас» модели, сохраняющий все ее острые «характерные» формы. Для сохранения точности описания границы результирующая шестигранная сетка должна покрывать  $\Lambda$  непрерывной цепочкой ее узлов и ребер [12, 13].

На рис. 2, а, б изображена STL-геометрия детали № 1, а также соответствующее множество  $\Lambda$ , содержащее 10 ребер.

## § 2. Генерация однородной ортогональной сетки

В предыдущей версии алгоритма построение структурированной ортогональной сетки осуществлялось последовательным обходом массива объемных данных и формированием ячеек из наборов восьми соседних вокселей на основе оценки положительности их значений [12, 13].

Однако, в случае наличия в модели геометрических особенностей, может возникнуть ситуация, когда участок малых размеров полностью находится внутри ячейки, при этом все восемь вокселей данной ячейки находятся вне тела.

В [9] предлагается включать ячейку в сетку, если она имеет хотя бы одно пересечение своих ребер либо диагоналей с границей модели, заданной в виде триангулированной поверхности.

В случае описания исходных данных в формате STL указанный подход позволяет решить задачу обнаружения участков малых размеров, либо толщины. Вместе с тем он не применим, когда исходная геометрия описана исключительно в формате воксельных данных.

В данной работе предлагается построение ячейки на основании относительного значения объема тела  $\Omega$ , содержащегося внутри ячейки  $c$ .

Обозначим через  $V_c$  объем шестигранника  $c$  и через  $V_{c \cap \Omega}$  объем тела  $\Omega$ , находящегося внутри  $c$ . Относительную величину объема внутри ячейки определим как  $V_c^\Omega = V_{c \cap \Omega} / V_c$ . Ячейка  $c$  включается в шестигранную сетку  $\mathcal{T}$ , если  $V_c^\Omega \geq \varepsilon$ , где  $\varepsilon \in (0, 1]$ .

Пусть  $\dim_z \times \dim_y \times \dim_x$  — размерность воксельного массива  $\mathcal{V}$  и  $h$  — шаг (интервал) между вокселями, участвующими в построении структурированной ортогональной сетки. В таком случае процесс построения однородной шестигранной сетки  $\mathcal{T}$  записывается в следующем виде:

1. **for** ( $z = 1; z < \dim_z / h; z++$ )
2.   **for** ( $y = 1; y < \dim_y / h; y++$ )
3.     **for** ( $x = 1; x < \dim_x / h; x++$ )
4.        $x_h = hx, y_h = hy, z_h = hz$
5.       Сформировать ячейку  $c = \{\nu^{(x_h+i, y_h+j, z_h+k)}\}_{i,j,k=0,h}$
6.       **if**  $V_c^\Omega \geq \varepsilon$  **then**
7.         Включить  $c$  в  $\mathcal{T}$
8.       **endif**
9.     **endfor**
10.   **endfor**
11. **endfor**

Значение  $V_c^\Omega$  вычислим из отношения части точек, находящихся внутри тела  $\Omega$ , к общему числу точек, равномерно распределенных внутри  $c$  (метод Монте–Карло).

Рассмотрим трилинейное отображение  $f_c$  из единичного куба в шестигранную ячейку

$$f_c: (u, v, w) \in [0, 1]^3 \rightarrow c(1-u)(1-v)(1-w)\mathbf{x}^1 + (1-u)(1-v)w\mathbf{x}^2 + (1-u)v(1-w)\mathbf{x}^3 + \\ + (1-u)vwx^4 + u(1-v)(1-w)\mathbf{x}^5 + u(1-v)w\mathbf{x}^6 + uv(1-w)\mathbf{x}^7 + uvw\mathbf{x}^8,$$

где  $\{\mathbf{x}^i\}_{i=1..8}$  — вершины ячейки  $c$ .

При размерности массива точек, распределенных внутри  $c$ , равной  $n \times n \times n$  вычисление  $V_c^\Omega$  будет выглядеть следующим образом:

1.  $s = 0$
2. **for** ( $k = 0; k < n; k++$ )
3.     **for** ( $j = 0; j < n; j++$ )
4.         **for** ( $i = 0; i < n; i++$ )
5.              $u_i = i/n, v_j = j/n, w_k = k/n$
6.              $\mathbf{x}^{(i,j,k)} = f_c(u_i, v_j, w_k)$
7.             **if**  $\mathcal{F}(\mathbf{x}^{(i,j,k)}) \geq 0$  **then**
8.                  $s = s + 1$
9.             **endif**
10.         **endfor**
11.     **endfor**
12. **endfor**
13.  $V_c^\Omega = s/n^3$

Пример построения структурированной ортогональной сетки для  $\varepsilon = 0.5$  приведен на рис. 2, в.

### § 3. Перестроение ячеек однородной ортогональной сетки

В случае если модель имеет существенное отличие в толщине размеров или небольшие и тонкие участки, построенной однородной шестигранной сетки не всегда достаточно для описания указанных особенностей геометрии тела. В работах [9, 15] предложены подходы для установления критериев уточнения с использованием цифровой топологии, анализирующие связность соседних пикселей (в двух измерениях) и вокселей (в трех измерениях).

Дадим следующую классификацию ячеек однородной шестигранной сетки  $\mathcal{T}$ :

- ячейка  $c$  является «граничной ячейкой», если  $V_c^\Omega < 1$ ;
- ячейка  $c$  является «внутренней ячейкой первого типа», если  $V_c^\Omega = 1$  и число прилегающих к ней ячеек вдоль граней меньше 6;
- ячейка  $c$  является «внутренней ячейкой второго типа», если  $V_c^\Omega = 1$  и число прилегающих к ней ячеек вдоль граней равно 6.

По аналогии с [9, 15] для точного описания геометрии потребуем не менее трех слоев ячеек внутри любого участка модели, описываемого сеткой.

Зададим следующий критерий формирования списка  $\Upsilon$  ячеек, помеченных для перестроения:

- граничная ячейка должна быть включена в  $\Upsilon$ , если в ее узлах отсутствуют прилегающие внутренние ячейки;
- граничная ячейка должна быть включена в  $\Upsilon$ , если все ее узлы находятся внутри тела (случай отверстия малых размеров, полностью находящегося внутри ячейки);
- внутренняя ячейка первого типа должна быть включена в  $\Upsilon$ , если в ее узлах отсутствуют прилегающие внутренние ячейки второго типа.

После выполнения указанных шагов для перестроения также помечаются ячейки, прилегающие вдоль граней к множеству ячеек из  $\Upsilon$ .

В примере, представленном на рис. 2, в, список  $\Upsilon$  состоит из граничных ячеек (выделены темным цветом), не имеющих прилегающих к ним внутренних шестигранников.

Для создания адаптивной сетки использованы шаблоны разбиения по «трем» ячейкам Р. Шнейдера. 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 рис. 1 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. Рассматривалось четыре варианта расположения «помеченных» узлов для разбиения шестигранника, все остальные случаи конвертировались к рассматриваемым четырем шаблонам [12, 13].

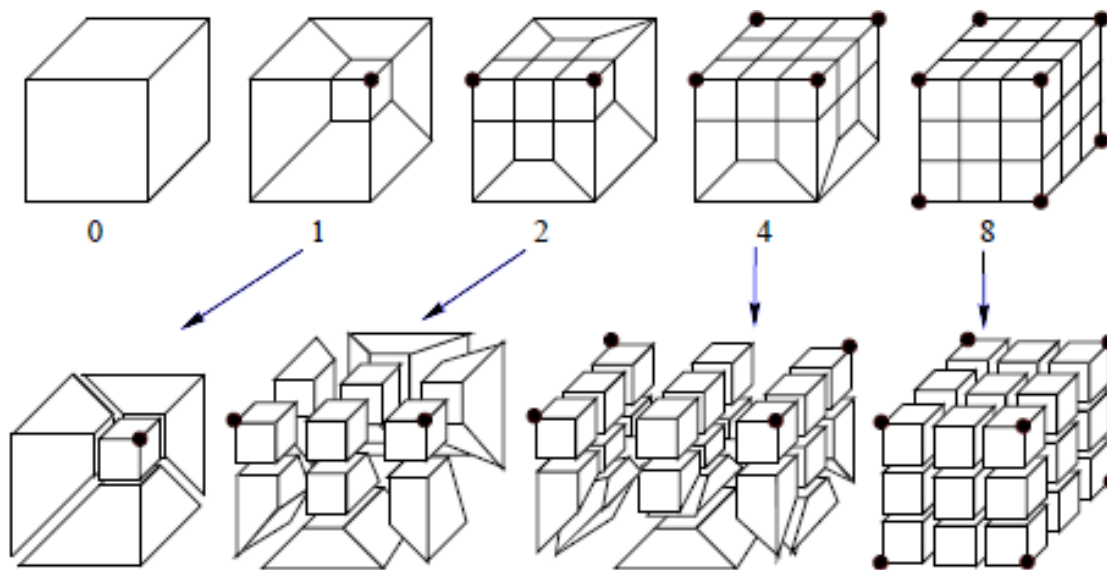


Рис. 1. Шаблоны перестроений

После завершения процедуры перестроения повторно проверяется условие удовлетворения пороговому значению объема, содержащегося внутри каждой ячейки сетки. Если  $V_c^\Omega < \varepsilon'$ , то  $c$  исключается из  $\mathcal{T}$ , где  $\varepsilon'$  — пороговое значение задаваемое после перестроения.

Таким образом, оценивается необходимость нахождения в сетке новых ячеек, полученных в результате перестроения, для части из которых возможен случай  $V_c^\Omega = 0$ .

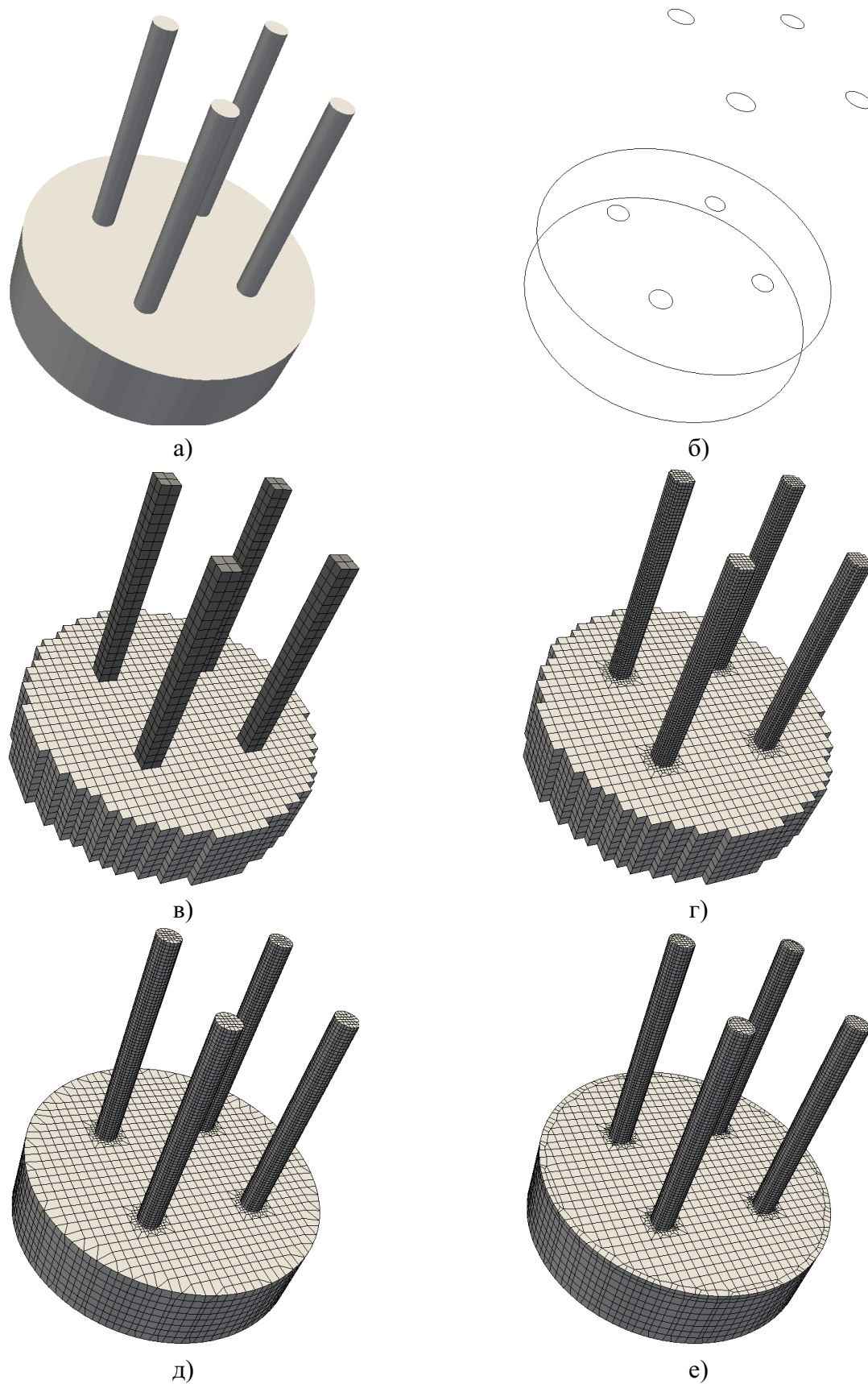
Кроме того, при  $\varepsilon' > \varepsilon$  рассматриваемая операция дополнительно корректирует остальную часть сетки. Если начальный параметр  $\varepsilon$  выбран достаточно малым, в  $\mathcal{T}$  имеются ячейки со значением  $V_c^\Omega \approx 0$ , которые затрудняют проведение процедуры восстановления контура модели. В целях повышения эффективности работы процедуры восстановления контура рекомендуется выбор параметра  $\varepsilon' > 0.5$ .

Пример перестроенной шестигранной сетки для случая  $\varepsilon' = 0.5$  изображен на рис. 2, г.

#### § 4. Восстановление контура модели

Следующим шагом алгоритма является проекция граничных узлов ортогональной сетки на множество  $\Lambda$  для обеспечения аппроксимации «характерных» вершин и ребер модели. Процесс адаптации выполняется до топологических операций и процедур сглаживания и включает три основных шага.

1. На первом из них на каждую «характерную» вершину проецируется соответствующий узел  $x$  ортогональной сетки. Как правило выбирается ближайший к вершине узел. В некоторых случаях целесообразней применить иные условия. В частности, если два



**Рис. 2.** Построение шестигранной сетки: а) САД-геометрия; б) «каркас» модели; в) структурированная ортогональная сетка; г) перестроенная ортогональная сетка; д) проекция сетки на «каркас» модели; е) сетка после разбиения вывернутых ячеек и вставки граничного слоя

и более узлов  $x_i$  являются ближайшими к «характерной вершине», следует выбрать узел, являющийся соседом для всех  $x_i$ .

2. Следующим шагом на каждое «характерное ребро» проецируется ближайший к нему узел шестигранной сетки.
3. Оставшиеся непокрытые участки  $\Lambda$  обрабатываются процедурой последовательной проекции узлов. Она начинается с любого узла, который уже спроецирован на  $\Lambda$ . Следующий узел  $x'$  выбирается из граничных соседей узла  $x$  в соответствии со следующими критериями:
  - граничный узел не принимается во внимание, если после проекции он будет содержать два и более вывернутых угла в прилегающем четырехугольнике, а также если проекция узла попадает на ребро прилегающего четырехугольника;
  - узлы с числом прилегающих шестигранников, превышающим четыре, имеют приоритет перед остальными;
  - среди узлов, удовлетворяющих первым двум условиям, выбирается тот, который расположен на минимальном расстоянии до «характерного» ребра.

После построения множества  $\Lambda$ , оставшиеся граничные узлы проецируются на поверхность модели. В алгоритме реализованы два подхода: трассировка лучей (для случая STL-геометрии) и итерационное смещение по значениям вокселей, применяемое как для объемного, так и поверхностного типа геометрии.

Согласно проведенным тестам реализация этих процедур позволяет полностью описать модель так, чтобы каждому «характерному» узлу был присвоен соответствующий узел сетки и каждое «характерное» ребро полностью покрывалось цепочкой ребер расчетной сетки.

Как видно из рис. 2, *д*, применение рассматриваемого подхода позволяет обеспечить точное описание исходной STL-модели.

## § 5. Улучшение качества сетки

Применяемые в алгоритме процедуры улучшения качества ячеек сетки можно разделить на две группы. К первой из них относятся операции, изменяющие структуру и число ячеек сетки, которые включают в себя:

- вставку объемного слоя граничных ячеек;
- вставку слоя граничных ячеек вдоль характерных ребер;
- перестроение вырожденных ячеек, прилегающих к характерному ребру.

Основной целью перечисленных процедур является удаление вывернутых ячеек сетки вдоль границы тела. Такие ячейки появляются в результате работы процедуры восстановления «контура модели», когда шестигранники могут иметь две или три грани, спроецированные на одну плоскость, либо два ребра, принадлежащие одному «характерному» ребру.

На рис. 2, *е* представлена полученная сетка после применения вышеназванных этапов оптимизации. Необходимо отметить, что все вывернутые углы вокруг «характерных» ребер были удалены.

Вторую группу составляют различные типы сглаживания по Лапласу (взвешенного по длине ребер, площадям квадратов и объему шестигранников), а также применяемая в более сложных случаях процедура оптимизированного сглаживания.

В данном алгоритме применяется модификация указанной процедуры для случая шестигранных ячеек. Рассматриваемый подход является итеративным и может улучшить значение любой меры ячейки  $q$ , изменяющейся в отрезке  $[-1; 1]$ . Алгоритм работает с несколькими

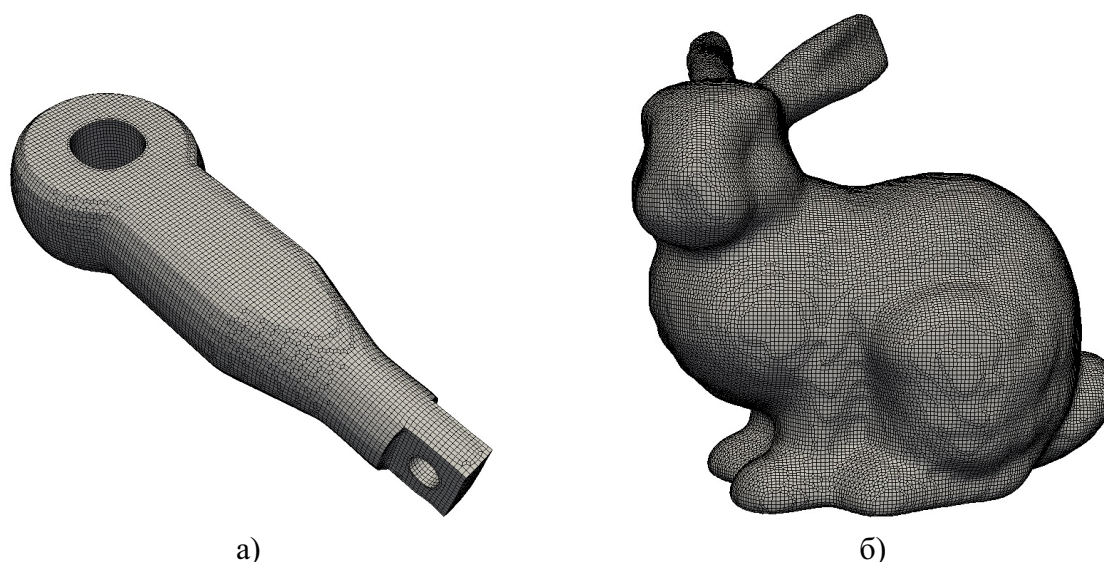


Рис. 3. Примеры построения однородных шестигранных сеток: а) стержень; б) кролик

критериями качества для шестигранника, среди которых: нормированный якобиан, обратное аспектное соотношение, скошенность, а также нормированная мера для угла искривленности четырехугольной грани.

Поскольку процедура оптимизированного сглаживания является затратной по времени, ее целесообразно применять к областям, где сглаживание по Лапласу оказалось малоэффективным.

Вид сетки после завершения этапа сглаживания, представлен на рис. 4, а.

## § 6. Тестовые примеры

Проведем сравнение результатов работы алгоритма с другими методами, описанными в [5,6,11], на примере построения однородных шестигранных сеток моделей стержня и кролика (рис. 3). В таблице 1 приведены значения мер минимального и среднего по ячейкам сетки нормированного якобиана до ( $*Q_J^{\min} \setminus *Q_J^{\text{aver}}$ ) и после проведения процедур улучшения качества сетки ( $Q_J^{\min} \setminus Q_J^{\text{aver}}$ ).

Для обеих моделей предлагаемый алгоритм имеет наибольшие значения среднего нормированного якобиана ( $*Q_J^{\text{aver}} > 0.92$  и  $Q_J^{\text{aver}} > 0.95$ ), что объясняется особенностью методов регулярной сетки, в которых шестигранники плохого качества располагаются вдоль границы области, при этом большинство внутренних ячеек имеют идеальную для проведения расчетов кубическую форму.

Наилучшие значения минимального нормированного якобиана после проведения процедур оптимизации демонстрируют алгоритмы из работ [6] для модели стержня ( $Q_J^{\min} = 0.55$ ) и [5] для модели кролика ( $Q_J^{\min} = 0.45$ ).

При этом наихудшие значения величины  $Q_J^{\min}$  имеют подходы, описанные в [11] и [6] для моделей стержня (0.04) и кролика (−0.78) соответственно. Наличие в сетке ячеек с отрицательными или близкими к нулю нормированными якобинами затрудняет ее использование в рамках конечно-элементных расчетов.

Далее рассмотрим результаты работы алгоритма для построения адаптивных сеток из поверхностного представления (STL-модели № 1 и № 2) и полученных с использованием томографического сканирования (фрагменты аорты и скелета кисти человека).

Все представленные шестигранные сетки не содержат вывернутых ячеек, значения мер



**Таблица 1.** Сравнение меры нормированного якобиана при построении шестигранных сеток различными методами

Алгоритм	Стержень		Кролик	
	$*Q_J^{\min} \setminus *Q_J^{\text{aver}}$	$Q_J^{\min} \setminus Q_J^{\text{aver}}$	$*Q_J^{\min} \setminus *Q_J^{\text{aver}}$	$Q_J^{\min} \setminus Q_J^{\text{aver}}$
Данный алгоритм	-0.49\0.92	0.2\0.95	-0.79\0.94	0.3\0.96
Gao X. et al. [11]	-\-	0.04\0.81	-\-	0.29\0.79
Takayama K. [6]	0.22\0.76	0.55\0.87	-0.77\0.74	-0.78\0.81
Livesu M. et al. [5]	0.10\0.84	0.32\0.93	-0.37\0.88	0.45\0.91

**Таблица 2.** Меры качества адаптивных шестигранных сеток

Параметры	STL-модель № 1	STL-модель № 2	Аорта	Кисть
dim	129 × 129 × 129	92 × 129 × 10	588 × 292 × 1025	513 × 360 × 177
$h$	4	2	8	2
$N$	25438	14430	33249	293430
$C$	31062	10916	27488	252789
$\Theta_w^{\max}$	69°	108°	101°	75°
$Q_J^{\min}$	0.2	0.2	0.2	0.2
$Q_a^{\max}$	11	8	11	12
$Q_s^{\max}$	0.86	0.85	0.87	0.87

качества сеток приведены в таблице 2.

Значительное в сравнении с другими примерами число узлов  $N = 293\,430$  и ячеек  $C = 252\,789$  шестигранной сетки модели человеческой кисти объясняется выбором малого шага между вокселями  $h = 2$ , а также большой размерностью массива объемных данных  $\text{dim} = 513 \times 360 \times 177$ .

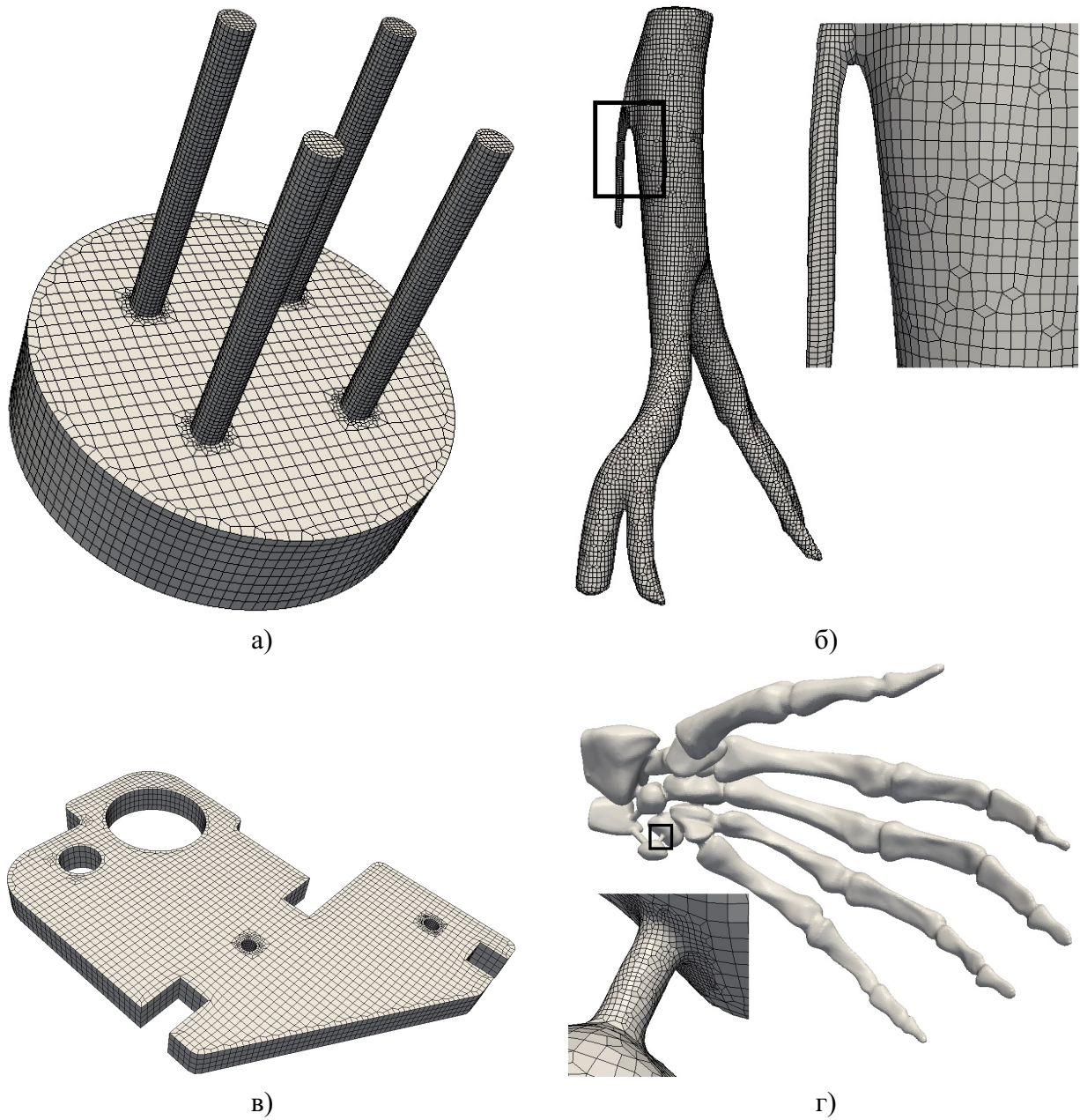
Минимальный нормированный якобиан  $Q_J^{\min}$  демонстрируемых сеток имеет значение  $Q_J^{\min} \geq 0.2$ , что приемлемо для конечного-элементных алгоритмов. Данная мера была выбрана в качестве контрольной в ходе работы процедуры оптимизированного сглаживания. Доля шестигранников, имеющих  $Q_J \approx 1$ , составила более 41 %, а ячеек с  $Q_J > 0.8$  — более 82 %.

Относительно большое значение ( $Q_a^{\max} > 8$ ) максимального аспектного отношения всех сеток  $Q_a^{\max}$  объясняется применением шаблонов согласующего перестроения, к ячейкам, имеющим один, два или четыре «помеченных» узла.

Максимальные значения скошенности  $Q_s^{\max}$  и угла искривленности четырехугольных граней  $\Theta_w^{\max}$  находятся в диапазонах  $[0.85, 0.87]$  и  $[69^\circ, 108^\circ]$  соответственно, что обусловлено наличием криволинейных поверхностей в описываемых геометриях, а также выбором значения нормированного якобиана в качестве основного контрольного критерия при работе процедуры оптимизированного сглаживания.

## § 7. Заключение

В работе представлена модификация ранее предложенного воксельного генератора шестигранной сетки для построения адаптивных сеток геометрий, имеющих участки относительно малых размеров. Разработаны следующие изменения в функционале.



**Рис. 4.** Примеры построения адаптивных неструктурированных шестигранных сеток: а) STL-модель № 1; б) фрагмент аорты; в) STL-модель № 2; г) скелет кисти человека

1. Применен критерий оценки величины относительного объема тела  $\Omega$ , содержащегося внутри шестигранника  $c$ , для построения однородной ортогональной сетки. Вычисление значения  $\mathcal{V}_c^\Omega$  осуществляется на основе отношения части точек, находящихся внутри тела  $\Omega$ , к общему числу точек, равномерно распределенных внутри  $c$  (метод Монте–Карло).

2. Разработана процедура перестроения ячеек сетки, описывающих участки малых геометрических размеров. Формирование списка  $\Upsilon$  шестигранников, подлежащих разбиению, основывается на требовании наличия не менее трех слоев ячеек внутри любого участка модели, описываемого сеткой. Для создания адаптивной сетки использованы шаблоны разбиения по «трем» ячейкам Р. Шнейдера. После завершения процедуры перестроения применен дополнительный анализ ячеек сетки на предмет удовлетворения величины  $\mathcal{V}_c^\Omega$  заданному пороговому значению  $\varepsilon'$ .

3. Модифицирован процесс восстановления каркаса модели в части проецирования на каждое «характерное ребро» ближайшего к ней узла шестигранной сетки (без задания дополнительных параметров), а также добавления проверки количества прилегающих к узлу шестигранников (больше четырех) в процедуру последовательной проекции узлов на непокрытые участки  $\Lambda$ .

Внедрение нового функционала позволяет генерировать адаптивную сетку, точно описывающую исходную геометрию, заданную как в объемном, так и поверхностном формате данных. Алгоритм протестирован на достаточном количестве моделей различной сложности.

Дальнейшие исследования направлены на развитие сферы применения алгоритма для построения адаптивных шестигранных сеток на основе оценки степени кривизны поверхности и проведение анализа робастности генератора на коллекциях моделей из работ [5].

#### СПИСОК ЛИТЕРАТУРЫ

1. Ray N., Sokolov D., Reberol M., Ledoux F., Lévy B. Hex-dominant meshing: Mind the gap! // Computer-Aided Design. 2018. Vol. 102. P. 94–103. <https://doi.org/10.1016/j.cad.2018.04.012>
2. Pellerin J., Johnen A., Verhetsel K., Remacle J.-F. Identifying combinations of tetrahedra into hexahedra: A vertex based strategy // Computer-Aided Design. 2018. Vol. 105. P. 1–10. <https://doi.org/10.1016/j.cad.2018.05.004>
3. Ruiz-Gironés E., Roca X., Sarrate J. The receding front method applied to hexahedral mesh generation of exterior domains // Engineering with Computers. 2012. Vol. 28. Issue 4. P. 391–408. <https://doi.org/10.1007/s00366-011-0233-y>
4. Bo Ren, Cheng Wang. Mapping-based 3D hexahedral finite element mesh generation method // Wuhan University Journal of Natural Sciences. 2007. Vol. 12. Issue 2. P. 255–259. <https://doi.org/10.1007/s11859-006-0025-1>
5. Livesu M., Pietroni N., Puppo E., Sheffer A., Cignoni P. LoopyCuts: practical feature-preserving block decomposition for strongly hex-dominant meshing // ACM Transactions on Graphics. 2020. Vol. 39. Issue 4. Article No.: 121. P. 121:1–121:17. <https://doi.org/10.1145/3386569.3392472>
6. Takayama K. Dual sheet meshing: An interactive approach to robust hexahedralization // Computer Graphics Forum. 2019. Vol. 38. Issue 2. P. 37–48. <https://doi.org/10.1111/cgf.13617>
7. Ledoux F., Weill J.-C. An extension of the reliable whisker weaving algorithm // Proceedings of the 16th International Meshing Roundtable. Berlin–Heidelberg: Springer, 2008. P. 215–232. [https://doi.org/10.1007/978-3-540-75103-8\\_13](https://doi.org/10.1007/978-3-540-75103-8_13)
8. Schneiders R., Bünten R. Automatic generation of hexahedral finite element meshes // Computer Aided Geometric Design. 1995. Vol. 12. Issue 7. P. 693–707. [https://doi.org/10.1016/0167-8396\(95\)00013-V](https://doi.org/10.1016/0167-8396(95)00013-V)
9. Lili Huang, Guoqun Zhao, Zhonglei Wang, Xiangwei Zhang. Adaptive hexahedral mesh generation and regeneration using an improved grid-based method // Advances in Engineering Software. 2016. Vol. 102. P. 49–70. <https://doi.org/10.1016/j.advengsoft.2016.09.004>

10. Zhang Yongjie, Bajaj C. Adaptive and quality quadrilateral/hexahedral meshing from volumetric data // *Computer Methods in Applied Mechanics and Engineering*. 2006. Vol. 195. Issues 9–12. P. 942–960. <https://doi.org/10.1016/j.cma.2005.02.016>
11. Gao Xifeng, Shen Hanxiao, Panozzo D. Feature preserving octree-based hexahedral meshing // *Computer Graphics Forum*. 2019. Vol. 38. Issue 5. P. 135–149. <https://doi.org/10.1111/cgf.13795>
12. Караваев А. С., Копысов С. П. Модификация генератора шестигранных сеток, основанного на воксельном представлении геометрии // *Вестник Удмуртского университета. Математика. Механика. Компьютерные науки*. 2020. Т. 30. Вып. 3. С. 468–479. <https://doi.org/10.35634/vm200308>
13. Karavaev A. S., Kopysov S. P. Hexahedral mesh generation using voxel field recovery // *Numerical Geometry, Grid Generation and Scientific Computing*. Cham: Springer, 2020. P. 295–305. [https://doi.org/10.1007/978-3-030-76798-3\\_19](https://doi.org/10.1007/978-3-030-76798-3_19)
14. Ito Yasushi, Shih A. M., Soni B. K. Octree-based reasonable-quality hexahedral mesh generation using a new set of refinement templates // *International Journal for Numerical Methods in Engineering*. 2009. Vol. 77. Issue 13. P. 1809–1833. <https://doi.org/10.1002/nme.2470>
15. Tchou Ko-Foa, Khachan M., Guibault F., Camarero R. Three-dimensional anisotropic geometric metrics based on local domain curvature and thickness // *Computer-Aided Design*. 2005. Vol. 37. Issue 2. P. 173–187. <https://doi.org/10.1016/j.cad.2004.05.007>
16. Qian Jin, Zhang Yongjie. Automatic unstructured all-hexahedral mesh generation from B-Reps for non-manifold CAD assemblies // *Engineering with Computers*. 2012. Vol. 28. Issue 4. P. 345–359. <https://doi.org/10.1007/s00366-011-0232-z>
17. Sun Lu, Zhao Guoqun, Ma Xinwu. Adaptive generation and local refinement methods of three-dimensional hexahedral element mesh // *Finite Elements in Analysis and Design*. 2012. Vol. 50. P. 184–200. <https://doi.org/10.1016/j.finel.2011.09.009>
18. Owen S. J., Shih R. M. A template-based approach for parallel hexahedral two-refinement // *Procedia Engineering*. 2015. Vol. 124. P. 31–43. <https://doi.org/10.1016/j.proeng.2015.10.120>

Поступила в редакцию 17.03.2023

Принята к публикации 01.07.2023

Караваев Александр Сергеевич, к. ф.-м. н., научный сотрудник, кафедра вычислительной механики, Удмуртский государственный университет, 426034, Россия, г. Ижевск, ул. Университетская, 1.

ORCID: <https://orcid.org/0009-0000-6510-8483>

E-mail: [karavaev-alexander@yandex.ru](mailto:karavaev-alexander@yandex.ru)

Копысов Сергей Петрович, д. ф.-м. н., профессор, кафедра вычислительной механики, Удмуртский государственный университет, 426034, Россия, г. Ижевск, ул. Университетская, 1.

ORCID: <https://orcid.org/0000-0003-2511-5894>

E-mail: [s.kopysov@gmail.com](mailto:s.kopysov@gmail.com)

**Цитирование:** А. С. Караваев, С. П. Копысов. Построение адаптивных шестигранных сеток из поверхностной и воксельной геометрических моделей // *Вестник Удмуртского университета. Математика. Механика. Компьютерные науки*. 2023. Т. 33. Вып. 3. С. 534–547.

**A. S. Karavaev, S. P. Kopysov**

### **Generation of adaptive hexahedral meshes from surface and voxel geometric models**

*Keywords:* hexahedral mesh generator, mesh refinement, volume data, CAD model, STL-geometry.

MSC2020: 65M50

DOI: [10.35634/vm230310](https://doi.org/10.35634/vm230310)

We present a modification of the developed hexahedral mesh generator from voxel data which allows constructing adaptive computational meshes. Construction of the refinement field is based on geometry features of the described model when it has a large thickness difference in dimensions or small and thin areas. A universal criterion for cells refinement is proposed which gives the possibility of its use in the case of volumetric (voxel) and surface (STL) representations of the model geometry. The refinement templates that provide conformal mesh closure are described. The results of the algorithm performance are given.

#### REFERENCES

1. Ray N., Sokolov D., Reberol M., Ledoux F., Lévy B. Hex-dominant meshing: Mind the gap!, *Computer-Aided Design*, 2018, vol. 102, pp. 94–103. <https://doi.org/10.1016/j.cad.2018.04.012>
2. Pellerin J., Johnen A., Verhetsel K., Remacle J.-F. Identifying combinations of tetrahedra into hexahedra: A vertex based strategy, *Computer-Aided Design*, 2018, vol. 105, pp. 1–10. <https://doi.org/10.1016/j.cad.2018.05.004>
3. Ruiz-Gironés E., Roca X., Sarrate J. The receding front method applied to hexahedral mesh generation of exterior domains, *Engineering with Computers*, 2012, vol. 28, issue 4, pp. 391–408. <https://doi.org/10.1007/s00366-011-0233-y>
4. Bo Ren, Cheng Wang. Mapping-based 3D hexahedral finite element mesh generation method, *Wuhan University Journal of Natural Sciences*, 2007, vol. 12, issue 2, pp. 255–259. <https://doi.org/10.1007/s11859-006-0025-1>
5. Livesu M., Pietroni N., Puppo E., Sheffer A., Cignoni P. LoopyCuts: practical feature-preserving block decomposition for strongly hex-dominant meshing, *ACM Transactions on Graphics*, 2020, vol. 39, issue 4, article no.: 121, pp. 121:1–121:17. <https://doi.org/10.1145/3386569.3392472>
6. Takayama K. Dual sheet meshing: An interactive approach to robust hexahedralization, *Computer Graphics Forum*, 2019, vol. 38, issue 2, pp. 37–48. <https://doi.org/10.1111/cgf.13617>
7. Ledoux F., Weill J.-C. An extension of the reliable whisker weaving algorithm, *Proceedings of the 16th International Meshing Roundtable*, Berlin–Heidelberg: Springer, 2008, pp. 215–232. [https://doi.org/10.1007/978-3-540-75103-8\\_13](https://doi.org/10.1007/978-3-540-75103-8_13)
8. Schneiders R., Bünten R. Automatic generation of hexahedral finite element meshes, *Computer Aided Geometric Design*, 1995, vol. 12, issue 7, pp. 693–707. [https://doi.org/10.1016/0167-8396\(95\)00013-V](https://doi.org/10.1016/0167-8396(95)00013-V)
9. Lili Huang, Guoqun Zhao, Zhonglei Wang, Xiangwei Zhang. Adaptive hexahedral mesh generation and regeneration using an improved grid-based method, *Advances in Engineering Software*, 2016, vol. 102, pp. 49–70. <https://doi.org/10.1016/j.advengsoft.2016.09.004>
10. Zhang Yongjie, Bajaj C. Adaptive and quality quadrilateral/hexahedral meshing from volumetric data, *Computer Methods in Applied Mechanics and Engineering*, 2006, vol. 195, issues 9–12, pp. 942–960. <https://doi.org/10.1016/j.cma.2005.02.016>
11. Gao Xifeng, Shen Hanxiao, Panozzo D. Feature preserving octree-based hexahedral meshing, *Computer Graphics Forum*, 2019, vol. 38, issue 5, pp. 135–149. <https://doi.org/10.1111/cgf.13795>
12. Karavaev A.S., Kopysov S.P. A modification of the hexahedral mesh generator based on voxel geometry representation, *Vestnik Udmurtskogo Universiteta. Matematika. Mekhanika. Komp'yuternye Nauki*, 2020, vol. 30, issue 3, pp. 468–479. <https://doi.org/10.35634/vm200308>

13. Karavaev A. S., Kopysov S. P. Hexahedral mesh generation using voxel field recovery, *Numerical Geometry, Grid Generation and Scientific Computing*, Cham: Springer, 2020, pp. 295–305. [https://doi.org/10.1007/978-3-030-76798-3\\_19](https://doi.org/10.1007/978-3-030-76798-3_19)
14. Ito Yasushi, Shih A. M., Soni B. K. Octree-based reasonable-quality hexahedral mesh generation using a new set of refinement templates, *International Journal for Numerical Methods in Engineering*, 2009, vol. 77, issue 13, pp. 1809–1833. <https://doi.org/10.1002/nme.2470>
15. Tchou Ko-Foa, Khachan M., Guibault F., Camarero R. Three-dimensional anisotropic geometric metrics based on local domain curvature and thickness, *Computer-Aided Design*, 2005, vol. 37, issue 2, pp. 173–187. <https://doi.org/10.1016/j.cad.2004.05.007>
16. Qian Jin, Zhang Yongjie. Automatic unstructured all-hexahedral mesh generation from B-Reps for non-manifold CAD assemblies, *Engineering with Computers*, 2012, vol. 28, issue 4, pp. 345–359. <https://doi.org/10.1007/s00366-011-0232-z>
17. Sun Lu, Zhao Guoqun, Ma Xinwu. Adaptive generation and local refinement methods of three-dimensional hexahedral element mesh, *Finite Elements in Analysis and Design*, 2012, vol. 50, pp. 184–200. <https://doi.org/10.1016/j.finel.2011.09.009>
18. Owen S. J., Shih R. M. A template-based approach for parallel hexahedral two-refinement, *Procedia Engineering*, 2015, vol. 124, pp. 31–43. <https://doi.org/10.1016/j.proeng.2015.10.120>

Received 17.03.2023

Accepted 01.07.2023

Aleksandr Sergeevich Karavaev, Candidate of Physics and Mathematics, Researcher, Department of Computational Mechanics, Udmurt State University, ul. Universitetskaya, 1, Izhevsk, 426034, Russia.

ORCID: <https://orcid.org/0009-0000-6510-8483>

E-mail: [karavaev-alexander@yandex.ru](mailto:karavaev-alexander@yandex.ru)

Sergei Petrovich Kopysov, Doctor of Physics and Mathematics, Professor, Department of Computational Mechanics, Udmurt State University, ul. Universitetskaya, 1, Izhevsk, 426034, Russia.

ORCID: <https://orcid.org/0000-0003-2511-5894>

E-mail: [s.kopysov@gmail.com](mailto:s.kopysov@gmail.com)

**Citation:** A. S. Karavaev, S. P. Kopysov. Generation of adaptive hexahedral meshes from surface and voxel geometric models, *Vestnik Udmurtskogo Universiteta. Matematika. Mekhanika. Komp'yuternye Nauki*, 2023, vol. 33, issue 3, pp. 534–547.