

Удаление самопересечений поверхностной неструктурированной сетки в задаче моделирования обледенения

А. А. Рыбаков

ФГБУ «Национальный исследовательский центр «Курчатовский институт», г. Москва, Российская Федерация

Адрес: 123182, Российская Федерация, г. Москва, пл. Академика Курчатова, д. 1
rybakov_aan@nrcki.ru

Аннотация

При численном решении задачи моделирования обледенения ключевым аспектом является адекватное представление поверхности образующегося ледяного нароста. Как правило, форма ледяного нароста описывается с помощью поверхностной неструктурированной сетки, которая эволюционирует во времени в соответствии с внешними условиями обледенения. Поверхностная сетка разделяет расчетную область на внутреннюю часть, представляющую собой область льда, и внешнюю часть. Таким образом, описывающая границу льда поверхностная сетка является двусторонней с определенным направлением роста льда в ее ячейках. При эволюции сетки возникновение самопересечений является критическим дефектом, приводящим к невозможности продолжения расчетов. Возникающие самопересечения должны быть удалены для продолжения вычислений. В работе предлагается метод удаления самопересечений поверхностной неструктурированной сетки, основанный на обходе внешней поверхности сетки и обработке пересечений ее ячеек в процессе обхода. Для ускорения обработки сетки используется BVH-дерево ее ячеек. Для избавления от ошибок точности, связанных с использованием операций над вещественными числами, используются вычисления в рациональных числах, в которых задачи поиска пересечения линейных геометрических примитивов решаются без потери точности. Предложенный метод удаления самопересечений поверхностной неструктурированной сетки реализован для обработки псевдотрехмерных профилей в рамках программного комплекса «Кристалл» моделирования обледенения летательных аппаратов.

Ключевые слова: моделирование обледенения, поверхностная неструктурированная сетка, самопересечение, удаление самопересечений, обход внешней поверхности

Финансирование: Работа выполнена в рамках государственного задания НИЦ «Курчатовский институт» (№ 1240531000453). Результаты получены с использованием оборудования Центра коллективного пользования вычислительными ресурсами МСЦ НИЦ «Курчатовский институт». При проведении исследований использовался суперкомпьютер MBC-10П, установленный в НИЦ «Курчатовский институт».

Конфликт интересов: автор заявляет об отсутствии конфликта интересов.

Для цитирования: Рыбаков А. А. Удаление самопересечений поверхностной неструктурированной сетки в задаче моделирования обледенения // Современные информационные технологии и ИТ-образование. 2026. Т. 22, № 1. С. 63-75. <https://doi.org/10.25559/SITITO.022.202601.63-75>

© Рыбаков А. А., 2026



Контент доступен под лицензией Creative Commons Attribution 4.0 License.
The content is available under Creative Commons Attribution 4.0 License.



Self-Intersections Elimination of a Surface Unstructured Mesh in the Problem of Icing Modeling

A. A. Rybakov

National Research center “Kurchatov Institute for Theoretical and Experimental physics”, Moscow, Russian Federation

Address: 1 Akademicheskaya Kurchatova Sq., Moscow 123182, Russian Federation

rybakov_aan@nrcki.ru

Abstract

While numerically solving icing modeling problem, a key aspect is adequately representing the surface of the developing ice accretion. Typically, the shape of the ice accretion is described using an unstructured surface mesh that evolves over time in response to external icing conditions. The surface mesh divides the computational scope into an inner part, representing the ice region, and an outer part. Thus, the surface mesh describing the ice boundary is two-sided, with a defined direction of ice growth within its cells. During mesh evolution, the occurrence of self-intersections is a critical defect, preventing further calculations. These self-intersections must be removed to continue the computations. This paper proposes a method for removing self-intersections from an unstructured surface mesh. This method traverses the outer surface of the mesh and processes the intersections of its cells during the traversal. To accelerate mesh processing, a BVH tree of its cells is used. To eliminate precision errors associated with operations on real numbers, calculations in rational numbers are used, which solve problems of finding the intersection of linear geometric primitives without loss of accuracy. The proposed method for removing self-intersections of an unstructured surface mesh has been implemented for processing pseudo-3D profiles within the «Crystal» software suite for aircraft icing modeling.

Keywords: icing modeling, surface unstructured mesh, self-intersection, self-intersection elimination, external surface traversal

Funding: This work was supported by state assignment of the National Research Centre “Kurchatov Institute” (124053100045-3). The results were obtained using equipment of the Centre for Shared Use of Computing Resources of Joint Supercomputer Centre National Research Centre “Kurchatov Institute”. The MVS-10P supercomputer installed at the National Research Centre “Kurchatov Institute” was used in the research.

Conflict of interests: The author declares no conflict of interests.

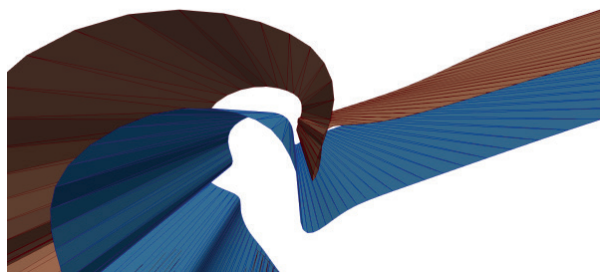
For citation: Rybakov A.A. Self-Intersections Elimination of a Surface Unstructured Mesh in the Problem of Icing Modeling. *Modern Information Technologies and IT-Education*. 2026;22(1):63-75. <https://doi.org/10.25559/SITITO.022.202601.63-75>



Введение

Задача моделирования обледенения летательных аппаратов является критически важной для обеспечения безопасности полетов [1]. Известны случаи, когда обледенение несущих частей и силовых установок летательных аппаратов приводило к авиакатастрофам с большим количеством человеческих жертв. В настоящее время существует большое количество программных пакетов для моделирования обледенения. К таким решениям относятся пакет инженерного проектирования ANSYS (включающий в себя модули FENSAP-ICE, DROP3D, ICE3D) [2], модуль iceFoam [3] как часть открытого программного обеспечения OpenFOAM, решение для моделирования обледенения в составе пакета инженерного анализа ЛОГОС [4, 5], модуль IceVision [6] в составе пакета FlowVision, а также ряд других пакетов решений, таких как LEWICE [7], AIPAC [8], ONERA [9], TRAJICE [10].

Как правило, поверхность образующегося ледяного налета описывается с помощью поверхностной неструктурированной сетки. Перестроение этой сетки в соответствии с объемом накопленного льда в ее ячейках является одной из центральных задач при моделировании обледенения [11]. Одной из проблем, связанных с эволюцией поверхностной сетки, является эффект сохранения и накопления локальных дефектов (острые пики или впадины), развитие которых приводит к ненадежному поведению программы и необходимости выполнять повторные запуски для получения результатов на протяжении времени моделирования. Такие перезапуски программы приводят к потере производительности при проведении расчетов. В различных приложениях проблема накопления локальных дефектов сетки преодолевается с помощью дополнительных эвристических алгоритмов сглаживания сетки, как это описано в [12, 13], локальной корректировки для сглаживания отдельных острых углов и впадин с помощью вспомогательного зависящего от значения угла коэффициента, как приведено в [14], и другими искусственными методами. В работе [15] предлагается подход, в котором сглаживание сетки является свойством метода перестроения.



Р и с 1. Возникновение самопересечения при перестроении сетки

Fig. 1. Self-intersection occurs when rebuilding the mesh

Источник: здесь и далее в статье все таблицы и рисунки составлены автором.

Source: Hereinafter in this article all tables and figures were made by the author.

Во время эволюции поверхностной сетки применение сглаживаний может отложить возникновение самопересечений,

но полностью их исключить нельзя (см. рис. 1), поэтому возникает необходимость обработки этого глобального дефекта для поддержания сетки в корректном состоянии описания замкнутой двусторонней поверхности. В статье рассматриваются существующие методы удаления самопересечений поверхностной неструктурированной сетки. Предлагается метод удаления самопересечений поверхностной неструктурированной сетки с помощью дробления ячеек по всем точкам и отрезкам пересечения с другими ячейками, выполняемого в процессе обхода внешней поверхности, после чего все необойденные ячейки классифицируются как лишние и впоследствии удаляются. Рассматривается применение метода удаления самопересечений для простых сеток и в общем случае, в котором вычисления по поиску точек и отрезков пересечения выполняются в рациональных координатах.

Удаление самопересечений поверхностной сетки

Различные методы сглаживания поверхностной сетки пригодны для устранения локальных дефектов, тогда как для удаления самопересечений необходимы принципиально другие подходы [16]. Удаление самопересечений будем рассматривать для трехмерной замкнутой поверхностной неструктурированной сетки с треугольными ячейками, для такой сетки заданы направления внешних нормалей ячеек, эта сетка разделяет пространство на внутреннюю и внешнюю области.

Одним из известных подходов к реконструкции поверхностной сетки является использование построенной тетраэдральной объемной сетки в расчетной области. Этот метод впервые был описан в [17], после чего были разработаны его усовершенствованные версии, направленные на компенсацию ошибок вычислений с плавающей точкой [18], а также на ускорение обработки [19]. Узким местом такого подхода является необходимость построения тетраэдральной объемной сетки в расчетной области, что связано с дополнительными вычислительными затратами, поэтому рассматриваются методы, работающие непосредственно на поверхностной сетке.

Среди ранних работ по поиску и удалению самопересечений (или пересечений различных сеток), работающих непосредственно на поверхностной сетке, следует отметить следующие. Рассмотрение пересечений в виде замкнутых ломаных и идентификация пересечений с помощью обхода ломаных на базе алгоритма TNOIT (*Tracing of the Neighbours Of Intersecting Triangles*) [20], при этом обычно рассматриваются непересекающиеся ломаные, которые могут быть обработаны независимо. Для поиска стартовых точек обхода ломаных пересечений эффективным оказывается использование BVH-деревьев (*Bounding Volume Hierarchy*) [21], позволяющее быстро находить пары потенциально пересекающихся ячеек. Обход ломаной пересечения порождает задачу дробления ячеек сетки на более мелкие таким образом, чтобы звенья ломаной пересечения являлись ребрами ячеек (триангуляция конфликтных ячеек по ломаной).

Центральной проблемой обработки пересечений сеток является определение пересечения пары ячеек – двух треугольников в пространстве. Прямое решение этой задачи может выполняться с ошибками из-за вычислений с плавающей



точкой (при использовании вещественных координат узлов). Из-за погрешности вычислений пересечение пары ячеек может быть определено некорректно. Для уточнения ситуаций, в которых проблема погрешности вычислений стоит особенно остро, введем понятие простой сетки. Трехмерную замкнутую поверхностную неструктурированную сетку с треугольными ячейками будем называть простой, если в ней отсутствуют конфликты между элементами, которые могут быть некорректно определены из-за погрешности вычислений. А именно: никакие два узла не совпадают, никакой узел не лежит на неинцидентном ему ребре, никакой узел не лежит на неинцидентной ему ячейке, никакие два несмежных ребра не пересекаются (ни по точке, ни по отрезку), отсутствует наложение ячеек. По аналогии можно определить простое взаимное расположение двух сеток. При использовании простых сеток с некоторым допустимым значением погрешности вычислений задача обработки пересечений и самопересечений решается гораздо проще. В общем случае, применяются различные подходы для компенсации погрешности вычислений. Среди них локальная коррекция сетки для приведения к простому виду [22], компенсация ошибок с помощью введения дополнительных контрольных кодов классификации пересечения пар ячеек и восстановления отдельных потерянных звеньев ломаной пересечения [23], использование рациональной арифметики для обеспечения вычислений с абсолютной точностью [24]. Среди современных методов удаления самопересечений отметим два наиболее развитых метода, основанных на поиске пересечений всех пар ячеек в сетке и дроблении ячеек по точкам и отрезкам пересечения: Exact and Efficient Intersection Resolution (EEIR) [25] и Fast and Robust Mesh Arrangements (FRMA) [26]. В этих методах вычисления выполняются с использованием чисел с плавающей точкой, а для компенсации ошибок вычислений используются неявные представления точек пересечения через другие точки, чьи координаты заданы точно: представление пересечения двух прямых Line-Line Intersection (LLI), представление пересечения прямой и плоскости Line-Plane Intersection (LPI), представление пересечения трех плоскостей Triplet-Plane Intersection (TPI). В работе [24] отмечено, что такой подход не гарантирует полностью избавление от ошибок точности, поэтому в ней рассмотрено применение рациональных координат, а для сокращения количества пар конфликтующих ячеек применяется предобработка сетки.

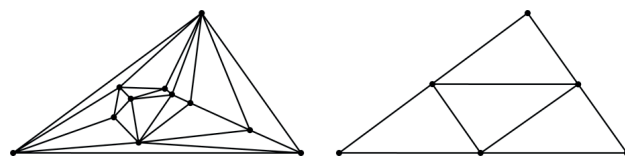
При рассмотрении удаления самопересечений сетки отметим основные особенности решения этой задачи. Самопересечения могут состоять из произвольных ломаных (в том числе пересекающихся) и плоских объектов (не исключено наложение ячеек). Сетка не может рассматриваться как простая, локальная коррекция не обеспечивает гарантированного успеха приведения сетки к простому виду. Так как ячейки сетки имеют определенные направления внешней нормали, то может быть определена внешняя поверхность сетки, разделяющая пространство на внутреннюю и внешнюю области, при этом во внешней области не будет содержаться ни одной ячейки, а все лишние ячейки будут находиться во внутренней области и являться претендентами на удаление. Наличие определенных внешних нормалей ячеек позволяет использовать это для развития нового метода поиска и удаления самопересечений, так

как требуется поиск только самопересечений, находящихся на внешней поверхности сетки, что может быть использовано для повышения производительности вычислений.

Адаптация сетки

Методы перестроения поверхностной сетки объединяет одно сходство – они сохраняют количество элементов сетки (узлы, ребра, ячейки) и связи между ними. Несмотря на некоторые специальные методы по предотвращению конфликтов между ячейками сетки и методы сглаживания, после формирования новой поверхности возможно возникновение самопересечений, появление ячеек неправильной формы, а также неравномерное распределение ячеек в сетке по размеру. Пока опустим самопересечения сетки и будем рассматривать только вопросы, касающиеся формы и размера ячеек.

Рассмотрим операцию разбиения ячейки на более мелкие. В качестве примера такого разбиения можно рассмотреть триангуляцию Делоне по заданному набору точек разбиения внутри ячейки (рис. 2, слева) [27]. Триангуляция Делоне обладает свойством максимизировать минимальный угол из всех углов треугольников, образованных в результате триангуляции. Таким образом, триангуляция Делоне позволяет избежать треугольников с очень острыми углами, возникновение которых может привести к нестабильности вычислений на поверхностной расчетной сетке. Стоит отметить, что разбиение ячейки можно производить с сохранением локальной кривизны сетки, как это показано в [28], но мы этот вариант рассматривать не будем, а будем производить разбиение исключительно в плоскости ячейки. Если точка разбиения находится не внутри ячейки, а на ее ребре, то разбить придется также вторую инцидентную ячейку этого ребра (если такая есть). Иногда для уменьшения размера требуется просто разбить ячейку на более мелкие без задания точек разбиения. В этом случае необходимо следить за качеством результирующих ячеек. Для определения качества ячейки треугольной формы $f = ABC$ можно пользоваться простым показателем качества $Q(f) = 4\sqrt{3}S_f / (AB^2 + BC^2 + AC^2)$, где $Q(f) = 1$ соответствует идеальному случаю равносторонней ячейки, а $Q(f) = 0$ – худший случай для ячеек с нулевой площадью [29]. Для выполнения разбиения ячейки на более мелкие с сохранением их качества можно просто выполнить разбиение по серединам всех ее сторон (рис. 2, справа).



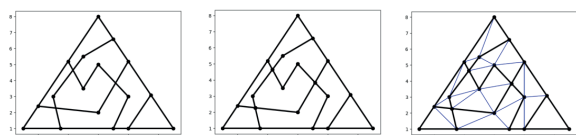
Р и с. 2. Разбиение ячейки на более мелкие треугольники

F i g. 2. Splitting a cell into smaller triangles

В задаче удаления самопересечений поверхностной неструктурированной сетки важна специфическая операция дробления ячейки – дробление по фиксированному множеству отрезков (для простой сетки) и по фиксированному множеству отрезков и точек (в общем случае). На рис. 3 слева показана ячейка, которая пересекается с расчетной сеткой по четырем отмеченным траекториям – ломаным (некоторые из этих

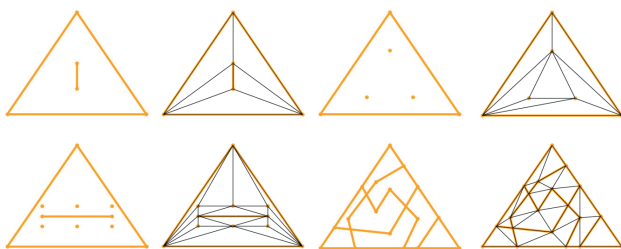


траекторий пересекаются друг с другом). Естественно, при выполнении триангуляции этой ячейки не должно появляться новых отрезков, которые пересекают один из отрезков какой-либо траектории пересечения с сеткой. Такая операция может быть выполнена с помощью модификации триангуляции Делоне – триангуляции Делоне с ограничениями [30, 31]. Если мы предполагаем, что при удалении самопересечений сетки требуется выполнять дробление ячейки по небольшому количеству точек и отрезков, то триангуляцию можно выполнять по наиболее простому жадному алгоритму с помощью последовательного добавления новых отрезков в триангуляцию. На рис. 3 приведен пример триангуляции по множеству отрезков, а на рис. 4 – по множеству отрезков и точек.



Р и с. 3. Разбиение ячейки на более мелкие по множеству фиксированных отрезков

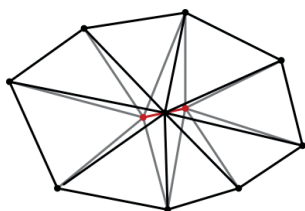
F i g. 3. Splitting a cell into smaller ones using a set of fixed segments



Р и с. 4. Разбиение ячейки на более мелкие по множеству фиксированных отрезков и точек

F i g. 4. Splitting a cell into smaller ones using a set of fixed segments and points

Триангуляция ячейки выполняется следующим образом. Сначала ищутся все точки пересечения траекторий друг с другом и выполняется дробление траекторий по этим точкам (рис. 3 в центре). После этого все отрезки траекторий считаются фиксированными и входят в финальную триангуляцию. Для завершения триангуляции к ним может понадобиться добавить другие отрезки, соединяющие между собой некоторые точки траекторий (на рис. 3 справа они отмечены синим цветом). Такую операцию дробления ячейки будем использовать в дальнейшем. В общем случае триангуляцию требуется выполнять по множеству фиксированных отрезков и точек, примеры таких триангуляций показаны на рис. 4.



Р и с. 5. Удаление короткого ребра

F i g. 5. Removal of short edge

Вторая операция, которая необходима для адаптации сетки, связана с огрублением. Достаточно часто во время выполнения триангуляции по множеству заданных точек могут появляться ячейки с низким показателем качества (у них могут присутствовать либо слишком острые углы, либо близкий к развернутому угол). Если в ячейке присутствует угол, близкий к развернутому, то избавиться от него поможет разбиение по наибольшей стороне (при этом в качестве точки разбиения следует выбрать основание высоты, опущенной из противоположного узла). Если же в треугольнике присутствуют лишь близкие к острым углы, то значит в нем есть слишком короткая сторона, которая может быть удалена. При удалении ребра AB удаляются обе инцидентные ей ячейки, узлы A и B соединяются в единый узел A' , а все инцидентные им ребра $E(A) \cup E(B)$ перенаправляются на узел A' с учетом удаления кратных ребер (рис. 5). Эту операцию будем называть стягиванием ребра [32]. Путем применения стягивания наиболее коротких ребер в расчетной сетке можно добиться произвольной степени огрубления.

Поиск пересекающихся ячеек

Для поиска самопересечения необходимо определить, какое отношение двух ячеек сетки можно трактовать как самопересечение. Конечно, простое наличие общих точек у двух ячеек не может служить критерием самопересечения, так как у каждой ячейки есть смежные ячейки (то есть две ячейки могут иметь как общую вершину, так и общее ребро). Так как смежными могут быть только ячейки, имеющие общие инцидентные объекты (общую инцидентную вершину или общее инцидентное ребро), а все отношения инцидентности зафиксированы в расчетной сетке, то не представляет труда отделить пересечение ячеек по общему инцидентному объекту от самопересечения. Отметим, что две смежные ячейки, имеющие одну общую вершину, могут быть пересекающимися. Аналогично две смежные ячейки, имеющие общее ребро, также могут пересекаться (в случае их наложения).

В общем случае для идентификации всех фактов самопересечения сетки требуется проанализировать все пары ячеек и проверить пересечение ячеек в паре (то есть проверить пересечение каждой ячейки с каждой). Так как прямой перебор всех пар ячеек имеет квадратичную сложность по количеству ячеек, то его использование на крупных сетках невозможно. В этом случае целесообразно выполнять поиск пар пересекающихся ячеек с помощью BVH-деревьев [33], в которых множество ячеек представлено в виде древовидной структуры, связанной с ограничением геометрических объектов прямоугольными параллелепипедами в пространстве. Прямоугольный параллелепипед будем определять как геометрическое место точек $P = (x, y, z)$ в пространстве, удовлетворяющих условиям $x_l \leq x \leq x_h$, $y_l \leq y \leq y_h$, $z_l \leq z \leq z_h$. Для такого прямоугольного параллелепипеда будем использовать обозначение $\text{Box}([x_l, x_h], [y_l, y_h], [z_l, z_h])$.

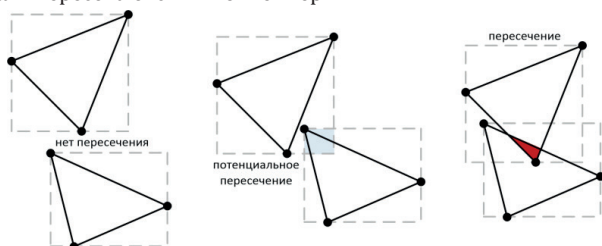
Контейнером произвольного множества точек M в пространстве, который будем обозначать $[M]$, будем называть наименьший прямоугольный параллелепипед, содержащий все эти точки:

$$[M] = \text{Box}\left(\left[\min_{P \in M} P_x, \max_{P \in M} P_x\right], \left[\min_{P \in M} P_y, \max_{P \in M} P_y\right], \left[\min_{P \in M} P_z, \max_{P \in M} P_z\right]\right)$$

Контейнер можно рассматривать для произвольного множества точек. Будем использовать контейнеры для построения



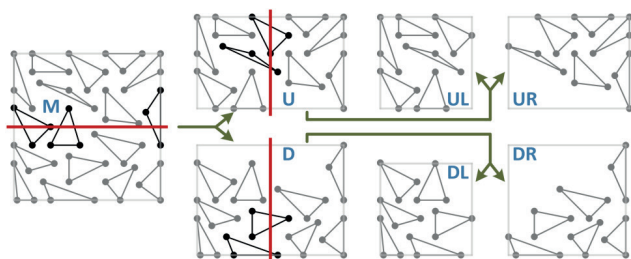
BVN-дерева сетки. Мы будем его использовать для ячейки расчетной сетки, а также для множества ячеек. Так как любой треугольник является выпуклой фигурой, то $[ABC] = [\{A, B, C\}]$, то есть для построения контейнера для треугольника достаточно рассмотреть только его вершины. При поиске пар пересекающихся ячеек будем использовать следующий факт: если два треугольника пересекаются, то пересекаются и их контейнеры: $ABC \cap A'B'C' \neq \emptyset \Rightarrow [ABC] \cap [A'B'C'] \neq \emptyset$. То есть если не пересекаются контейнеры двух треугольников, то не пересекаются и сами треугольники: $[ABC] \cap [A'B'C'] = \emptyset \Rightarrow ABC \cap A'B'C' = \emptyset$ (см. рис. 6). Два треугольника назовем потенциально пересекающимися, если пересекаются их контейнеры.



Р и с. 6. Потенциальное пересечение треугольников в двумерном варианте
F i g. 6. Potential intersection of triangles in two dimensions

В отличие от анализа пары треугольников на пересечение, проверка на пересечение двух прямоугольных параллелепипедов со сторонами, параллельными координатным осям, является простой операцией. То есть на первом этапе будем искать пары потенциально пересекающихся ячеек сетки.

Для поиска потенциально пересекающихся ячеек построим бинарное BVN-дерево. Корнем дерева является контейнер всех ячеек. Рассмотрим процедуру построения дерева, то есть разделения контейнера на два более мелких на примере двумерного случая, проиллюстрированного на рис. 7. Пусть мы хотим разделить контейнер некоторого множества треугольников (M) по горизонтальной прямой на два множества: верхнее (U) и нижнее (D). Тогда в верхнее множество попадут все треугольники, лежащие выше прямой, либо пересекающие ее. В нижнее множество попадут все треугольники, лежащие ниже прямой, либо пересекающие ее. После выделения верхнего и нижнего множества треугольников, для каждого из них строятся свои контейнеры, которые становятся дочерними контейнерами исходного. После этого получившиеся контейнеры можно делить дальше, используя произвольное направление разбиения (X, Y, Z).



Р и с. 7. Схема построения BVN-дерева
F i g. 7. Scheme for constructing a BVN tree

В частности на рис. 7 продемонстрировано разбиение исходного контейнера по схеме $[M] \rightarrow \{[U], [D]\} \rightarrow \{\{[UL], [UR]\}, \{[DL], [DR]\}\}$. За одну операцию контейнер можно разбивать на произвольное количество дочерних контейнеров аналогичным образом. Проводить разбиение контейнеров можно произвольное количество раз, листовые контейнеры в построенном дереве могут содержать произвольное количество треугольников, вплоть до одного треугольника на контейнер. Если при разбиении контейнера по любому направлению в один из дочерних контейнеров попадают все ячейки этого контейнера, то дальнейшее разбиение невозможно.

Введем следующие обозначения. Пусть M и M' – два множества треугольников, $K = [M]$ и $K' = [M']$ – их контейнеры. Обозначим через $P(K, K')$ множество пар потенциально пересекающихся треугольников (T, T') таких, что $T \in M$, $T' \in M'$. Через $C(K)$ обозначим множество дочерних контейнеров для контейнера K . Тогда множество пар потенциально пересекающихся треугольников может быть вычислено рекурсивно следующим образом:

Построенное BVN-дерево позволяет существенно сократить количество проверяемых потенциальных пересечений треугольников [34], так как если $K \cap K' = \emptyset$, L – дочерний

$$P(K, K') = \begin{cases} \bigcup_{\substack{L \in C(K) \\ L' \in C(K')}} P(L, L') & \text{если } C(K) \neq \emptyset, C(K') \neq \emptyset \\ \bigcup_{L \in C(K)} P(L, K') & \text{если } C(K) \neq \emptyset, C(K') = \emptyset \\ P(K', K) & \text{если } C(K) = \emptyset, C(K') \neq \emptyset \\ \left\{ (T, T') : T \in M, T' \in M', \right. \\ \left. T \neq T', [T] \cap [T'] \neq \emptyset \right\} & \text{если } C(K) = \emptyset, C(K') = \emptyset \end{cases}$$

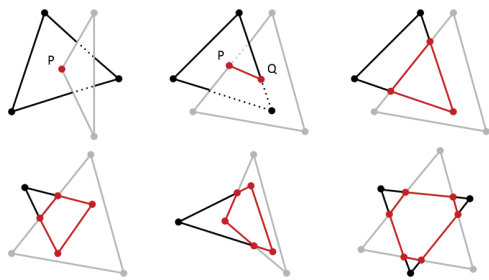
контейнер для K , а L' – дочерний контейнер для K' , то $L \cap L' = \emptyset$. Для поиска всех пар потенциально пересекающихся треугольников из множества всех треугольников расчетной сетки M нужно найти пересечение этого контейнера с самим собой $P([M], [M])$. Сложность нахождения всех пар потенциально пересекающихся треугольников составляет $n \log n$, где n – количество треугольников.

Чтобы определить, действительно пересекаются ли два потенциально пересекающихся треугольника, рассмотрим задачу пересечения треугольника и отрезка в пространстве. Пусть в пространстве задан треугольник $Tri(A, B, C)$ и отрезок $Segm(P, Q)$. Для поиска точек пересечения этого треугольника и отрезка требуется найти решение следующей системы уравнений

$$\begin{cases} x_A + (x_B - x_A)\beta + (x_C - x_A)\gamma = x_P + (x_Q - x_P)\alpha \\ y_A + (y_B - y_A)\beta + (y_C - y_A)\gamma = y_P + (y_Q - y_P)\alpha \\ z_A + (z_B - z_A)\beta + (z_C - z_A)\gamma = z_P + (z_Q - z_P)\alpha \end{cases}$$

при ограничениях $0 \leq \alpha \leq 1$, $\beta \geq 0$, $\gamma \geq 0$, $\beta + \gamma \leq 1$. Эта система уравнений может не иметь решений, может иметь ровно одно решение (если прямая пересекает плоскость треугольника) либо имеет бесконечное количество решений (если прямая лежит в плоскости треугольника).





Р и с. 8. Пересечение двух треугольников в пространстве
F i g. 8. Intersection of two triangles in space

Пусть теперь в пространстве заданы два треугольника. Так как треугольник является выпуклой фигурой, то пересечение двух треугольников либо пусто, либо также является выпуклой фигурой (это может быть выпуклая фигура с количеством вершин от 1 до 6, см. рис. 8). Вершины области пересечения двух треугольников это точки пересечения трех сторон первого треугольника со вторым треугольником и наоборот. То есть для поиска вершин области пересечения двух треугольников в общем случае нужно решить 6 задач пересечения треугольника с отрезком в пространстве.

При нахождении пересечений двух треугольников возможны ошибки определения пересечений из-за точности представления координат вершин с помощью вещественных чисел. Так возможно возникновение ошибок определения пересечений в случае наложения отдельных объектов (вершин, сторон, самих треугольников), нахождения вершины треугольника на стороне или в плоскости другого треугольника и других ситуаций. В работе [23] описан подход, в котором некоторые ошибки, связанные с вычислениями с использованием вещественных чисел, могут быть устранены с помощью обнаружения некорректных пересечений прямой со сторонами треугольника.

	точка	прямая	отрезок	плоскость	треугольник
точка	None, Point	None, Point	None, Point	None, Point	None, Point
прямая	совпадение точек	принадлежность точки прямой	решение системы уравнений	решение системы уравнений	не требуется
отрезок		решение системы уравнений	решение системы уравнений	решение системы уравнений	не требуется
плоскость				не требуется	не требуется
треугольник					None, Point, Segment, Triangle, Quadrangle, Pentagon, Hexagon

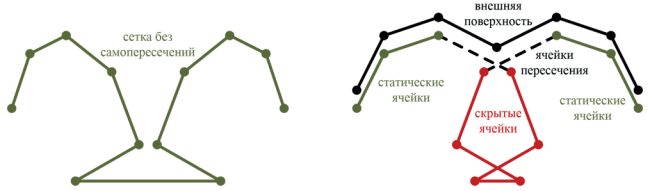
Р и с. 9. Пересечение геометрических примитивов
F i g. 9. Intersection of geometric primitives

Другим подходом, с помощью которого можно полностью избежать проблем точности, связанных с использованием ве-

щественных координат, является представление координат узлов сетки с помощью рациональных чисел. Для решения задачи поиска самопересечений поверхностной неструктурированной сетки используются задачи определения пересечений следующих объектов: точка, прямая, отрезок, плоскость, треугольник (см. рис. 9). Поиск пересечений всех этих объектов на плоскости и в пространстве являются стандартными задачами вычислительной геометрии и решаются с использованием только действий сложения, вычитания, умножения и деления. Таким образом, поиск пересечения всех перечисленных объектов решается в рациональных числах без потери точности. Для использования этого подхода координаты узлов поверхностной сетки необходимо представить в виде рациональных чисел. Основным недостатком использования вычислений в рациональных числах является более медленная обработка по сравнению с использованием вещественных чисел, однако полное избавление от ошибок точности вычислений компенсирует это замедление.

Обход внешней поверхности

Для удаления самопересечений используется стратегия обхода внешней поверхности результирующей сетки с поиском точек и отрезков пересечения обходимых ячеек и их дробления по мере необходимости. На рис. 10 слева показана сетка до возникновения самопересечения (в двумерном случае). На том же рисунке справа показана сетка после возникновения самопересечения и обозначена ее внешняя поверхность. Все ячейки расчетной сетки можно разделить на три категории (см. рис. 10).

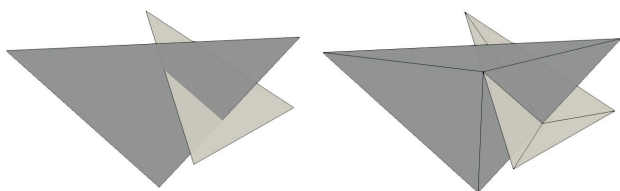


Р и с. 10. Сетка до самопересечения (слева), категории ячеек при самопересечении сетки (справа)

F i g. 10. Mesh before self-intersection (left), cell categories when the mesh self-intersects (right)

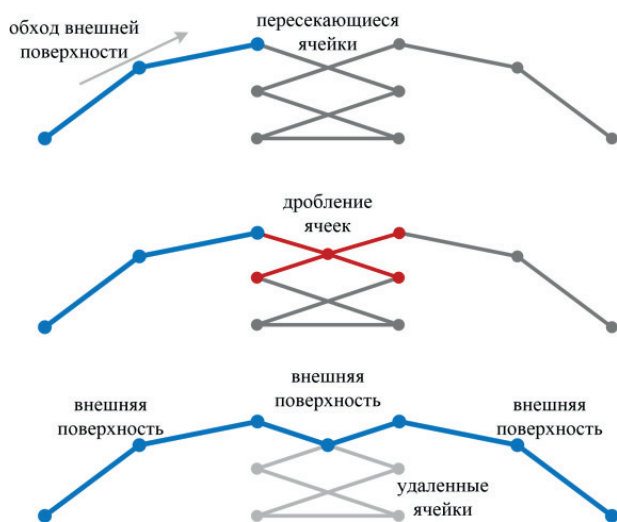
Статическими будем называть ячейки, которые не пересекаются ни с какими другими ячейками, и которые должны стать частью итоговой сетки (будем считать, что в любой момент времени у нас есть возможность указать хотя бы одну статическую ячейку). Скрытыми будем называть ячейки, которые не пересекаются ни с какими другими ячейками, но которые не должны стать частью итоговой сетки (ячейки внутренних петель самопересечения поверхности, которые должны быть удалены). Ячейками пересечения (или конфликтными ячейками) будем называть ячейки, которые пересекаются с какими-то другими ячейками. Как видно из рис. 10 справа, при возникновении самопересечения область скрытых ячеек сама содержит ячейки, пересекающиеся с другими ячейками. Однако, так как вся скрытая область должна быть удалена после обхода внешней поверхности сетки, то такие пересечения вовсе не требуется обрабатывать, что позволяет избежать лишних вычислений.





Р и с. 11. Разбиение треугольника по точкам пересечения
F i g. 1. Splitting of a triangle by intersection points

Рассмотрим подход к удалению самопересечений, при котором ячейки пересечения дробятся на более мелкие по всем точкам и отрезкам пересечения [23]. В качестве примера на рис. 11 слева показаны два треугольника, которые пересекаются по отрезку (две точки пересечения). После выполнения дробления получаем конструкцию, показанную на рис. 11 справа. В общем случае точек, по которым необходимо разбивать ячейку, может быть сколь угодно много, и дробление должно выполняться с помощью выполнения триангуляции по этим точкам, причем некоторые ребра триангуляции должны быть фиксированы, как было отмечено ранее.



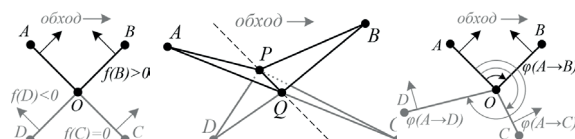
Р и с. 12. Обход внешней поверхности сетки с дроблением. Начало обхода со статической ячейки (сверху), дробление ячеек пересечения (в центре), удаление скрытых ячеек (внизу).

F i g. 12. Traversing the outer surface of a mesh with splitting. Starting from a static cell (top), splitting intersection cells (center), and removing hidden cells (bottom).

Обход внешней поверхности сетки начинается с произвольной статической ячейки (рис. 12 сверху). Далее выполняется обход ячеек сетки, в котором переход между ячейками выполняется через общее ребро. Обход продолжается пока не будет встречена ячейка пересечения. При попадании в ячейку пересечения выполняется поиск всех ее отрезков и точек пересечения, осуществляется ее дробление, а также дробление всех пересекающихся с ней ячеек (рис. 12 в центре). Далее обход продолжается (рис. 12 внизу).

После дробления ячеек по отрезкам и точкам пересечения

между ячейками сетки могут остаться только следующие виды отношений: две ячейки не имеют общих точек, две ячейки имеют одну общую вершину, две ячейки имеют одно общее ребро. При этом в сетке появляются ребра, имеющие более двух инцидентных ячеек. Для простой сетки верно то, что если ребро имеет более двух инцидентных ячеек, то это количество в точности равно 4, причем эти 4 инцидентные ячейки попарно лежат в одной плоскости (первая и третья в порядке обхода вокруг общего ребра лежат в одной плоскости, вторая и четвертая также лежат в одной плоскости).



Р и с. 13. Обход внешней поверхности при устранении самопересечений: простая сетка (слева), схема обхода (в центре), общий случай (справа)

F i g. 13. Traversal of the outer surface with self-intersection elimination: simple mesh (left), traversal diagram (center), general case (right)

Рассмотрим процедуру выбора следующей ячейки обхода сетки для ребер, имеющих количество инцидентных ячеек, больше двух (рис. 13, в центре). На этом рисунке обозначено направление обхода слева направо, вслед за ячейкой, инцидентной вершине A , в результирующую сетку должна войти ячейка, инцидентная вершине B . Рассмотрим эту процедуру для простых сеток, а также для сеток в общем виде. Для простоты будем рассматривать задачу в проекции на плоскость, перпендикулярную ребру.

Рассмотрим случай простой сетки (рис. 13, слева). Пусть уже известно, что ячейка, инцидентная вершине A , помечена, а ее внешняя нормаль равна n_A . Из оставшихся трех ячеек (инцидентных вершинам B, C, D соответственно) необходимо выбрать только одну для продолжения обхода сетки, а остальные удалить. Так как сетка простая, то $\angle AOC = \angle BOD = \pi$, а значит $(n_A, OB) > 0$, $(n_A, OC) = 0$, $(n_A, OD) < 0$. Таким образом, из всех ячеек, необходимо выбрать ту, для которой значение (n_A, OP) максимально.

Теперь рассмотрим сетку в общем виде. В этом случае количество ячеек, инцидентных рассматриваемому ребру, может быть произвольным больше двух, а про значение функции (n_A, OP) сказать ничего нельзя. Для выбора следующей ячейки для обхода сетки будем поворачивать текущую ячейку вокруг рассматриваемого ребра в направлении n_A до совпадения с первой ячейкой (рис. 13, справа). Эта первая ячейка и должна быть выбрана в качестве следующей для продолжения обхода. Если через $\phi(A \rightarrow P)$ обозначить угол поворота исходной ячейки в направлении n_A до ячейки, инцидентной вершине P , то в качестве следующей ячейки для обхода необходимо выбрать ту, для которой $\phi(A \rightarrow P)$ будет минимально.

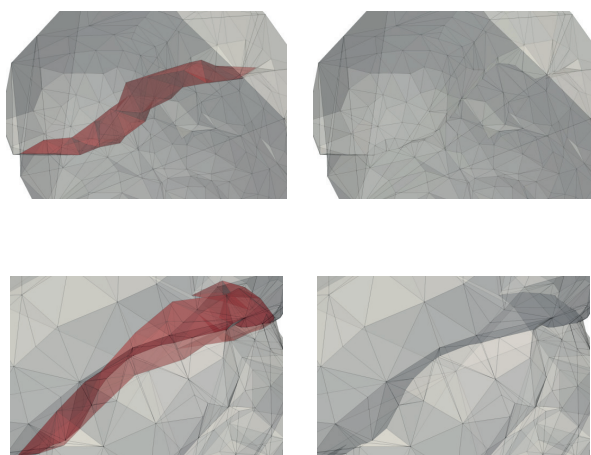
Объединяя вместе оба рассмотренных случая, получаем критерий выбора следующей для обхода ячейки: должна быть выбрана ячейка, для которой значение показателя $f(P)$ максимально, где $f(P) = (n_A, OP)$ для простых сеток и $f(P) = -\phi(A \rightarrow P)$ в общем случае.

После завершения обхода сетки все помеченные ячейки счи-



таются ячейками результирующей поверхности, остальные ячейки удаляются.

Дробление ячейки выполняется только в случае ее обхода. Ячейки, которые находятся под внешней поверхностью и пересекаются с другими ячейками, не обрабатываются, что повышает производительность нахождения самопересечений в случае необходимости удаления обширных внутренних областей.



Р и с. 14. Демонстрация удаления самопересечений
F i g. 14. Demonstration of self-intersections elimination

На рисунке рис. 14 проиллюстрированы примеры устранения самопересечений сетки в виде скрытых петель. После удаления скрытых ячеек результирующая расчетная сетка снова становится корректной и может быть использована для дальнейших вычислений.

Отдельным видом поверхностных неструктурированных сеток, используемых в расчетах, является псевдотрехмерный профиль – незамкнутая расчетная сетка, полученная из двумерного профиля, представляющая собой ленту шириной в одну расчетную ячейку.

Поиск точек пересечения ячеек псевдотрехмерного профиля не может быть выполнен точно даже с использованием рациональных координат. Однако, с псевдотрехмерным профилем удобно работать, если принудительно выровнять его положение по одной из координат, например выровнять его параллельно плоскости OXY , повернув таким образом, чтобы вершины сетки имели координаты $(x, y, 0)$ или (x, y, d) где d – ширина профиля. К такому выровненному профилю может быть применен предложенный метод удаления самопересечений.

Алгоритмы метода удаления самопересечений, основанного на обходе внешней поверхности

Для поиска пар потенциально пересекающихся ячеек используется бинарное BVH-дерево с рекурсивным разделением пополам по одному из направлений, параллельному осям координат OX, OY, OZ [35]. При этом для ограничивающих объемов используется ε -превышение (к максимальному значению

координаты прибавляется ε , от минимального значения координаты отнимается ε) для избегания ошибок, связанных с точностью вычислений. Разделение объема выполняется по тому направлению, которое обеспечивает наиболее равномерное распределение ячеек по дочерним объемам. Деление заканчивается, когда охватывающий объем не может быть разделен ни по одному направлению. Поиск пар потенциально пересекающихся ячеек выполняется глобально для всей сетки перед началом обхода внешней поверхности.

Для определения пересечений геометрических примитивов используются рациональные координаты [36]. Поиск пересечений примитивов точка-отрезок, отрезок-отрезок, точка-треугольник, отрезок-треугольник, треугольник-треугольник выполняется без потери точности, и результатом таких пересечений является точка, отрезок, либо набор отрезков, составляющих выпуклую плоскую фигуру. Геометрические объекты считаются неизменяемыми, для каждого типа примитивов используется кэширование во избежание создания дублирующих объектов. Для линейных элементов сетки создаются несущие геометрические примитивы (для отрезка – содержащая его прямая, для треугольника – плоскость) для быстрого определения лежащих на одной прямой отрезков и лежащих в одной плоскости треугольников.

Для триангуляции ячейки по множеству точек и отрезков используется жадный алгоритм триангуляции с ограничениями [30]. На первом шаге алгоритма в триангуляцию сначала добавляются все отрезки, являющиеся ограничениями (фиксированные отрезки), в случае пересечения они дробятся по всем точкам пересечения. На втором шаге алгоритма создается список всех возможных отсортированных по длине отрезков с концами в точках триангуляции, после чего производится попытка последовательного добавления отрезков в итоговую триангуляцию, начиная с кратчайшего.

Алгоритм обхода внешней поверхности состоит из следующих шагов:

1. Строится BVH-дерево ячеек. Сложность построения $O(n \log n)$, где n – количество ячеек. Переход к шагу 2.
2. Выполняется глобальный поиск всех пар потенциально пересекающихся ячеек. Для каждой ячейки F инициализируется список потенциально пересекающихся с ней ячеек $P(F)$. Сложность $O(n \log n)$. Переход к шагу 3.
3. Начинается обход внешней поверхности в ширину по дуальному графу сетки, начиная с любой статической ячейки. Переход к шагу 4.
4. Если для текущей ячейки F выполняется $P(F) \neq \emptyset$, то переход к шагу 5, иначе выполняется переход к шагу 6.
5. Текущая ячейка помечается. Если на следующую непомеченную ячейку обхода переход выполняется через ребро E такое, что $|F(E)| = 2$, то переходим на следующую ячейку. Если же переход выполняется через ребро E , для которого $|F(E)| > 2$, то выбираем следующую ячейку перехода с помощью функции $f(F)$ и переходим на нее. Переход к шагу 4.
6. Поиск пересечений для ячейки F с каждой ячейкой из $P(F)$ и дробление по точкам и отрезкам пересечения с помощью жадного алгоритма. Операция рекурсивно продолжается для всех ячеек из $P(F)$, для которых пересечение найдено. Текущей остается ячейка F' , имеющая с ячейкой F общее



ребро, через которое был осуществлен переход на ячейку F . Переход к шагу 5.

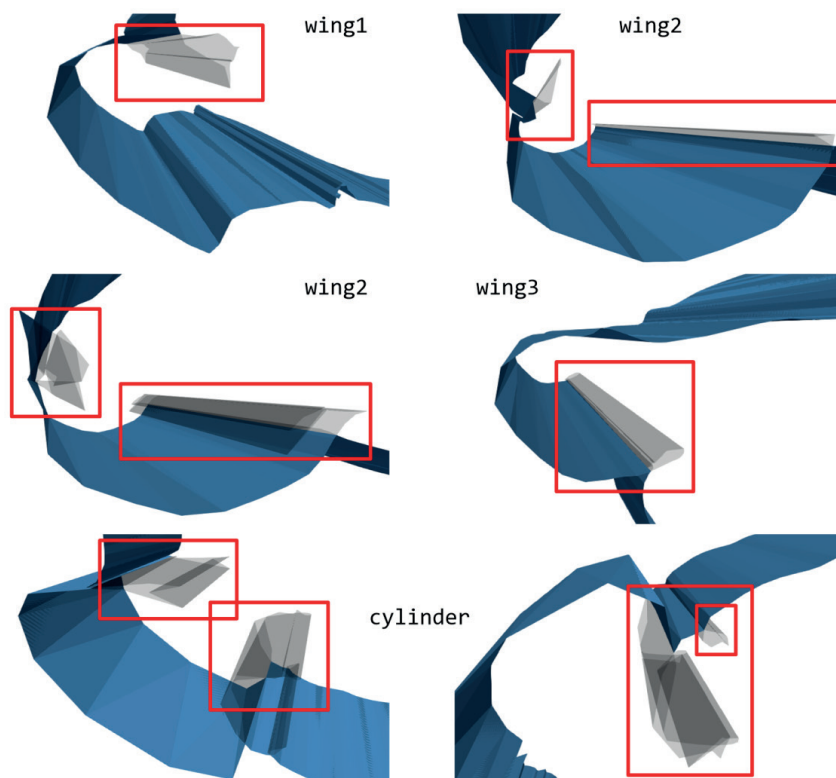
7. Конец обхода. Сложность обхода в среднем $O(n)$. Все помеченные ячейки формируют внешнюю поверхность. Остальные ячейки удаляются.

Метод удаления самопересечений, основанный на обходе внешней поверхности, в применении к псевдотрехмерным профилям реализован в программном комплексе «Кристалл»¹ и апробирован при решении практической задачи по моделированию процесса обледенения.

Таблица 1. Статистика применения алгоритма удаления самопересечений сетки на тестовых примерах

Table 1. Statistics of application of the mesh self-intersection elimination algorithm on test examples

тест	элементы	r_p	r_i	r_p / r_s	новые элементы
wing1	720 / 1440 / 720	4%	2,9%	96% / 62%	102% / 105% / 101%
wing2	720 / 1440 / 720	3,8%	2%	99% / 27%	101% / 104% / 99%
wing2	720 / 1440 / 720	4%	4,4%	96% / 20%	105% / 112% / 96%
wing3	720 / 1440 / 720	3,9%	0,5%	96% / 50%	100% / 101% / 97%
cylinder	720 / 1440 / 720	14%	9,7%	86% / 23%	113% / 129% / 90%
dbl_small	84 / 240 / 160	89%	23,5%	83% / 51%	120% / 175% / 122%
dbl_mid	3844 / 11520 / 7680	2%	3,8%	81% / 50%	106% / 110% / 86%
dbl_big	4996 / 14976 / 9982	6%	9,4%	67% / 50%	119% / 132% / 85%



Р и с. 15. Демонстрация удаления самопересечений тестовых сеток wing1, wing2, wing3, cylinder

Fig. 15. Demonstration of self-intersection removal for test meshes wing1, wing2, wing3, cylinder

¹ Свидетельство о государственной регистрации программы для ЭВМ № 2023666962 Российская Федерация. «Программный модуль компьютерного моделирования процесса обледенения элементов авиационных силовых установок» («Кристалл 2023»): № 2023665836 : заявл. 25.07.2023 : опубл. 08.08.2023 / А. В. Горячев, П. А. Горячев, А. А. Рыбаков ; заявитель Федеральное автономное учреждение «Центральный институт авиационного моторостроения имени П.И. Баранова». EDN: RHUBFI. Свидетельство о государственной регистрации программы для ЭВМ № 2020615575 Российская Федерация. «Программный модуль расчета процесса обледенения элементов авиационных силовых установок» («КРИСТАЛЛ»): № 2020614698 : заявл. 12.05.2020 : опубл. 26.05.2020 / А. В. Горячев, П. А. Горячев, А. А. Рыбаков ; заявитель Федеральное государственное унитарное предприятие «Центральный институт авиационного моторостроения имени П.И. Баранова». EDN: VXFFDG



В таблице 1 приведена статистика применения алгоритма удаления самопересечений сетки с помощью обхода внешней поверхности на тестовых примерах (демонстрация результатов удаления самопересечений приведена на рис. 15. В столбце «тест» приведено название тестовой сетки, в столбце «элементы» – количество узлов, ребер и ячеек сетки (для сетки wing2 было рассмотрено два варианта развития самопересечений). Значение r_p соответствует процентной доле количества пар потенциально конфликтующих ячеек от общего количества пар ячеек. Значение r_i соответствует процентной доле количества ячеек, которые конфликтуют с другими ячейками сетки. Значения r_b и r_s соответствуют процентным долям количества нераздробленных ячеек (больших ячеек) и количества получающихся в результате дробления ячеек (малых ячеек), которые обрабатываются в результате обхода внешней поверхности; чем ниже это значение, тем больше ячеек не требуется обрабатывать по причине нахождения их в скрытой области. В столбце «новые элементы» приведено количество вершин, ребер и ячеек результирующей сетки в процентном выражении относительно количества соответствующих элементов исходной сетки.

Заключение

Разработан метод удаления самопересечений поверхностной неструктурированной сетки, основанный на обходе ее внешней поверхности, которая определяется направлениями внешних нормалей ячеек. При разработке метода учтены основные особенности возникновения самопересечений: самопересечения могут состоять из произвольного набора ломаных (в том числе

пересекающихся) и плоских объектов, сетка может содержать наложение отдельных элементов (узлов, ребер, ячеек), возникновение самопересечений нарушает свойство поверхностной сетки разделения пространства на внутреннюю и внешнюю области. Обход внешней поверхности избавляет от необходимости обрабатывать пересечения ячеек, входящих в скрытую область, что снижает количество обрабатываемых пар пересекающихся ячеек. Во время обхода внешней поверхности все ячейки, которые пересекаются с другими ячейками сетки, дробятся на более мелкие по всем объектам пересечения. Метод использует рациональные координаты для определения пересечения ячеек, а также для разбиения ячеек по точкам и отрезкам пересечения, что избавляет от проблемы точности вычислений. Для поиска пар потенциально пересекающихся ячеек используется классический подход с применением BVH-дерева, что позволяет быстро находить пары потенциально пересекающихся ячеек. Основой метода удаления самопересечений являются алгоритмы определения пересечений геометрических примитивов, алгоритм триангуляции треугольника по множеству точек и отрезков, алгоритм обхода внешней поверхности. Метод в применении к псевдотрехмерным профилям реализован в программном комплексе «Кристалл» и апробирован при решении практической задачи по моделированию процесса ледообразования, в которой поверхностная сетка описывает внешнюю границу ледяного покрова обледеневаящего тела. Метод удаления самопересечений поверхностной неструктурированной сетки путем обхода ее внешней поверхности обеспечивает корректное состояние сетки, описывающей границу расчетной подобласти.

References

- [1] Raj L., Yee K., Myong R. Sensivity of ice and aerodynamic performance degradation to critical physical and modeling parameters affecting airfoil icing. *Aerospace Science and Technology*. 2020;98:105659. <https://doi.org/10.1016/j.ast.2019.105659>
- [2] Martini F., Ibrahim H., Montoya L., Rizk P., Ilinca A. Turbulence modeling of iced wind turbine airfoils. *Energies*. 2022;15:8325. <https://doi.org/10.3390/en15228325>
- [3] Koshelev K.B., Melnikova V.G., Strijhak S.V. Development of iceFoam solver for modeling ice accretion. *Proceedings of ISP RAS*. 2020;32(4):217-234. (In Russ., abstract in Eng.) [https://doi.org/10.15514/ISPRAS-2020-32\(4\)-16](https://doi.org/10.15514/ISPRAS-2020-32(4)-16)
- [4] Galanov N.G., Sarazov A.V., Zhuchkov R.N., Kozelkov A.S. Application of various ice accretion simulation approaches in the LOGOS software package. *Journal of Physics: Conference Series*. 2021;2099:012029. <https://doi.org/10.1088/1742-6596/2099/1/012029>
- [5] Galanov N.G., Kozelkov A.S., Zhuchkov R.N., Sarazov A.V. Testing of an Ice Accretion Simulation Technique Implemented in the Software Package Logos. In: *Informacionny'e sistemy i texnologii IST-2021, sbornik materialov XXVII Mezhdunarodnoj nauchno-texnicheskoj konferencii, 2021*. Nizhny Novgorod: NNSTU, 2021. p. 848-854. (In Russ., abstract in Eng.) EDN: UDSRKB
- [6] Sorokin K.E., et al. Numerical Simulation of Ice Accretion in FlowVision Software. *Computer Research and Modeling*. 2020;2020(1):83-96. (In Russ., abstract in Eng.) <https://doi.org/10.20537/2076-7633-2020-12-1-83-96>
- [7] Shannon T., McClain S. An Assessment of LEWICE Roughness and Convection Enhancement Models. *SAE International Journal of Advances and Current Practices in Mobility*. 2020;2(1):128-139. <https://doi.org/10.4271/2019-01-1977>
- [8] Domingos R., Silva D. 3D computational methodology for bleed air ice protection system parametric analysis. In: *SAE Technical Paper 2015-01-2109*. SAE 2015 International Conference on Icing of Aircraft, Engines, and Structures. Prague, Czech Republic; 2015. 9 p. <https://doi.org/10.4271/2015-01-2109>
- [9] Villedieu P., Trontin P., Chauvin R. Glaciated and mixed phase ice accretion modeling using ONERA 2D icing suite. In: *6th AIAA Atmospheric and Space Environments Conference*, 16-20 June 2014 (AIAA 2014-2199). Atlanta, GA: American Institute of Aeronautics and Astronautics; 2014. <https://doi.org/10.2514/6.2014-2199>
- [10] Son C.-K., Oh S.-J., Yee K. Prediction of glaze ice accretion on 2D airfoil. *Journal of the Korean Society for Aeronautical & Space Sciences*. 2010;38(8):747-757. <https://doi.org/10.5139/JKSAS.2010.38.8.747>
- [11] Bourgault-Cote S., Hasanzadeh K., Lavoie P., Laurendeau E. Multi-layer methodologies for conservative ice growth. In: *7th European Conference for Aeronautics and Aerospace Sciences (EUCASS)*. EUCASS; 2017. 15 p. <https://doi.org/10.13009/EUCASS2017-258>
- [12] Thompson D., Tong X., Arnoldus Q., Collins E., McLaurin D., Luke E. Discrete surface evolution and mesh deformation for aircraft icing applications. In: *Proceedings of the 5th AIAA Atmospheric and Space Environments Conference (AIAA 2013-2544)*. San Diego, CA: American Institute of Aeronautics and Astronautics; 2013. 20 p. <https://doi.org/10.2514/6.2013-2544>



- [13] Tong X., Thompson D., Arnoldus Q., Collins E., Luke E. Three-dimensional surface evolution and mesh deformation for aircraft icing applications. *Journal of Aircraft*. 2017;54:1047-1063. <https://doi.org/10.2514/1.C033949>
- [14] Ye H., Liu Y., Chen B., Liu Z., Zheng J., Pang Y., Chen J. Hybrid grid generation for viscous flow simulation in complex geometries. *Advances in Aerodynamics*. 2020;2:17. 18 p. <https://doi.org/10.1186/s42774-020-00042-x>
- [15] Rybakov A.A. Mesh geometric rebuilding using the common envelope of a spheres family in the ice accretion problem. *Modern Information Technologies and IT-Education*. 2023;19(2):282-291. (In Russ., abstract in Eng.) <https://doi.org/10.25559/SITITO.019.202302.282-291>
- [16] Freylekhman S., Rybakov A. Self-intersection elimination for unstructured surface computational meshes. *Lobachevskii Journal of Mathematics*. 2022;43(10):2846-2852. <https://doi.org/10.1134/S1995080222130133>
- [17] Xiao Z., Chen J., Zheng Y., Zheng J., Wang D. Booleans of triangulated solids by a boundary conforming tetrahedral mesh generation approach. *Computers & Graphics*. 2016;59:13-27. <https://doi.org/10.1016/j.cag.2016.04.004>
- [18] Hu Y., Zhou Q., Gao X., Jacobson A., Zorin D., Panozzo D. Tetrahedral meshing in the wild. *ACM Transactions on Graphics*. 2018;37(4):60. <https://doi.org/10.1145/3197517.3201353>
- [19] Hu Y., Schneider T., Wang B., Zorin D., Panozzo D. Fast tetrahedral meshing in the wild. *ACM Transactions on Graphics*. 2020;39(4):117. <https://doi.org/10.1145/3386569.3392385>
- [20] Lo. S.H., Wang W.X. A fast robust algorithm for the intersection of triangulated surfaces. *Engineering with Computers*. 2004;20:11-21. <https://doi.org/10.1007/s00366-004-0277-3>
- [21] Gorshkov A.V., Korshunova A.L. Optimal intersection search algorithm in Monte Carlo simulation of light transport in human brain. *Vestnik Nizhegorodskogo universiteta im. N. I. Lobachevskogo* = Vestnik of Lobachevsky University of Nizhni Novgorod. 2012;(5):73-80. (In Russ., abstract in Eng.) EDN: PZKPSV
- [22] Magalhaes S., Franklin W., Andrade M. Fast exact parallel 3D mesh intersection algorithm using only orientation predicates. In: Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL '17). New York, NY, USA: Association for Computing Machinery; 2017. Article number: 44. <https://doi.org/10.1145/3139958.3140001>
- [23] Skorkovská V., Kolingerová I., Benes B. A Simple and Robust Approach to Computation of Meshes Intersection. In: Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2018) – Volume 1: GRAPP. SCITEPRESS – Science and Technology Publications, Lda.; 2018. <https://doi.org/10.5220/0006538401750182>
- [24] Liu T., Ye H., Meng X., Liu Z., Chen J. Robust and fast local repair for intersecting triangle meshes. *Computer-Aided Design*. 2026;190:103963. <https://doi.org/10.1016/j.cad.2025.103963>
- [25] Guo J.-P., Fu X.-M. Exact and efficient intersection resolution for mesh arrangements. *ACM Transactions on Graphics*. 2024;43(6):165. <https://doi.org/10.1145/3687925>
- [26] Cherchi G., Livesu M., Scateni R., Attene M. Fast and robust mesh arrangements using floating-point arithmetic. *ACM Transactions on Graphics*. 2020;39(6):250. <https://doi.org/10.1145/3414685.3417818>
- [27] Rivara M.C., Rodriguez-Moreno P.A. Tuned Terminal Triangles Centroid Delaunay Algorithm for Quality Triangulation. In: Roca X., Loseille A. (eds.) 27th International Meshing Roundtable. IMR 2018. *Lecture Notes in Computational Science and Engineering*. Vol. 127. Cham: Springer; 2019. p. 211-228. https://doi.org/10.1007/978-3-030-13992-6_12
- [28] Rakotoarivelo H., Ledoux F. Accurate Manycore-Accelerated Manifold Surface Remesh Kernels. In: Roca X., Loseille A. (eds.) 27th International Meshing Roundtable. IMR 2018. *Lecture Notes in Computational Science and Engineering*. Vol. 127. Cham: Springer; 2019. p. 405-423. https://doi.org/10.1007/978-3-030-13992-6_22
- [29] Borouchaki H., Laug P., George P.-L. Parametric surface meshing using a combined advancing-front generalized Delaunay approach. *International Journal of Numeric Methods in Engineering*. 2000;49:233-259. [https://doi.org/10.1002/1097-0207\(20000910/20\)49:1/23.0.CO;2-G](https://doi.org/10.1002/1097-0207(20000910/20)49:1/23.0.CO;2-G)
- [30] Skvortsov A.V. *Triangulyaciya Delone i ee primeneniye* [Delaunay triangulation and its applications]. Tomsk: Izdatel'stvo Tomskogo universiteta; 2002. 128 p. (In Russ., abstract in Eng.) EDN: RZIMNT
- [31] Skvortsov A.V. Algorithms Implementation Features of Delaunay Triangulation with Restrictions. *Tomsk State University Journal*. 2002;(275):90-94. (In Russ., abstract in Eng.) EDN: OYCDAY
- [32] Panchal D., Jayaswal D. Feature sensitive geometrically faithful highly regular direct triangular isotropic surface remeshing. *Sadhana*. 2022;47(94):1-19. <https://doi.org/10.1007/s12046-022-01866-7>
- [33] Budak A.S., Frolov V.A., Galaktionov V.A., Garifullin A.R. Study of surface representation methods based on signed distance functions. *Programming and Computer Software*. 2025;3:15-26. <https://doi.org/10.7868/S3034584725030027>
- [34] Jung W., Shin H., Choi B. Self-intersection removal in triangular mesh offsettings. *CAD Journal*. 2004;(1):477-484. <https://doi.org/10.1080/16864360.2004.10738290>
- [35] Glumova E.S. Algorithm Adaptive Point Cloud Partitioning based on Bounding Volume Hierarchy. In: Graphic information systems and immersive technologies in the product life cycle stages (GraphiCon 2025). Yoshkar-Ola: Volga State University of Technology; 2025. p. 967-984. (In Russ., abstract in Eng.) <https://doi.org/10.25686/978-5-8158-2474-4-2025-976-984>
- [36] Lazard S., Valque L. Removing self-intersections in 3D meshes while preserving floating-point coordinates. hal-04907149. 2025. Available at: <https://inria.hal.science/hal-04907149v1> (accessed 15.02.2026).



*Поступила 15.02.2026; одобрена после рецензирования 29.03.2026; принята к публикации 02.04.2026.
Submitted 15.02.2026; approved after reviewing 29.03.2026; accepted for publication 02.04.2026.*

Об авторе:

Рыбаков Алексей Анатольевич, ведущий научный сотрудник, ФГБУ «Национальный исследовательский центр «Курчатовский институт»» (123182, Российская Федерация, г. Москва, пл. Академика Курчатова, д. 1), кандидат физико-математических наук, ORCID: <https://orcid.org/0000-0002-9755-8830>, rybakov_aan@nrcki.ru

Автор прочитал и одобрил окончательный вариант рукописи.

About the author:

Alexey A. Rybakov, Lead Researcher, National Research center "Kurchatov Institute for Theoretical and Experimental physics" (1 Akademicheskaya Kurchatova Sq., Moscow 123182, Russian Federation), Cand. Sci. (Phys.-Math.), ORCID: <https://orcid.org/0000-0002-9755-8830>, rybakov_aan@nrcki.ru

The author has read and approved the final manuscript.

