



УДК 004.021

DOI: 10.255559/SITITO.14.201801.213-221

## ПРОГРАММНЫЙ МОДУЛЬ ДЛЯ ПОСТРОЕНИЯ ПЕРЕСЕЧЕНИЯ ТРИАНГУЛИРОВАННЫХ ПОВЕРХНОСТЕЙ

**В.В. Курганский, Э.В. Гайдук**

Ярославский государственный университет им. П.Г. Демидова, г. Ярославль, Россия

### Аннотация

Предлагается эффективный алгоритм реализации булевых операций над триангулированными поверхностями, а именно: дизъюнкции, конъюнкции и булевой разности, и его программная реализация. Идея алгоритма состоит в следующем. Сначала выясняются пары пересекающихся треугольников: локализуется место пересечения двух поверхностей с помощью ограничивающих объём параллелепипедов и дальнейшего их пересечения. Затем строится линия пересечения для каждой пары треугольников: выбирается пара пересекающихся треугольников, строится отрезок, по которому они пересекаются. Далее, благодаря введённой структуре данных, выбираются «соседние» треугольники, среди которых выбираются те, которые образуют пересекающуюся пару. После этого описанный процесс продолжается. После этого треугольники, участвующие в пересечении, ретриангулируются. Для каждого треугольника известны все ребра, по которым он пересекается с треугольниками из другой поверхности. Эти ребра являются структурными ребрами в задаче триангуляции с ограничениями для данного треугольника. Затем поверхности объединяются в одну и формируются циклы пересечений. Далее по циклам пересечения выделяются подповерхности, ограниченные найденными циклами. Поскольку линия пересечения поверхностей строилась последовательно, возможно задать направление каждого ребра. Выбирается любое ребро из линии пересечения. В строящуюся подповерхность добавляется треугольник, который включает в себя это ребро и его ориентация совпадает с направлением ребра. Из линии пересечения удаляется выбранное на предыдущем шаге ребро, но добавляются два новых – оставшиеся ребра добавленного треугольника, их ориентация задается таким образом, чтобы не нарушалась возможность пройти по линии пересечения по циклу. Этот процесс повторяется до тех пор, пока линия пересечения не пуста. Процесс построения подповерхности необходимо проделать для каждой линии пересечения. После этого все треугольники, не вошедшие ни в одну из построенных подповерхностей, объединяются и образуют новую подповерхность. Затем из полученных подповерхностей в зависимости от выполняемой булевой операции собираются итоговые поверхности. Рассмотренный алгоритм реализован на языке Java 7 и успешно внедрён в специализированные программные продукты: «3D-SchoolEdit» и «3D-ChemistryEdit». Первый из них представляет собой программу для построения 3D-моделей

### Об авторах:

**Курганский Владимир Витальевич**, студент, математический факультет, Ярославский государственный университет им. П.Г. Демидова (150003, Россия, г. Ярославль, ул. Советская, д. 14); ORCID: <https://orcid.org/0000-0002-3648-3396>, [Vladimir.kurgansky@gmail.com](mailto:Vladimir.kurgansky@gmail.com)

**Гайдук Эльвира Васильевна**, магистрант, математический факультет, Ярославский государственный университет им. П.Г. Демидова (150003, Россия, г. Ярославль, ул. Советская, д. 14); ORCID: <https://orcid.org/0000-0001-7428-8375>, [elja5@yandex.ru](mailto:elja5@yandex.ru)

© Курганский В.В., Гайдук Э.В., 2018



для школьных задач по стереометрии с возможностью 3D-печати данных моделей, второй – редактор химических соединений с возможностью 3D-печати.

**Ключевые слова**

Образование; ИТ; программный продукт; 3D-технологии; алгоритм.

**SOFTWARE MODULE FOR CONSTRUCTING THE INTERSECTION OF TRIANGULATED SURFACES**

**Vladimir V. Kurgansky, Elvira V. Gayduk**

P.G. Demidov Yaroslavl State University, Yaroslavl, Russia

**Abstract**

*The effective algorithm is proposed for implementing Boolean operations over triangulated surfaces, namely, disjunction, conjunction and Boolean difference, and its software implementation. The idea consists in as follow.*

*The first step is to determine pairs of intersecting triangles: localizing the intersection of the two surfaces using the bounding volume of the parallelepipeds and the future of their intersection.*

*The second step is constructing an intersection line for each pair of triangles: a pair of intersecting triangles is selected, and the segment along which they intersect is constructed. Further, thanks to the entered data structure, "adjacent" triangles are selected, among which are selected those that form the intersecting pair. The process described above continues as long as such triangles can be detected. After that the triangles involved in the intersection are retriangulated. For each triangle, all the edges are known on which it intersects with triangles from another surface. These edges are structural edges in the triangulation problem with constraints for a given triangle.*

*The third step is to combine all surfaces into one surface. Further, subsurfaces are constructed along the loops of intersection limited by the found loops. Since the intersection line of the surfaces was constructed in sequence, it is possible to specify the direction of each edge. Any edge from the intersection line is selected. The triangle is added to the subsurface under construction, which includes this edge and its orientation is the same as the direction of the edge. The edge which was selected previously is deleted from intersection line, but two new edges are added is the remaining edges of added triangle.*

*The third step is to combine all surfaces into one surface. Further, subsurfaces are constructed along the cycles of intersection limited by the found cycles. Since the intersection line of the surfaces was constructed in sequence, it is possible to specify the direction of each edge. Any edge from the intersection line is selected. The triangle is added to the subsurface under construction, which includes this edge and its orientation is the same as the direction of the edge. The edge which was selected previously is deleted from intersection line, but two new edges are added is the remaining edges of added triangle. Their orientation is set in such a way that the opportunity to pass along the intersection line along the cycle is not broken. This process is repeated as long as the intersection line is not empty. The process of constructing subsurface must be done for each intersection line. After that, all triangles that have not entered any of the constructed surfaces are combined and form a new subsurface. The final surfaces are collected from obtained subsurfaces depending on the Boolean operation being performed.*

*The algorithm is implemented in Java 7 and successfully integrated in specialized software products: "3D-School-Edit" and "3D-Chemistry-Edit". The first program is the software for creating 3D-models for school tasks on stereometry with the possibility 3D-printing of these models. The second one is the editor of chemical compounds with the possibility of 3D printing.*



## Keywords

*Education; IT; software; 3D-technologies; algorithm.*

## Введение

В задачах, связанных с обработкой трехмерных поверхностей, важную роль играет вычисление результата применения булевых операций. Часто практическая задача обработки данных сводится к поиску объединения, пересечения, либо булевой разности двух поверхностей.

Рассмотрим конкретный пример такой практической задачи, возникшей при создании специализированных программных продуктов «3D-SchoolEdit» и «3D-ChemistryEdit» (в команду разработчиков которых входят авторы данной статьи).

Первая из этих программ «3D-SchoolEdit» – это редактор школьных планиметрических и стереометрических чертежей, который позволяет автоматически строить по данным чертежам 3D-модели для вывода на 3D-печать. Например, каркасные модели для 3D-печати строятся следующим образом. Вокруг каждой вершины геометрического объекта (например, тетраэдра) строится сфера, а вокруг каждого ребра – цилиндр. Таким образом, модель для 3D-печати – это поверхность, ограничивающая объединение стереометрических объектов (шаров и цилиндров).

Программа «3D-ChemistryEdit» – это редактор химических соединений, предназначенный для школьных уроков химии.

Как это общепринято, в редакторе для трехмерной визуализации атомов используются шары разного диаметра, а для обозначения связей между ними – цилиндры, расположенные под определенными углами. Таким образом, визуализируемая поверхность, соответствующая молекуле, – это снова объединение цилиндров и шаров.

Важной особенностью разрабатываемых программных продуктов является возможность экспорта отображаемых чертежей и химических соединений в формат пригодный для 3D-печати. При этом оказывается актуальной проблема уменьшения экспортируемых данных, поскольку модели зачастую имеют высокую детализацию и, как следствие, занимают большой объем памяти. В частности, добиться

уменьшения объема данных можно за счет удаления внутренних частей объединения поверхностей – шаров и цилиндров, из которых состоят модели. Таким образом, возникает необходимость построения результата применения булевых операций к трехмерным поверхностям. Причем в рамках данной статьи в качестве поверхностей рассматриваются не только цилиндры и шары, необходимые для «сборки» моделей молекул и каркасных моделей геометрических чертежей. Задача исследуется в более общей постановке: для выпуклых тел.

Поскольку триангуляция является наиболее популярным и, во многих отношениях, удобным способом представления трехмерных поверхностей, в рамках данной работы остановимся на рассмотрении указанных булевых операций именно над триангулированными трехмерными поверхностями.

В настоящей работе предлагается эффективный алгоритм реализации объединения, пересечения, булевой разности триангулированных поверхностей. Данный алгоритм основан на идеях, предложенных в работах, который состоит в следующем [1, 6 - 25]. Сначала выясняются пары пересекающихся треугольников, затем строится линия их пересечения для каждой пары. После, треугольники, участвующие в пересечении, ретриангулируются. Затем поверхности объединяются в одну и формируются циклы пересечений. Далее по циклам пересечения выделяются подповерхности, ограниченные найденными циклами. Затем из полученных подповерхностей в зависимости от выполняемой булевой операции собираются итоговые поверхности.

Алгоритм дополнен и улучшен за счёт введения структуры данных, которая позволяет ускорить построение линии пересечения поверхностей. Также использован подход, который позволяет ускорить поиск первой пары пересекающихся треугольников за счет локализации места пересечения поверхностей. Всё это позволяет уменьшить время работы конкретной реализации алгоритма.



В работе используется термин *полигональная сетка*, под которым будем понимать совокупность вершин, ребер и граней, определяющие форму многогранного объекта в трехмерной компьютерной графике и объемном моделировании.

Как было указано выше, в рамках настоящей работы в качестве полигональной сетки будет рассматриваться сетка с гранями-треугольниками, то есть *триангулированная поверхность*.

### Постановка задачи

Рассматривается следующая задача. На вход подаются две триангулированные поверхности в формате STL, ограничивающие выпуклые тела  $P_1$  и  $P_2$ . Требуется построить поверхность, ограничивающую объем пересечения  $I = P_1 \cap P_2$ , объединения  $U = P_1 \cup P_2$  и булевой разности  $I = P_1 - P_2$  исходных тел. Затем экспортировать файл 3D-модели в формат STL.

На входные триангулированные поверхности накладываются следующие ограничения:

- поверхности ограничивают выпуклые многогранники;
- циклы пересечения данных поверхностей не имеют самопересечений;
- циклы пересечения поверхностей не имеют пересечений или касаний друг с другом.

Эти требования обусловлены конкретным реализованным подходом к решению задачи.

### Описание алгоритма

Основные шаги алгоритма следующие:

Поиск пары пересекающихся треугольников исходных поверхностей;

Построение линии пересечения поверхностей и ре-триангуляция мест пересечения;

Разделение поверхностей на части по линии пересечения;

Сборка полученных частей поверхностей.

Далее на каждом шаге алгоритма остановимся более подробно.

### Поиск пары пересекающихся треугольников исходных поверхностей

Предлагается использовать некоторый известный подход к разбиению пространства, что необходимо для локализации пересечения поверхностей: принцип бинарного дерева ограничивающих объемов Bounding Volume

Hierarchy (BVH) [2]. Идея состоит в том, чтобы геометрически простым, ограничивающим объем, объектом приблизить место пересечения двух полигональных сеток, упростив этим поиск пары пересекающихся треугольников. Предлагается приближать место пересечения, разбивая исходное пространство, в которое целиком входят полигональные сетки, на два геометрически простых объекта на каждом шаге. Затем каждый из этих объектов разбивается еще на два объекта и так далее, пока место пересечения не будет содержаться целиком в одном объекте. В качестве ограничивающего объема используется ориентированный осям прямоугольный параллелепипед Axis-Aligned Bounding Box (AABB) [3]. С помощью AABB достаточно просто искать пересечения.

Для каждой полигональной сетки, участвующей в операции, строится, ограничивающий её объем, ориентированный осям прямоугольный параллелепипед.

После этого полученные AABB пересекаются, их пересечением также является параллелепипед (рис. 1).

Так как каждый треугольник сетки гарантированно находится внутри соответствующего AABB, можем утверждать следующее: каждый, участвующий в пересечении с другой сеткой треугольник, полностью или частично находится внутри параллелепипеда, полученного пересечением исходных AABB.

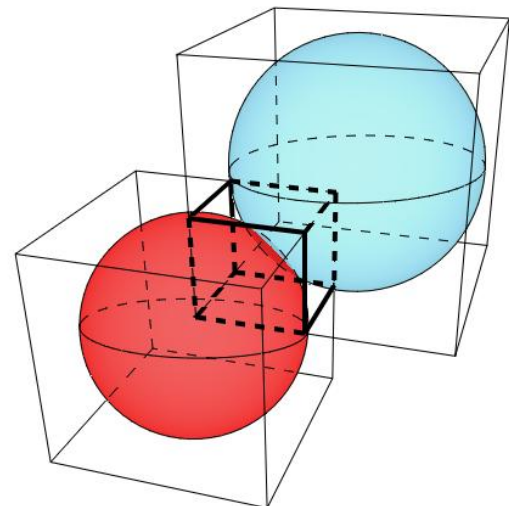


Рисунок 1. Пересечение двух AABB поверхностей выпуклых тел



Таким образом, в каждой сетке выделяются только те треугольники, которые потенциально могут участвовать в пересечении, то есть задевают или полностью оказываются в пересечении соответствующих параллелепипедов.

### Построение линии пересечения поверхностей и ре-триангуляция мест пересечения

Далее находится первая пара пересекающихся треугольников.

После отыскания первой пересекающейся пары треугольников, происходит процесс построения линии пересечения поверхностей.

В работе используется структура данных «Winged edge», в которой для каждого ребра сохраняются списки смежных рёбер, инцидентных граней и вершин [4]. Указанная структура позволяет осуществлять поиск линии пересечения только среди «соседей» (смежных по ребру треугольников). Благодаря такому подходу, поиск и построение «соседних» пересечений существенно ускоряется.

После пересечения треугольников необходимо провести их ре-триангуляцию (см. рис. 2). Причем в данном случае необходимо построение так называемой триангуляции с ограничениями.

В общем случае в задаче триангуляции с ограничениями задаётся исходное множество точек – узлов триангуляции и множество отрезков-ограничителей, гарантированно присутствующих в триангуляции [5]. В рассматриваемой ситуации, ребрами-ограничителями служат рёбра, по которым пересекаются треугольники из разных поверхностей.

Эта операция необходима для того, чтобы в дальнейшем корректно разделить треугольники на части в зависимости от необходимой булевой операции, не нарушив структуру сетки.

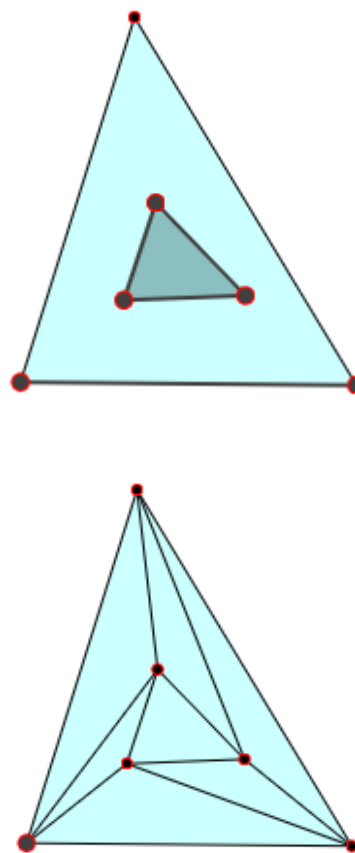


Рисунок 2. Пример ре-триангуляции треугольника, содержащего ребра-ограничения

Для возможности корректного разделения, применяется цепной алгоритм ре-триангуляции с ограничениями в местах пересечения поверхностей [5].

Таким образом, последовательно выполняя пересечения треугольников и добавляя ребра, строим один из циклов пересечения поверхностей.

### Разделение поверхностей на части по линии пересечения

Далее происходит «разрезание» имеющихся поверхностей на подповерхности по линиям пересечения.

Поскольку каждый треугольник ориентирован, выбираются те, у которых ориентация совпадает с текущим ребром линии пересечения, ориентированным по направлению обхода. Таким образом, на каждом шаге захватывается треугольник, а из линии пересечения удаляется ребро, которое было





сонаправлено одному из ребёр треугольника. В свою очередь, оставшиеся два ребра треугольника вставляются в линию пересечения на место удалённого. Процесс повторяется до тех пор, пока линия пересечения имеет рёбра (рис. 3).

#### Сборка полученных частей поверхностей

В результате предыдущих трёх шагов алгоритма, имеется несколько частей исходных поверхностей, которые остаётся соединить по тому или иному принципу, в зависимости от выполняемой булевой операции: объединение, пересечение или булева разность.

Поскольку выделенные части в местах пересечений были ре-триангулированы и корректно разделены, «склеить» их не составляет труда, так как рёбра из цикла, по которому были разделены полигональные сетки, входят во все разделённые части и являются в них краем (рис. 4).

После всех шагов алгоритма результат булевой операции экспортируется в файл, в соответствии с выбранным пользователем форматом для хранения 3D-моделей.

#### Заключение

В данной работе был рассмотрен алгоритм выполнения булевых операций над триангулированными поверхностями. Алгоритм был реализован на языке Java 7 и успешно внедрён, как часть модуля для работы с триангулированными поверхностями, в редактор химических соединений с возможностью 3D-печати «3D-ChemistryEdit» и редактор геометрических чертежей «3D-SchoolEdit». Итогом внедрения данного алгоритма является уменьшение размера экспортируемых 3D-моделей и, как следствие, ускорение работы редакторов.

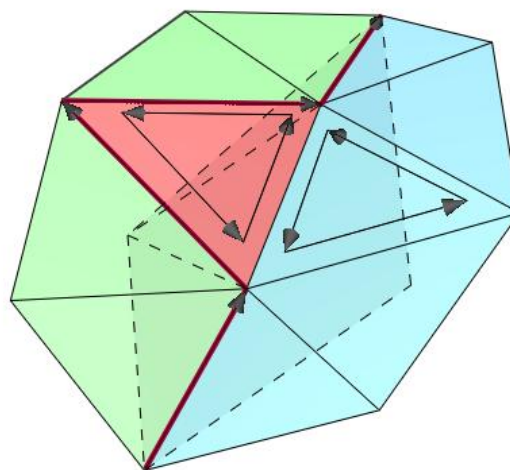
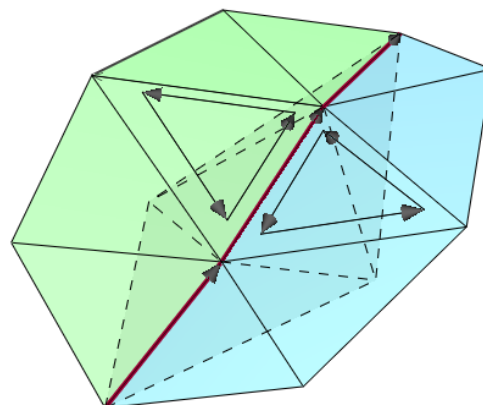


Рисунок 3. Выделение части поверхности

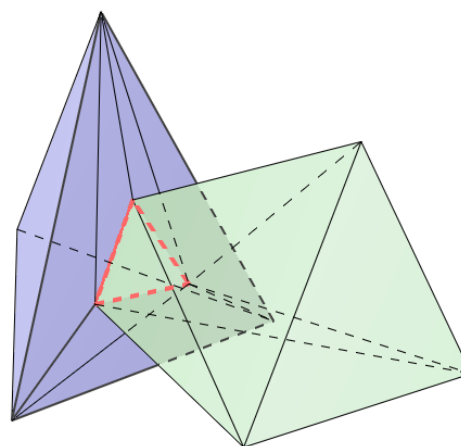


Рисунок 4. Пример сборки получившихся частей поверхностей

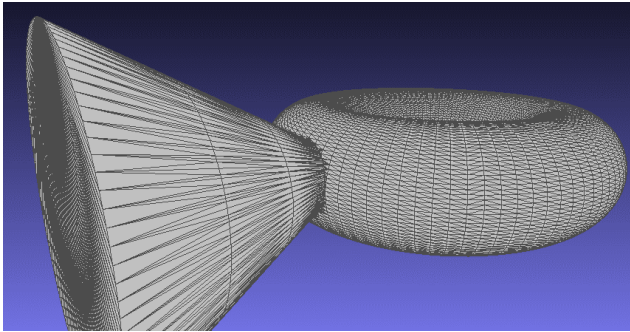


Рисунок 5. Пересечение конуса и тора

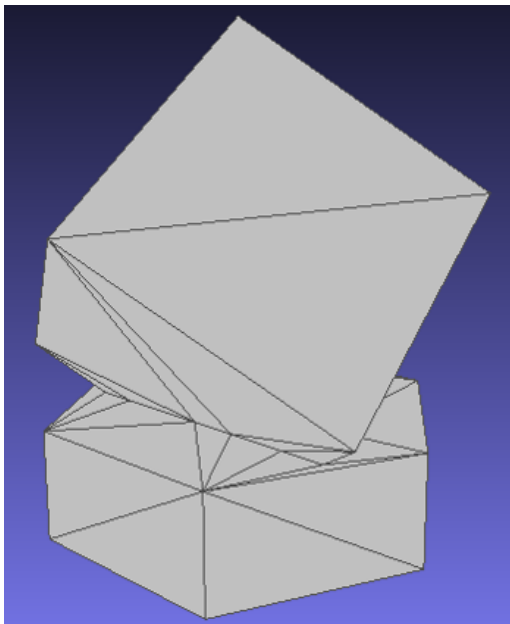


Рисунок 6. Пересечение куба и гексагона

Планируется модификация алгоритма, которая снизит требования к входным данным. А именно, снимет ограничение выпуклости и запрет пересечения циклов друг с другом.

### Примеры работы алгоритма

В качестве иллюстрации работы алгоритма приведем примеры, изображенные на рисунках 5 - 8.

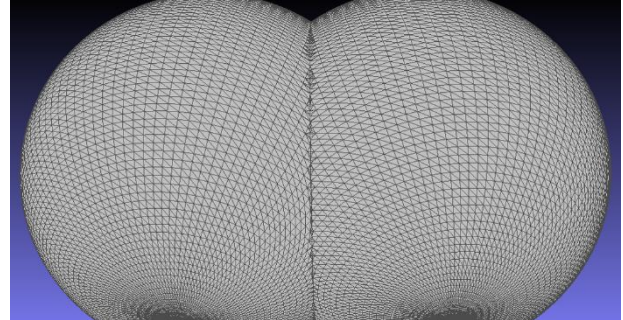


Рисунок 7. Пересечение двух сфер

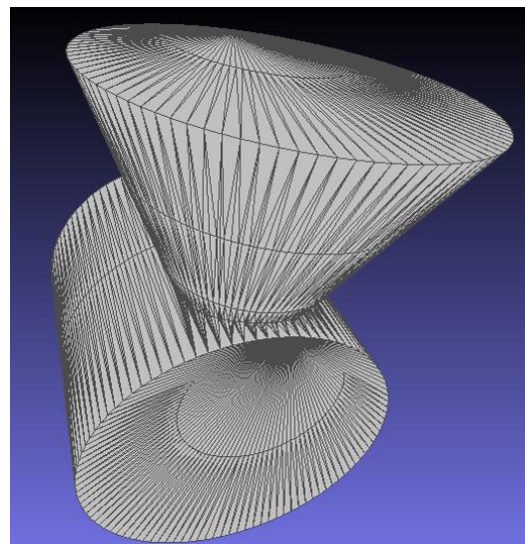


Рисунок 8. Пересечение конуса и цилиндра

### СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Mei G, Tipper J.C. Simple and robust boolean operations for triangulated surface // Computing Research Repository. 2013. Vol. abs/1308.4434. Pp. 1-18. URL: <https://arxiv.org/pdf/1308.4434.pdf> (дата обращения: 10.02.2018).
- [2] Early split clipping for bounding volume hierarchies / M. Ernst, G. Greiner // Proceedings of the IEEE Symposium on Interactive Ray Tracing. 2007. Pp.73 - 78. DOI: <http://doi.org/10.1109/RT.2007.4342593>
- [3] Bergen G.V.D. Efficient Collision Detection of Complex Deformable Models using AABB Trees // Journal of Graphics Tools. 1997. Vol. 2, issue 4. Pp. 1-13. DOI: <https://doi.org/10.1080/10867651.1997.10487480>
- [4] Baumgart B.G. Winged Edge Polyhedron Representation. Technical Report, Stanford University, Stanford, CA, USA. 1972.
- [5] Скворцов А.В. Триангуляция Делоне и её применение. Томск: Изд-во Том. ун-та. 2002. 128 с.
- [6] Sâm Landier. Boolean operations on arbitrary polygonal and polyhedral meshes // Computer-Aided Design. 2017. Vol. 85. Pp. 138-153. DOI: <https://doi.org/10.1016/j.cad.2016.07.013>
- [7] Kugler M., Hostettler A., Soler L., Rémond Y., George D. A new algorithm for volume mesh refinement on merging geometries: Application to liver and vascularization // Journal of Computational and Applied Mathematics. 2018. Vol. 330, Issue C, March 2018. Pp. 429-440. DOI: <https://doi.org/10.1016/j.cam.2017.09.012>



- [8] Bagley B., Sastry Sh.P., Whitaker R.T. A Marching-tetrahedra Algorithm for Feature-preserving Meshing of Piecewise-smooth Implicit Surfaces // *Procedia Engineering*. 2016. Vol. 163. Pp. 162-174. DOI: <https://doi.org/10.1016/j.proeng.2016.11.042>
- [9] Xu S., Keyser J. Fast and robust Booleans on polyhedral // *Computer-Aided Design*. 2013. Vol. 45, issue 2. Pp. 529-534. DOI: <https://doi.org/10.1016/j.cad.2012.10.036>
- [10] Attene M. Direct repair of self-intersecting meshes // *Graphical Models*. 2014. Vol. 76, issue 6. Pp. 658-668. DOI: <https://doi.org/10.1016/j.gmod.2014.09.002>
- [11] Zhoufang Xiao, Jianjun Chen, Yao Zheng, Jianjing Zheng, Desheng Wang. Booleans of triangulated solids by a boundary conforming tetrahedral mesh generation approach // *Computers & Graphics*. 2016. Vol. 59, issue C (October 2016). Pp. 13-27. DOI: <http://dx.doi.org/10.1016/j.cag.2016.04.004>
- [12] A practical and robust method to compute the boundary of three-dimensional axis-aligned boxes / D.L. Monterde, L. Martinez, M. Vigo, N. Pla // *GRAPP 2014 - Proceedings of the 9th International Conference on Computer Graphics Theory and Applications*. 5-8 Jan. 2014. Pp. 34-42. URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&number=7296027&isnumber=7295996> (дата обращения: 10.02.18).
- [13] Guo K.B., Zhang L.C., Wang C.J. et al. Boolean operations of STL models based on loop detection // *International Journal of Advanced Manufacturing Technology*. 2007. Vol. 33, issue 5-6. Pp. 627-633. DOI: <https://doi.org/10.1007/s00170-006-0487-5>
- [14] Wang Ch. Approximate Boolean Operations on Large Polyhedral Solids with Partial Mesh Reconstruction // *IEEE transactions on visualization and computer graphics*. 2011. Vol. 17, issue 6. Pp. 836-849. DOI: <https://doi.org/10.1109/TVCG.2010.106>
- [15] Jiang X., Peng Q., Cheng X., Dai N., Cheng Ch., Li D. Efficient Booleans algorithms for triangulated meshes of geometric modeling // *Computer-Aided Design and Applications*. 2016. Vol. 13, no. 4. Pp. 419-430. DOI: <https://doi.org/10.1080/16864360.2015.1131530>
- [16] Chen, Ming & Yu Chen, Xiao & Tang, Kai & Yuen, Matthew. Efficient Boolean Operation on Manifold Mesh Surfaces // *Computer-Aided Design and Applications*. 2013. Vol. 7. Pp. 405-415. DOI: <https://doi.org/10.3722/cadaps.2010.405-415>
- [17] Lo S.H., Wang W.X. A fast robust algorithm for the intersection of triangulated surfaces // *Engineering with Computers*. 2004. Vol. 20, issue 1. Pp. 11-21. DOI: <https://doi.org/10.1007/s00366-004-0277-3>
- [18] Campen M., Kobbelt L. Exact and Robust (Self-)Intersections for Polygonal Meshes // *Computer Graphics Forum*. 2010. Vol. 29. Pp. 397-406. DOI: <https://doi.org/10.1111/j.1467-8659.2009.01609.x>
- [19] Zaharescu A., Boyer E., Horaud R. Topology-Adaptive Mesh Deformation for Surface Evolution, Morphing, and Multiview Reconstruction // *IEEE Transactions on pattern analysis and machine intelligence*. 2011. Vol. 33, no. 4. DOI: <https://doi.org/10.1109/TPAMI.2010.116>
- [20] Garcia A.L., Ruiz de Miras J., Feito F.R. Evaluation of Boolean operations between free-form solids using extended simplicial chains and PN triangles // *The Visual Computer*. 2011. Vol. 27, issue 6-8. Pp. 531-541. DOI: <https://doi.org/10.1007/s00371-011-0566-y>
- [21] Hoffmann C. Geometric and solid modeling. Morgan Kaufmann, San Mateo, California, 1989.
- [22] Lo S.H., Wang W.X. Finite element mesh generation over intersecting curved surfaces by tracing of neighbours // *Finite Elements In Analysis And Design*. 2005. Vol. 41, no. 4. Pp. 351-370. DOI: <http://dx.doi.org/10.1016/j.finel.2004.07.002>
- [23] Pavic D., Campen M., Kobbelt L. Hybrid booleans // *Computer Graphics Forum*. 2010. Vol. 29, issue 1. Pp. 75-87. DOI: <https://doi.org/10.1111/j.1467-8659.2009.01545.x>
- [24] Industrial application of exact Boolean operations for meshes / M. Schifko, B. Juttler, B. Kornberger // *Proceedings of the 26th Spring Conference on Computer Graphics, (SCCG '10)*. ACM, New York, NY, USA, 2010. Pp. 165-172. DOI: <http://dx.doi.org/10.1145/1925059.1925089>

Поступила 28.11.2017; принята к публикации 10.02.2018; опубликована онлайн 30.03.2018.

## REFERENCES

- [1] Mei G, Tipper J.C. Simple and robust boolean operations for triangulated surface. *Computing Research Repository*. 2013. Vol. abs/1308.4434. Pp. 1-18. Available at: <https://arxiv.org/pdf/1308.4434.pdf> (accessed 10.02.2018).
- [2] Ernst M., Greiner G. Early split clipping for bounding volume hierarchies. *Proceedings of the IEEE Symposium on Interactive Ray Tracing*. 2007. Pp. 73 - 78. DOI: <http://doi.org/10.1109/RT.2007.4342593>
- [3] Bergen G.V.D. Efficient Collision Detection of Complex Deformable Models using AABB Trees. *Journal of Graphics Tools*. 1997; 2(4):1-13. DOI: <https://doi.org/10.1080/10867651.1997.10487480>
- [4] Baumgart B.G. Winged Edge Polyhedron Representation. Technical Report, Stanford University, Stanford, CA, USA. 1972.
- [5] Skvortsov A.V. Delaunay triangulation and its application. Tomsk: Publishing house Tom. University. 2002. 128 p. (In Russian)
- [6] Sâm Landier. Boolean operations on arbitrary polygonal and polyhedral meshes. *Computer-Aided Design*. 2017; 85:138-153. DOI: <https://doi.org/10.1016/j.cad.2016.07.013>
- [7] Kugler M., Hostettler A., Soler L., Rémond Y., George D. A new algorithm for volume mesh refinement on merging geometries: Application to liver and vascularization. *Journal of Computational and Applied Mathematics*. 2018; 330(C):429-440. DOI: <https://doi.org/10.1016/j.cam.2017.09.012>
- [8] Bagley B., Sastry Sh.P., Whitaker R.T. A Marching-tetrahedra Algorithm for Feature-preserving Meshing of Piecewise-smooth Implicit Surfaces. *Procedia Engineering*. 2016; 163:162-174. DOI: <https://doi.org/10.1016/j.proeng.2016.11.042>
- [9] Xu S., Keyser J. Fast and robust Booleans on polyhedral. *Computer-Aided Design*. 2013; 45(2):529-534. DOI: <https://doi.org/10.1016/j.cad.2012.10.036>
- [10] Attene M. Direct repair of self-intersecting meshes. *Graphical Models*. 2014; 76(6):658-668. DOI: <https://doi.org/10.1016/j.gmod.2014.09.002>
- [11] Zhoufang Xiao, Jianjun Chen, Yao Zheng, Jianjing Zheng, Desheng Wang. Booleans of triangulated solids by a boundary conforming





- tetrahedral mesh generation approach. *Computers & Graphics*. 2016; 59(C):13-27. DOI: <http://dx.doi.org/10.1016/j.cag.2016.04.004>
- [12] Monterde D.L., Martinez J., Vigo M., Pla N. A practical and robust method to compute the boundary of three-dimensional axis-aligned boxes. *GRAPP 2014 - Proceedings of the 9th International Conference on Computer Graphics Theory and Applications*. 5-8 Jan. 2014. Pp. 34-42. Available at: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7296027&isnumber=7295996> (accessed 10.02.18).
- [13] Guo K.B., Zhang L.C., Wang C.J. et al. Boolean operations of STL models based on loop detection. *International Journal of Advanced Manufacturing Technology*. 2007; 33(5-6):627-633. DOI: <https://doi.org/10.1007/s00170-006-0487-5>
- [14] Wang Ch. Approximate Boolean Operations on Large Polyhedral Solids with Partial Mesh Reconstruction. *IEEE transactions on visualization and computer graphics*. 2011; 17(6):836-849. DOI: <https://doi.org/10.1109/TVCG.2010.106>
- [15] Jiang X., Peng Q., Cheng X., Dai N., Cheng Ch., Li D. Efficient Booleans algorithms for triangulated meshes of geometric modeling. *Computer-Aided Design and Applications*. 2016; 13(4):419-430. DOI: <https://doi.org/10.1080/16864360.2015.1131530>
- [16] Chen, Ming & Yu Chen, Xiao & Tang, Kai & Yuen, Matthew. Efficient Boolean Operation on Manifold Mesh Surfaces. *Computer-Aided Design and Applications*. 2013; 7:405-415. DOI: <https://doi.org/10.3722/cadaps.2010.405-415>
- [17] Lo S.H., Wang W.X. A fast robust algorithm for the intersection of triangulated surfaces. *Engineering with Computers*. 2004; 20(1):11-21. DOI: <https://doi.org/10.1007/s00366-004-0277-3>
- [18] Campen M., Kobbelt L. Exact and Robust (Self-)Intersections for Polygonal Meshes. *Computer Graphics Forum*. 2010; 29:397-406. DOI: <https://doi.org/10.1111/j.1467-8659.2009.01609.x>
- [19] Zaharescu A., Boyer E., Horaud R. Topology-Adaptive Mesh Deformation for Surface Evolution, Morphing, and Multiview Reconstruction. *IEEE Transactions on pattern analysis and machine intelligence*. 2011; 33(4). DOI: <https://doi.org/10.1109/TPAMI.2010.116>
- [20] Garcia A.L., Ruiz de Miras J., Feito F.R. Evaluation of Boolean operations between free-form solids using extended simplicial chains and PN triangles. *The Visual Computer*. 2011; 27(6-8):531-541. DOI: <https://doi.org/10.1007/s00371-011-0566-y>
- [21] Hoffmann C. Geometric and solid modeling. Morgan Kaufmann, San Mateo, California, 1989.
- [22] Lo S.H., Wang W.X. Finite element mesh generation over intersecting curved surfaces by tracing of neighbours. *Finite Elements In Analysis And Design*. 2005; 41(4):351-370. DOI: <http://dx.doi.org/10.1016/j.finel.2004.07.002>
- [23] Pavic D., Campen M., Kobbelt L. Hybrid Booleans. *Computer Graphics Forum*. 2010; 29(1):75-87. DOI: <https://doi.org/10.1111/j.1467-8659.2009.01545.x>
- [24] Schifko M., Juttler B., Kornberger B. Industrial application of exact Boolean operations for meshes. *Proceedings of the 26th Spring Conference on Computer Graphics, (SCCG '10)*. ACM, New York, NY, USA, 2010. Pp. 165-172. DOI: <http://dx.doi.org/10.1145/1925059.1925089>

Submitted 28.11.2017;

Revised 10.02.2018;

Published 30.03.2018.

## About the authors:

**Vladimir V. Kurgansky**, student, the Faculty of Mathematics, P.G. Demidov Yaroslavl State University (14 Sovetskaja St., Yaroslavl 150003, Russia); ORCID: <https://orcid.org/0000-0002-3648-3396>, [Vladimir.kurgansky@gmail.com](mailto:Vladimir.kurgansky@gmail.com)

**Elvira V. Gayduk**, Graduate student, the Faculty of Mathematics (14 Sovetskaja Str., Yaroslavl 150003, Russia); ORCID: <https://orcid.org/0000-0001-7428-8375>, [elja5@yandex.ru](mailto:elja5@yandex.ru)



This is an open access article distributed under the Creative Commons Attribution License which unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited (CC BY 4.0).