

DOI: <https://doi.org/10.15688/mpcm.jvolsu.2025.4.3>

УДК 519.688

Дата поступления статьи: 07.10.2025

ББК 22.19

Дата принятия статьи: 14.10.2025

МЕТОД СДВИГА КРАЙНИХ ТОЧЕК ДЛЯ ПОСТРОЕНИЯ ТРИАНГУЛЯЦИИ ПЛОСКОЙ ОБЛАСТИ, ЗАДАННОЙ В OBJ-ФОРМАТЕ

Алексей Александрович Клячин

Доктор физико-математических наук, заведующий кафедрой математического анализа и теории функций,

Волгоградский государственный университет

Aleksey.klyachin@volsu.ru, klyachin-aa@yandex.ru

Просп. Университетский, 100, 400062 г. Волгоград, Российская Федерация

Аннотация. В работе представлен метод построения триангуляции плоской области, граница которой задается формате obj-файла. Идея нашего подхода заключается в следующем. На первом шаге плоскость равномерно разбивается на квадраты со стороной $h > 0$. На втором шаге строится триангуляция фигуры, образованной квадратами, у которых хотя бы три вершины попадают в область. Окончательное построение триангуляции осуществляется на третьем шаге: определяется векторное поле, вдоль которого крайние узлы сдвигаются на границу области по определенному правилу. Разработано соответствующее программное обеспечение и построена триангуляция некоторых фигур, полученных их моделированием в системе Blender.

Ключевые слова: триангуляция области, сдвиг крайних узлов, расчетная сетка, разбиение плоскости.

Введение

Пусть задан конечный набор точек $\{P_i\}_{i=1}^m$ на плоскости \mathbf{R}^2 . Триангуляцией данного набора точек называется совокупность невырожденных треугольников $\mathcal{T} = \{T_j\}_{j=1}^N$ такая, что

1) вершины треугольников принадлежат множеству точек $\{P_i\}_{i=1}^m$ и любая точка P_i является вершиной хотя бы одного треугольника;

2) треугольники либо имеют общую вершину, либо общее ребро, либо не имеют общих точек.

Отметим, что всякая точка P_i является вершиной некоторого набора треугольников из триангуляции \mathcal{T} . Если сумма углов при вершине P_i по таким треугольникам равна 2π , то узел P_i будем называть внутренним узлом триангуляции. В противном случае узел назовем крайним.

Пусть Ω ограниченная область в \mathbf{R}^2 . Триангуляцией области Ω называется триангуляция произвольного конечного набора точек, лежащего в замыкании области Ω . Известны различные подходы к построению расчетной треугольной сетки (триангуляции). Например, берется конечное число точек, лежащих в $\bar{\Omega}$, и формируется на их основе триангуляция одним из соответствующих алгоритмов (см., например, [1–3]). Имеются и другие способы, в которых применяются либо вариационные принципы, либо разрезание (измельчение) области (или ее границы) на простые части, либо решение различных краевых задач, либо построение выпуклой триангуляции Делоне с последующим удалением лишних треугольников и т. д. (см., например, [4–15]).

В настоящей работе представлен метод, общая идея которого заключается в следующем. Строится начальная триангуляция области в виде набора треугольников, рассчитанных по координатной сетке. После этого крайние узлы сдвигаются на границу области по определенному правилу. Данный способ применяется для областей, которые ограничены замкнутыми ломаными линиями, что позволяет их достаточно легко моделировать в среде **Blender**, сохранив в формате **obj**. Это, в свою очередь, даст возможность автоматизировать процесс триангуляции, не прибегая к трудоемкому шагу выписывания математических формул, задающих границу области. Отметим также, что подход, связанный со сдвигом крайних узлов на границу области не является новым. Например, он был применен для построения пространственной сетки в работе [16]. Отличие нашего алгоритма заключается в том, что в этой работе область задается одним неравенством и несколько иначе осуществляется сам сдвиг крайних узлов на границу области.

1. Общая идея метода

Рассмотрим разбиение плоскости (x, y) прямыми $x = ih$, $y = jh$, где $i, j = 0, \pm 1, \pm 2, \dots$ и h – произвольное положительное число. Будем считать, что область Ω задана своей границей $\partial\Omega$, которая определяется конечным набором замкнутых не пересекающихся друг с другом ломаных линий $\Gamma_0, \Gamma_1, \dots, \Gamma_p$. При этом внешним контуром считаем ломаную линию Γ_0 . Обозначим через L_1, L_2, \dots, L_n все звенья этих ломаных линий (см. рисунок 1). Будем строить начальную триангуляцию следующим образом. Зафиксируем целые числа i, j и рассмотрим квадрат с вершинами (ih, jh) , $((i+1)h, jh)$, $((i+1)h, (j+1)h)$, $(ih, (j+1)h)$. В триангуляцию включаем оба треугольника, которые получаются разбиением данного квадрата некоторой его диагональю при условии, что все вершины такого квадрата попадают в область Ω . Ниже будет описан и способ определения принадлежности точки области Ω и выбор диагонали. В том случае, когда в области Ω оказываются только 3 вершины этого квадрата, в триангуляцию включаем треугольник, который образуют эти вершины. В остальных случаях такой квадрат не используем для построения триангуляции. Пусть $\{P_i = (x_i, y_i)\}_{i=1}^m$ – множество узлов построенной триангуляции, а $\mathcal{T}_h = \{T_k\}_{k=1}^N$ – все ее треугольники.

Построенная начальная триангуляция (см. рисунок 2, левая часть) будет, вообще говоря, слабо приближать границу области $\partial\Omega$ из-за того, что у треугольников триангуляции \mathcal{T}_h катеты параллельны осям координат, а гипотенузы образуют с ними угол $\frac{\pi}{4}$. Для того, чтобы триангуляция хорошо аппроксимировала границу $\partial\Omega$, выполним сле-

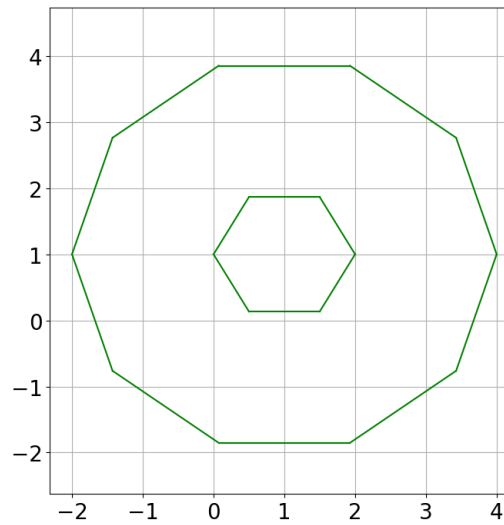


Рис. 1. Область, заданная набором замкнутых ломаных.

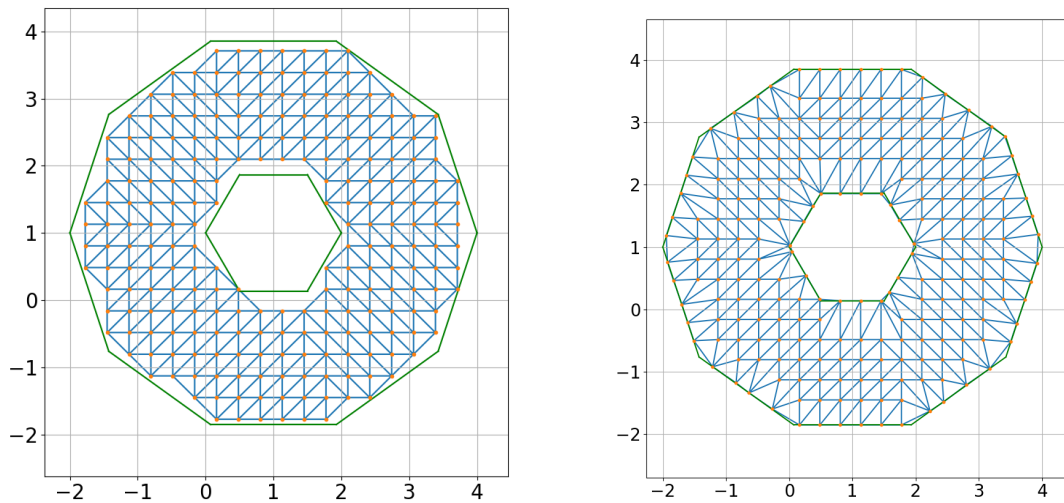


Рис. 2. Вид начальной (слева) и окончательной (справа) триангуляции \mathcal{T}_h^* .

дующие действия. Во всех треугольниках T_k , в которых есть вершины P_i , являющиеся крайними узлами триангуляции \mathcal{T}_h , заменим эти вершины на точку $P'_i \in \partial\Omega$, которую будем определять следующим правилом. Найдем вначале ближайший отрезок L_k , в полосу которого попадает крайний узел, а затем найдем ближайшую вершину V_j ломаных линий Γ_s , $s = 0, 1, \dots, p$. Тогда точка $P'_i \in \partial\Omega$ определяется так

$$P'_i = \begin{cases} V_j, & \text{если } |V_j - P_i| < \text{dist}(P_i, L_k) \\ Q \in L_k : |Q - P_i| = \text{dist}(P_i, L_k), & \text{если } |V_j - P_i| \geq \text{dist}(P_i, L_k). \end{cases}$$

Для упрощения процедуры вычисления точки P'_i мы строим векторное поле $\vec{E}(x, y)$, вдоль которого сдвигается точка P_i до искомой точки P'_i . Получаем новую триангуляцию T^*_h (рисунок 2, правая часть), которая помимо внутренних точек области Ω будет содержать узлы, лежащие на $\partial\Omega$.

Теперь поясним некоторые детали алгоритма, которые мы упомянули выше. Для определения принадлежности точки области Ω мы воспользовались стандартным подходом, описанным, например, в [17]. Именно, вычисляется количество пересечений луча, выходящего из точки (x, y) и направленного вертикально вверх, с границей области. Если это количество четное, то считаем, что точка находится вне области Ω . В противном случае точка принадлежит области.

Поясним теперь способ выбора диагонали, разбивающей квадраты на два треугольника. При построении начальной триангуляции может возникнуть ситуация, когда у некоторого треугольника все его вершины будут крайними узлами триангуляции. В этом случае, при сдвиге этих узлов на границу области, треугольник будет почти вырожденным для малых значений h . Чтобы избежать этого, мы будем проводить диагональ в квадрате следующим образом. Пусть точка Q является центром квадрата, а Q' – ближайшая к ней точка границы $\partial\Omega$, лежащая на отрезке L_k . Тогда возьмем ту диагональ квадрата, которая образует меньший угол с нормалью к L_k .

И, наконец, триангуляцию можно немного поправить с помощью стандартного метода, основанного на том, чтобы любой внутренний узел стал центром множества вершин многоугольника, составленного из треугольников, окружающих этот узел. Тогда сама поправка будет выглядеть так. Пробегаем по всем внутренним узлам P_i и для каждой из них определяем вершины P_{i_1}, \dots, P_{i_r} , являющиеся соседними с P_i . После чего вершину P_i заменяем на

$$P'_i = \frac{1}{r} \sum_{j=1}^r P_{i_j}.$$

Данную процедуру (цикл по всем внутренним узлам триангуляции) делаем несколько раз, пока не будет достигнута необходимая точность. В итоге получается исправленная триангуляция с оптимизированной формой ячеек (рисунок 3)

2. Описание алгоритма триангуляции

Приведем схему алгоритма описанной выше триангуляции. Мы считаем, что задан набор точек $\{V_j\}_{j=1}^q$, которые соединяются отрезками $\{L_k\}_{k=1}^l$ таким образом, что образуют набор ломаных замкнутых линий $\Gamma_0, \Gamma_1, \dots, \Gamma_p$. Предполагается, что каждая из этих линий ограничивает некоторый многоугольник на плоскости, при этом $\Gamma_s \cap \Gamma_t = \emptyset$ для $s \neq t$.

- 1) Рассмотрим квадрат $[a, b] \times [a, b]$, который содержит область Ω вместе со своей границей. В этом квадрате строится векторное поле $\vec{E}(x, y)$ следующим образом (рисунок 4). Перебираем все отрезки L_k и среди них находим ближайший к точке (x, y) , который удовлетворяет условию

$$|\alpha(x - x_c) + \beta(y - y_c)| \leq \frac{1}{2}|B' - B''|\sqrt{\alpha^2 + \beta^2},$$

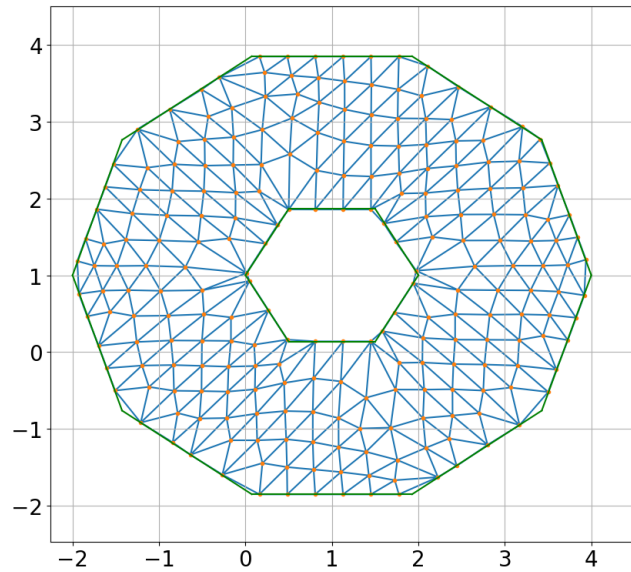


Рис. 3. Вид исправленной триангуляции T_h^* .

где точки $B' = (x', y')$, $B'' = (x'', y'')$ являются концами отрезка L_k ,

$$x_c = \frac{x' + x''}{2}, \quad y_c = \frac{y' + y''}{2}$$

и

$$\alpha = x'' - x', \quad \beta = (y'' - y').$$

Если расстояние от точки (x, y) до найденного отрезка не превосходит расстояния до ближайшего узла $V_j = (\hat{x}, \hat{y})$, то полагаем

$$\vec{E}(x, y) = \text{sgn}(\beta x - \alpha y + \gamma) \left(\frac{\beta}{\sqrt{\alpha^2 + \beta^2}}, -\frac{\alpha}{\sqrt{\alpha^2 + \beta^2}} \right),$$

где $\gamma = \beta x' - \alpha y'$. В противном случае,

$$\vec{E}(x, y) = \left(\frac{x - \hat{x}}{\sqrt{(x - \hat{x})^2 + (y - \hat{y})^2}}, \frac{y - \hat{y}}{\sqrt{(x - \hat{x})^2 + (y - \hat{y})^2}} \right).$$

- 2) Положим $x_i = a + i(b-a)/n$, $y_j = a + j(b-a)/n$, где $i, j = 0, 1, \dots, n$ и $A_{ij} = (x_i, y_j)$. Для каждого $i, j = 0, 1, \dots, n-1$ определяем принадлежность вершин (x_i, y_j) , (x_{i+1}, y_j) , (x_i, y_{j+1}) и (x_{i+1}, y_{j+1}) области Ω . Если 3 из этих точек лежат в Ω , а четвертая не принадлежит Ω , то в триангуляцию добавляем треугольник, который образуют эти три вершины. Если все 4 точки лежат в Ω , то в триангуляцию добавляем два треугольника, полученные разбиением квадрата $A_{ij}A_{i+1j}A_{i+1j+1}A_{ij+1}$ одной его диагональю. Диагональ определяется так. Пусть $x_0 = (x_{i+1} + x_i)/2$, $y_0 = (y_{j+1} + y_j)/2$. Если для векторов $\vec{e}_1 = (1, 1)$, $\vec{e}_2 = (-1, 1)$ выполнено

$$|\langle \vec{E}(x_0, y_0), \vec{e}_1 \rangle| > |\langle \vec{E}(x_0, y_0), \vec{e}_2 \rangle|,$$

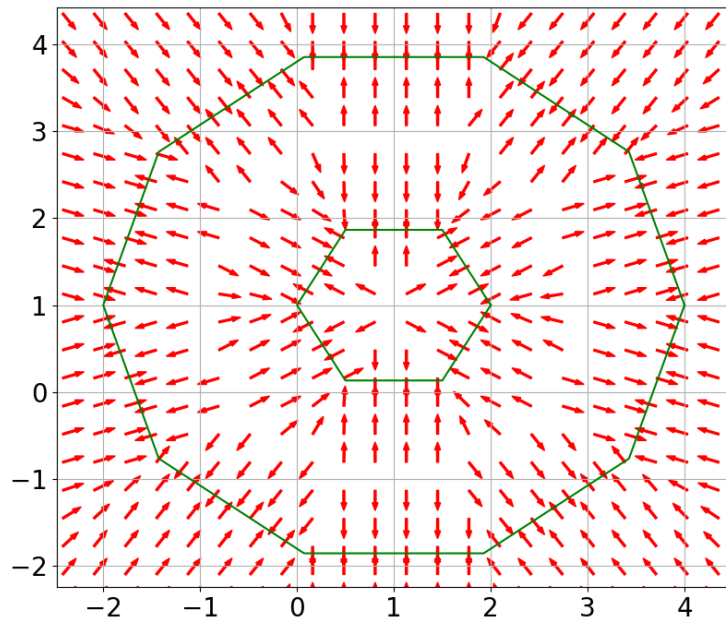


Рис. 4. Векторное поле сдвига крайних узлов.

то выбираем диагональ $A_{ij}A_{i+1j+1}$. В противном случае, берем диагональ $A_{i+1j}A_{ij+1}$. Тем самым будет получена начальная триангуляция (см. рисунок 2).

- 3) Проходим по всем крайним узлам (x, y) триангуляции \mathcal{T}_h и меняем их на ближайшую точку пересечения границы области $\partial\Omega$ с лучом, входящим из узла (x, y) в направлении вектора $\vec{E}(x, y)$. Получаем новую триангуляцию \mathcal{T}_h^* .

3. Примеры триангуляции областей, заданных в obj-формате

Описание границы плоской области может быть дано в виде текстового файла формата obj, например так:

```
# Blender 4.4.0
# www.blender.org
mtllib curves.mtl
o B?zierCircle
v -1.417338 0.000000 0.368239
v -1.453928 0.000000 0.343656
v -1.484190 0.000000 0.316054
v -1.508353 0.000000 0.285619
v -1.526646 0.000000 0.252536
...
l 1 2
l 2 3
l 1 3 4
```

```

1 4 5
1 5 6
...

```

Строки, помеченные буквой «v» содержат координаты вершин замкнутой ломаной линии, а буквой «l» – номера этих вершин отдельного звена ломаной линии. Отметим, что основу алгоритма составляет шаг построения векторного поля $\vec{E}(x, y)$, вдоль которого осуществляется сдвиг на границу области крайних точек начальной триангуляции. Для расчета этого векторного поля и сдвига крайних точек были написаны отдельные функции `field()` и `move_to_bound()` на языке Python, в которые передаются массивы вершин и звеньев ломаных линий, считанные из соответствующего `obj`-файла. Автору статьи были предоставлены два файла (`curves.obj` и `ring.obj`), полученные средствами программы Blender, в которых хранятся границы двух областей.

Рассмотрим первую область (рисунок 5). Данная область задана границей, образованной из двух замкнутых ломаных линий, которые состоят из 53 и 64 звеньев соответственно. Представленным в статье методом была получена триангуляция данной области, состоящая из 467 узлов, соединенных в 808 треугольников (рисунок 6).

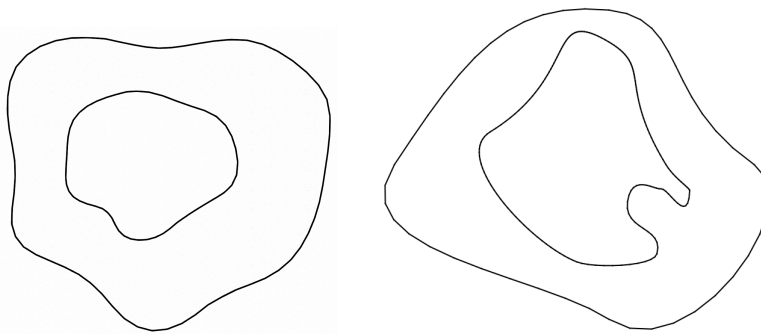


Рис. 5. Первая (слева) и вторая (справа) области

Рассмотрим вторую область (рисунок 5). Данная область задана границей, образованной из двух замкнутых ломаных линий, которые состоят из 192 и 48 звеньев соответственно. Представленным в статье методом была получена триангуляция данной области, состоящая из 1851 узлов, соединенных в 3406 треугольников (рисунок 7).

4. Заключение

В работе был описан метод триангуляции плоской области, которая задана своей границей в виде набора замкнутых ломаных линий. Предполагается, что информация об этих линиях хранится в файлах формата `obj`. Основу алгоритма составляет идея сдвига крайних точек начальной триангуляции на границу области. Для выполнения процесса сдвига строится единое векторное поле, направленное к границе области. Разработано соответствующее программное обеспечение и построена триангуляция двух областей,

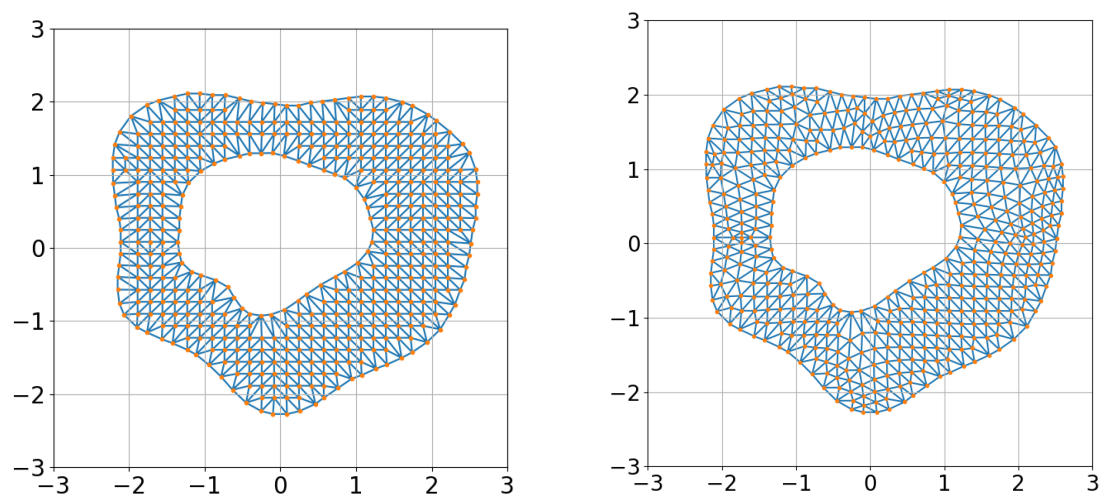


Рис. 6. Триангуляция первой области (слева) и ее исправленный вариант (справа)

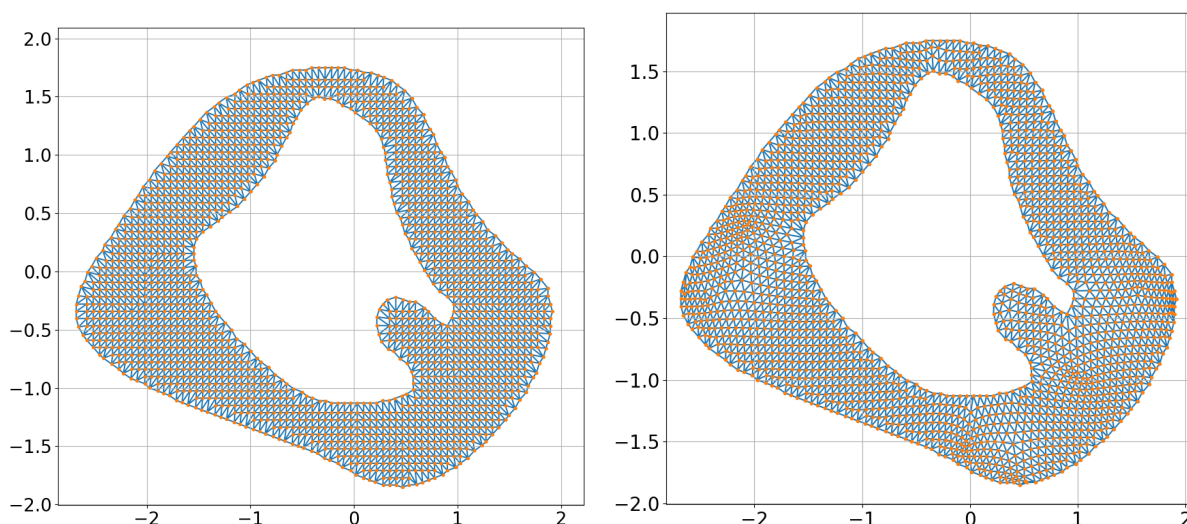


Рис. 7. Триангуляция второй области (слева) и ее исправленный вариант (справа)

границы которых получены моделированием в среде **Blender**. Рассмотренный в данной статье подход дает возможность автоматизировать процесс триангуляции, не прибегая к трудоемкому шагу выписывания математических формул, задающих границу области – область строится в среде визуального моделирования, а сохраненный в формате **obj** файл передается на обработку представленному в статье алгоритму, который и завершает процесс триангуляции.

СПИСОК ЛИТЕРАТУРЫ

1. Скворцов, А. В. Обзор алгоритмов построения триангуляции Делоне. / А. В. Скворцов // Вычислительные методы и программирование. — 2002. — № 3. — С. 14–39.

2. Скворцов, А. В. Алгоритмы построения триангуляции с ограничениями. / А. В. Скворцов // Вычислительные методы и программирование. — 2002. — № 3. — С. 82–92.
3. Клячин, В. А. Алгоритм триангуляции, основанный на условии пустого выпуклого множества. / В. А. Клячин // Вестник ВолГУ. Серия 1. Математика. Физика. — 2015. — № 3(28). — С. 27–33.
4. Алейников, С. М. Алгоритм генерации сетки в методе граничных элементов для плоских областей. / С. М. Алейников, А. А. Седаев // Математическое моделирование. — 1995. — № 7(7). — С. 81–93.
5. Немировский, Ю. В. Автоматизированная триангуляция многосвязных областей со сгущением и разрежением узлов. / Ю. В. Немировский, С. Ф. Пятаев // Вычислительные технологии. — 2000. — № 5(2). — С. 82–91.
6. Гаранжа, В. А. Билипшицевы параметризации негладких поверхностей и построение поверхностных расчетных сеток / В. А. Гаранжа // Ж. вычисл. матем. и матем. физ. — 2005. — № 8(45). — С. 1383–1398.
7. Гаранжа, В. А. Регуляризация барьерного вариационного метода построения расчетных сеток / В. А. Гаранжа, И. Е. Капорин // Ж. вычисл. матем. и матем. физ. — 1999. — № 9(39). — С. 1489–1503.
8. Андрекайте, А. А. Вариационные методы построения расчетных сеток для конечно-элементных расчетов в многосвязных областях / А. А. Андрекайте // Вестник научно-технического развития. — 2010. — № 8(36). — С. 1–7.
9. Зорькин, Д. Ю. Триангуляция методом измельчения плоской области, заданной системой неравенств / Д. Ю. Зорькин, И. А. Тарасова, Н. В. Клячина // Вестник кибернетики. — 2025. — № 24(1). — С. 11–18.
10. Караваев, А. С. Алгоритмы построения и перестроения неструктурированных четырехугольных сеток в односвязных областях / А. С. Караваев, С. П. Копысов, А. Б. Пономарев // Вычислительная механика сплошных сред. — 2012. — № 2(5). — С. 144–150.
11. Полянский, Д. Ю. Триангуляция плоских областей решением методом конечных элементов в форме Галеркина задачи Дирихле / Д. Ю. Полянский // Известия высших учебных заведений. Поволжский регион. Физико-математические науки. — 2012. — № 3. — С. 38–46.
12. Гилева, Л. В. Обоснование асимптотической устойчивости алгоритма триангуляции трехмерной области. / Л. В. Гилева, В. В. Шайдуров // Сиб. журн. вычисл. матем. — 2000. — № 3:2. — С. 123–136.
13. Клячин, А. А. Построение триангуляции плоских областей методом измельчения. / А. А. Клячин // Вестник Волгоградского государственного университета. Серия 1, Математика. Физика. — 2017. — № 2(39). — С. 18–28.
14. Бичерахова, О. С. Построение треугольных сеток многосвязных областей на основе триангуляции Делоне / О. С. Бичерахова // Математическая физика и компьютерное моделирование. — 2025. — № 3(28). — С. 37–49.
15. Mandad, M. Guaranteed-quality higher-order triangular meshing of 2D domains / M. Mandad, M. Campen // ACM Trans Graph. — 2021. — Vol. 40. — P. 1–14.
16. Labelle, F. Isosurface Stuffing: Fast Tetrahedral Meshes with Good Dihedral Angles / F. Labelle, J. Shewchuk // ACM Transactions on Graphics. — 2007. — Vol. 26, № 3. — P. 1–10.
17. Шикин, А. В. Компьютерная графика. Полигональные модели. / А. В. Шикин, А. В. Борисков. — М. : ДИАЛОГ- МИФИ, 2001. — 464 с.

REFERENCES

1. Skvortsov A.V. Overview of Delaunay Triangulation Construction Algorithms.. *Vychislitelnye metody i programmirovaniye*. [Computational methods and programming.], 2002, no. 3, pp. 14-39.

2. Skvortsov A.V. Algorithms for Constructing Triangulation with Constraints. *Vychislitelnye metody i programmirovaniye*. [Computational methods and programming.], 2002, no. 3, pp. 82-92.
3. Klyachin V.A. Triangulation Algorithm Based on the Empty Convex Set Condition. *Vestnik VolGU. Seriya 1. Matematika. Fizika* [VolSU Bulletin. Series 1. Mathematics. Physics], 2015, no. 3(28), pp. 27-33.
4. Aleynikov S.M., Sedaev A.A. Mesh Generation Algorithm in the Boundary Element Method for Planar Domains.. *Matematicheskoe modelirovaniye*. [Mathematical modeling.], 1995, no. 7(7), pp. 81-93.
5. Nemirovskiy Yu.V., Pyataev S.F. Automated Triangulation of Multi-Connected Regions with Node Densification and Sparseness.. *Vychislitelnye tekhnologii* [Computing technologies], 2000, no. 5(2), pp. 82-91.
6. Garanzha V.A. Bi-Lipschitz Parametrizations of Non-Smooth Surfaces and Construction of Surface Computational Grids. *Zh. vychisl. matem. i matem. fiz.* [J. Comput. Mathematics and Mathematics Phys.], 2005, no. 8(45), pp. 1383-1398.
7. Garanzha V.A., Kaporin I.E. Regularization of the Barrier Variational Method for Constructing Computational Grids. *Zh. vychisl. matem. i matem. fiz.* [J. Comput. Mathematics and Mathematics Phys.], 1999, no. 9(39), pp. 1489-1503.
8. Andrekayte A.A. Variational Methods for Constructing Computational Meshes for Finite Element Calculations in Multiply Connected Domains. *Vestnik nauchno-tekhnicheskogo razvitiya* [Bulletin of Scientific and Technological Development], 2010, no. 8(36), pp. 1-7.
9. Zorkin D.Yu., Tarasova I.A., Klyachina N.V. Triangulation by the Method of Refining a Flat Domain Defined by a System of Inequalities. *Vestnik kibernetiki* [Bulletin of Cybernetics], 2025, no. 24(1), pp. 11-18.
10. Karavaev A.S., Kopysov S.P., Ponomaryov A.B. Algorithms for Constructing and Reconstructing Unstructured Quadrilateral Meshes in Simply Connected Domains. *Vychislitel'naya mekhanika sploshnykh sred.* [Computational continuum mechanics.], 2012, no. 2(5), pp. 144-150.
11. Polyanskiy D.Yu. Triangulation of Planar Regions by the Finite Element Solution in Galerkin Form of the Dirichlet Problem. *Izvestiya vysshikh uchebnykh zavedeniy. Povolzhskiy region. Fiziko-matematicheskie nauki*. [News of higher educational institutions. Volga region. Physical and mathematical sciences.], 2012, no. 3, pp. 38-46.
12. Gileva L.V., Shaydurov V.V. Justification of the Asymptotic Stability of the Triangulation Algorithm for a Three-Dimensional Domain. *Sib. zhurn. vychisl. matem.* [Sib. J. Comput. Math.], 2000, no. 3:2, pp. 123-136.
13. Klyachin A.A. Construction of Triangulation of Flat Regions by the Refinement Method.. *Vestnik Volgogradskogo gosudarstvennogo universiteta. Seriya 1: Matematika. Fizika* [Bulletin of Volgograd State University. Series 1: Mathematics. Physics.], 2017, no. 2(39), pp. 18-28.
14. Bicherakhova O.S. Construction of Triangular Meshes of Multiply Connected Regions Based on Delaunay Triangulation. *Matematicheskaya fizika i kompyuternoe modelirovaniye*. [Mathematical Physics and Computer Simulation], 2025, no. 3(28), pp. 37-49.
15. Mandad M., Campen M. Guaranteed-Quality Higher-Order Triangular Meshing of 2D Domains. *ACM Trans Graph*, 2021, vol. 40, pp. 1-14.
16. Labelle F., Shewchuk J. Isosurface Stuffing: Fast Tetrahedral Meshes with Good Dihedral Angles. *ACM Transactions on Graphics*, 2007, vol. 26, no. 3, pp. 1-10.
17. Shikin A.V., Boriskov A.V. Computer Graphics. Polygonal Models.. Moscow, DIALOG-MIFI, 2001. 464 p.

TRIANGULATION OF A FLAT REGION GIVEN IN OBJ FORMAT

Aleksey A. Klyachin

Doctor of Physical and Mathematical Sciences, Head of Department of Mathematical Analysis and Function Theory,
Volgograd State University
Aleksey.klyachin@volsu.ru, klyachin-aa@yandex.ru
Prosp. Universitetsky, 100, 400062 Volgograd, Russian Federation

Abstract.

This paper describes a method for triangulating a flat region defined by its boundary as a set of closed polygonal lines. Information about these lines is stored in verb'obj files. The algorithm is based on the idea of shifting the extreme points of the initial triangulation to the region's boundary. To perform this shift, a single vector field directed toward the region's boundary is constructed. Corresponding software was developed, and a triangulation of two regions, the boundaries of which were obtained through modeling in the verb'Blender environment, was constructed. The approach discussed in this paper enables the automation of the triangulation process without the labor-intensive step of writing out mathematical formulas defining the region's boundary. The region is constructed in a visual modeling environment, and the file saved in verb'obj format is passed for processing to the algorithm presented in the paper, which completes the triangulation process.

Key words: triangulation of a region, shift of extreme nodes, computational grid, plane partitioning.