



DOI: <https://doi.org/10.15688/mpcm.jvolsu.2025.3.4>

УДК 514.142.2+514.174.6  
ББК 22.19

Дата поступления статьи: 17.07.2025  
Дата принятия статьи: 27.08.2025

## ПОСТРОЕНИЕ ТРЕУГОЛЬНЫХ СЕТОК МНОГОСВЯЗНЫХ ОБЛАСТЕЙ НА ОСНОВЕ ТРИАНГУЛЯЦИИ ДЕЛОНЕ <sup>1</sup>

**Ольга Сергеевна Бичерахова**

Ассистент кафедры математического анализа и теории функций,  
Волгоградский государственный университет  
mm-231\_965942@volsu.ru  
<https://orcid.org/0009-0000-3089-6029>  
просп. Университетский, 100, 400062 г. Волгоград, Российская Федерация

**Аннотация.** В статье рассмотрена проблема построения триангуляции многосвязных областей. Существующие методы построения треугольных сеток либо предназначены для выпуклых областей, либо очень сложны в программной реализации. В данной работе предложен способ построения треугольных расчетных сеток многосвязных областей, основанный на классической триангуляции Делоне конечной системы точек. Отличительным свойством предложенного метода является то, что алгоритм возвращает триангуляцию многосвязной области, удовлетворяющую условию Делоне. Данный алгоритм отличается простотой реализации и подходит для триангуляции областей, заданных как математически, так и с помощью .obj файла. Показаны примеры триангуляции различных плоских многосвязных областей, которые задаются разными способами.

**Ключевые слова:** треугольная сетка, триангуляция Делоне, многосвязная область, невыпуклая триангуляция, условие Делоне.

### Введение

Бичерахова О.С., 2025  
©

При численном решении краевых задач математической физики в областях сложной формы важным вопросом, который приходится отдельно исследовать, является построение расчетных сеток. Наиболее подходящими при этом являются сетки с треугольными ячейками, так как треугольники (тетраэдры в пространственном случае) представляют собой простейшие линейные фигуры, геометрические характеристики которых достаточно просто вычисляются. Стоит отметить, что триангуляции широко используются и при

решении других задач (см., например, [11; 15; 21]). Имеется довольно большое число работ, в которых описаны различные методы и подходы к построению треугольных сеток (см., например, [1–10; 12–14; 16–18; 20]). Очень часто для соответствующих расчетов используется триангуляция Делоне. Однако непосредственно применить ее для области, не являющейся выпуклой, невозможно, так как треугольники такой триангуляции образуют выпуклый многоугольник (выпуклый многогранник в пространстве). В работе [19] приводятся основные способы построения триангуляции Делоне с ограничениями, которые можно применять в том числе и для построения треугольной сетки в многосвязной невыпуклой области. Однако нужно отметить, что эти алгоритмы довольно сложны при их программной реализации. Как правильно замечено в этой статье, «многие внешне простые алгоритмы требуют учета многочисленных особых случаев, без чего алгоритм на практике оказывается неработоспособным». В данной работе предлагаем достаточно простой подход для построения треугольных сеток в многосвязной области, избавляющий от рассмотрения «многочисленных особых случаев».

### 1. Построение триангуляции

Пусть задана ограниченная область  $\Omega \subset \mathbb{R}^2$ . Триангуляцией области  $\Omega$  будем называть триангуляцию, построенную по некоторому конечному набору точек  $\{P_i\}_{i=1}^m$ , лежащих в замыкании этой области  $\bar{\Omega}$ . Треугольники этой триангуляции будем обозначать через  $T_1, T_2, \dots, T_N$ .

Будем предполагать, что область  $\Omega$  имеет вид

$$\Omega = \Omega_0 \setminus \bigcup_{i=1}^r \bar{\Omega}_i, \quad (1)$$

где  $\Omega_i, i = 0, 1, \dots, r$  – ограниченные односвязные области с кусочно-гладкими границами, удовлетворяющие условиям

$$\bar{\Omega}_i \subset \Omega_0, \quad i = 1, 2, \dots, r, \quad \bar{\Omega}_i \cap \bar{\Omega}_j = \emptyset, \quad i \neq j.$$

Предложенный способ построения сетки основан на применении стандартной выпуклой триангуляции Делоне. Дадим общее описание алгоритма триангуляции данной области без подробностей, связанных со способом задания области и выбора вершин триангуляции. Для начала отметим, что из определения триангуляции Делоне следует, что этот метод построения треугольной сетки подходит только для выпуклых областей. Если построить триангуляцию Делоне на невыпуклом многоугольнике, то триангуляция все равно получится выпуклой (рис. 1). Аналогичная ситуация будет, если область многосвязная, например, когда ограничена несколькими замкнутыми контурами.

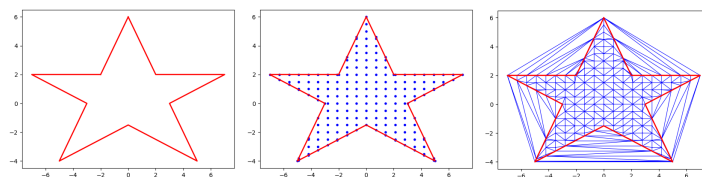


Рис. 1. Исходный многоугольник, добавленные узлы, триангуляция Делоне

На идею для алгоритма натолкнул пример построения триангуляции Делоне системы точек, расположенных на некотором замкнутом контуре.

Рассмотрим такой пример. Предположим, задана замкнутая простая кривая уравнениями

$$\begin{cases} x(t) = (2 + 0.2 \sin 3t) \cos t \\ y(t) = (2 + 0.2 \sin 3t) \sin t, \quad t \in [0, 2\pi]. \end{cases}$$

Пусть  $m$  – некоторое натуральное число и определим систему точек

$$P_i = (x(i \cdot \frac{2\pi}{m}), y(i \cdot \frac{2\pi}{m})), \quad i = 0, 1, \dots, m - 1.$$

Построим триангуляцию Делоне этого множества точек для  $m = 10$  (рис. 2).

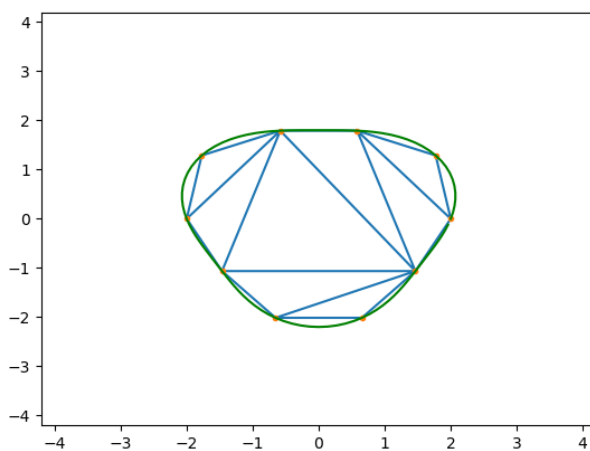


Рис. 2. Триангуляция Делоне точек контура

Можно заметить, что контур достаточно хорошо аппроксимируется сторонами полученных треугольников. Добавим к этому набору узлов точки, лежащие вне контура (рис. 3).

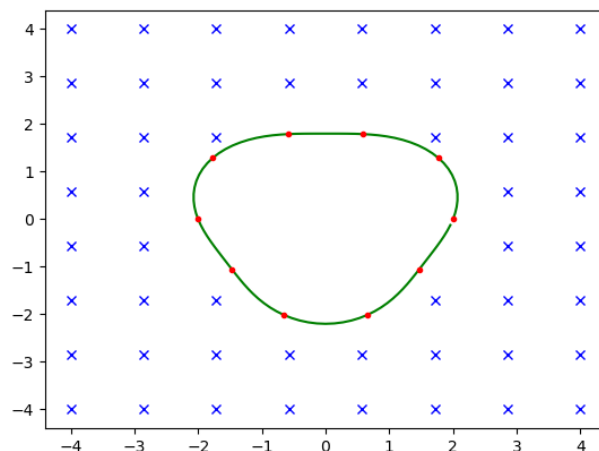


Рис. 3. Добавление точек вне контура

Далее построим триангуляцию Делоне нового множества точек (см. рис. 4).

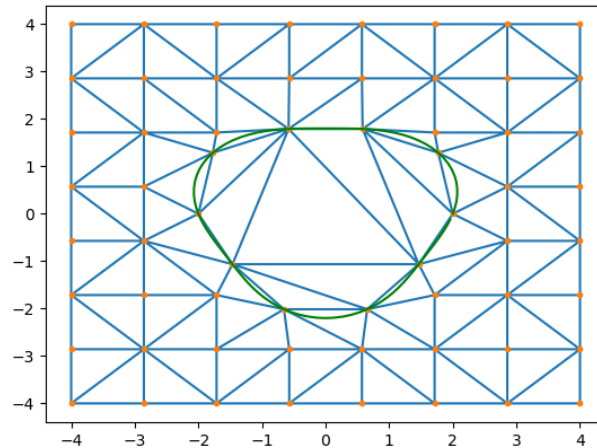


Рис. 4. Триангуляция Делоне нового набора точек

После этого можно просто удалить из построенной триангуляции «лишние» треугольники, например, центры которых лежат внутри контура. В итоге будут удаляться только треугольники, а все выбранные узлы сохраняются за счет того, что внутри контура их нет. Тем самым получим триангуляцию пересечения квадрата  $[-4, 4] \times [-4, 4]$  с внешностью контура (рис. 5). Итак, основная идея предложенного подхода заключается в том, чтобы выбирать узлы будущей триангуляции, не лежащие внутри контуров  $\partial\Omega_i, i = 1, 2, \dots, r$ .

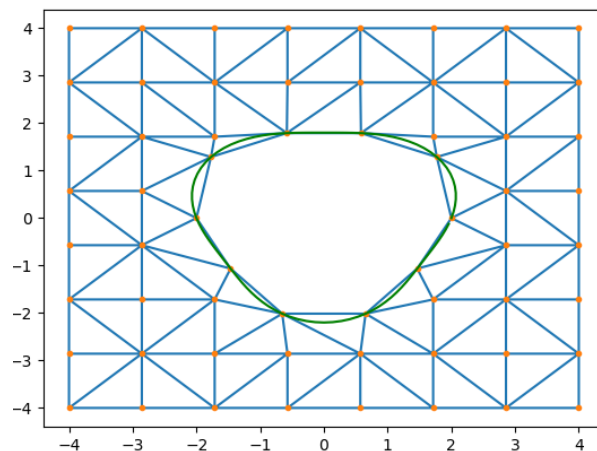


Рис. 5. Триангуляция Делоне без «лишних» треугольников

Таким образом, основываясь на рассмотренном выше простом примере, был получен следующий алгоритм, в котором новая триангуляция  $\mathcal{T}^*$  будет формироваться так.

- 1) Зададим некоторым способом набор точек  $\{P_i\}_{i=1}^m \subset \bar{\Omega}$  (рис. 3).
- 2) Далее рассмотрим выпуклую триангуляцию Делоне  $\mathcal{T}$  набора точек  $\{P_i\}_{i=1}^m$  (рис. 4). Пусть  $\{T_k\}_{k=1}^N$  – множество всех треугольников данной триангуляции.
- 3) Полагаем  $k = 1$ .
- 4) Если  $k \leq N$ , переходим к шагу 5). Иначе алгоритм завершается.

5) Вычислим центрoид

$$O = \frac{1}{3}(A + B + C)$$

треугольника  $T_k$  с вершинами  $A, B, C$ .

6) Если центрoид  $O$  не принадлежит хотя бы одной области  $\overline{\Omega}_i, i = 1, \dots, r$  и принадлежит области  $\Omega_0$ , то добавляем треугольник  $T_k$  в триангуляцию  $\mathcal{T}^*$ ,  $k = k + 1$ , переходим на шаг 4). Иначе  $k = k + 1$ , переходим на шаг 4).

Данный алгоритм был реализован на языке программирования Python с использованием функций библиотеки Shapely. В частности, для определения принадлежности центрoида треугольника к области использовалась функция `within`, возвращающая `True`, если одна геометрия (в нашем случае центрoид треугольника) принадлежит другой (выколотой области). Ниже приведен код, реализующий приведенный выше алгоритм.

```
def form_new_triangulation(T, omega_i, omega_0, P):
new_T = []
for k in range(len(T)):
x = (P[T[k][0]][0] + P[T[k][1]][0] + P[T[k][2]][0]) / 3
y = (P[T[k][0]][1] + P[T[k][1]][1] + P[T[k][2]][1]) / 3
p = Point([x, y])
flag = 0
for i in range(len(omega_i)):
poly = Polygon(omega_i[i])
if (p.within(poly)):
flag = 1
poly = Polygon(omega_0)
if (not p.within(poly)):
flag = 1
if (flag == 0):
new_T.append(T[k])
return new_T
```

Сложность данного алгоритма порядка  $O(Nr)$ , так как для каждого из  $N$  треугольников выполняется проверка принадлежности центрoида  $r$  областям. Если количество треугольников фиксировано, то относительно количества подобластей получаем сложность  $O(r)$ .

**Замечание 1.** Класс областей, для которых применим описанный алгоритм, можно расширить следующим образом. Если некоторые компоненты границы  $\partial\Omega$  не ограничивают никакую односвязную область, то в алгоритме для таких компонент просто не нужно будет удалять лишние треугольники.

Например, если  $\Omega = \Omega_0 \setminus \overline{\Omega}_1$ , где  $\Omega_1$  представляет собой верхнюю полуокружность  $x^2 + y^2 = 1$ , а  $\Omega_0$  – квадрат с вершинами в точках  $(-2, -2), (-2, 2), (2, 2), (2, -2)$ , результат работы алгоритма будет следующий:

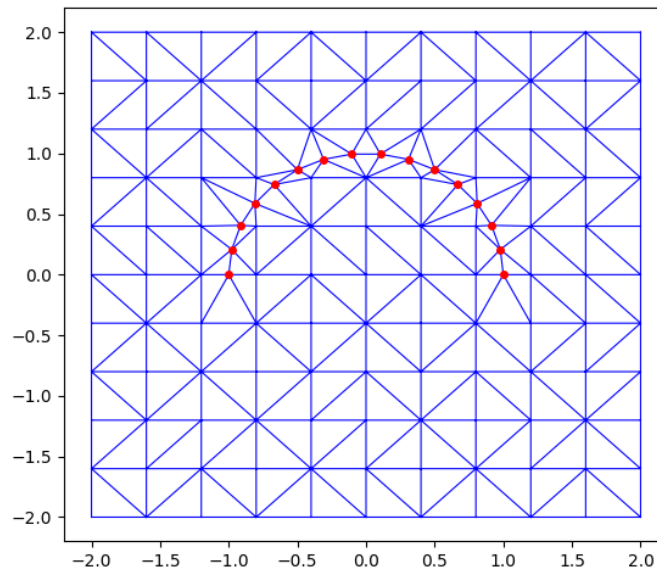


Рис. 6. Триангуляция области, построенной в соответствии с замечанием 1

Результат работы алгоритма триангуляции невыпуклой области продемонстрирован на рисунке 5. Причем, стоит отметить, что полученная триангуляция удовлетворяет условию Делоне [20].

**Теорема 1.** Построенная триангуляция удовлетворяет условию Делоне.

**Доказательство.** Отметим, что исходная триангуляция  $\mathcal{T}$  удовлетворяет условию Делоне. Тогда внутри окружности, описанной вокруг любого треугольника  $T_k$ , не попадает ни одна из заданных точек  $P_i$  триангуляции. Построенная триангуляция  $\mathcal{T}^*$  содержит тот же набор вершин и ее треугольники входят в триангуляцию  $\mathcal{T}$ . Таким образом, условие Делоне для новой триангуляции автоматически сохраняется.

## 2. Примеры триангуляции многосвязных областей

Отметим, что во всех приведенных в данной работе примерах в качестве узлов берутся точки прямоугольной сетки, лежащие в области, к которым добавлены точки, лежащие на соответствующих контурах.

Для начала рассмотрим область вида (1), которая может быть задана системой неравенств. В качестве примера рассмотрим область, заданную неравенствами:

$$\begin{cases} x^2 + y^2 < 100 \\ \frac{(x+2)^2}{9} + \frac{(y-1)^2}{25} > 1 \\ (x-5)^2 + y^2 > 4 \end{cases} \quad (2)$$

На рисунке 7 продемонстрирована работа описанного алгоритма.

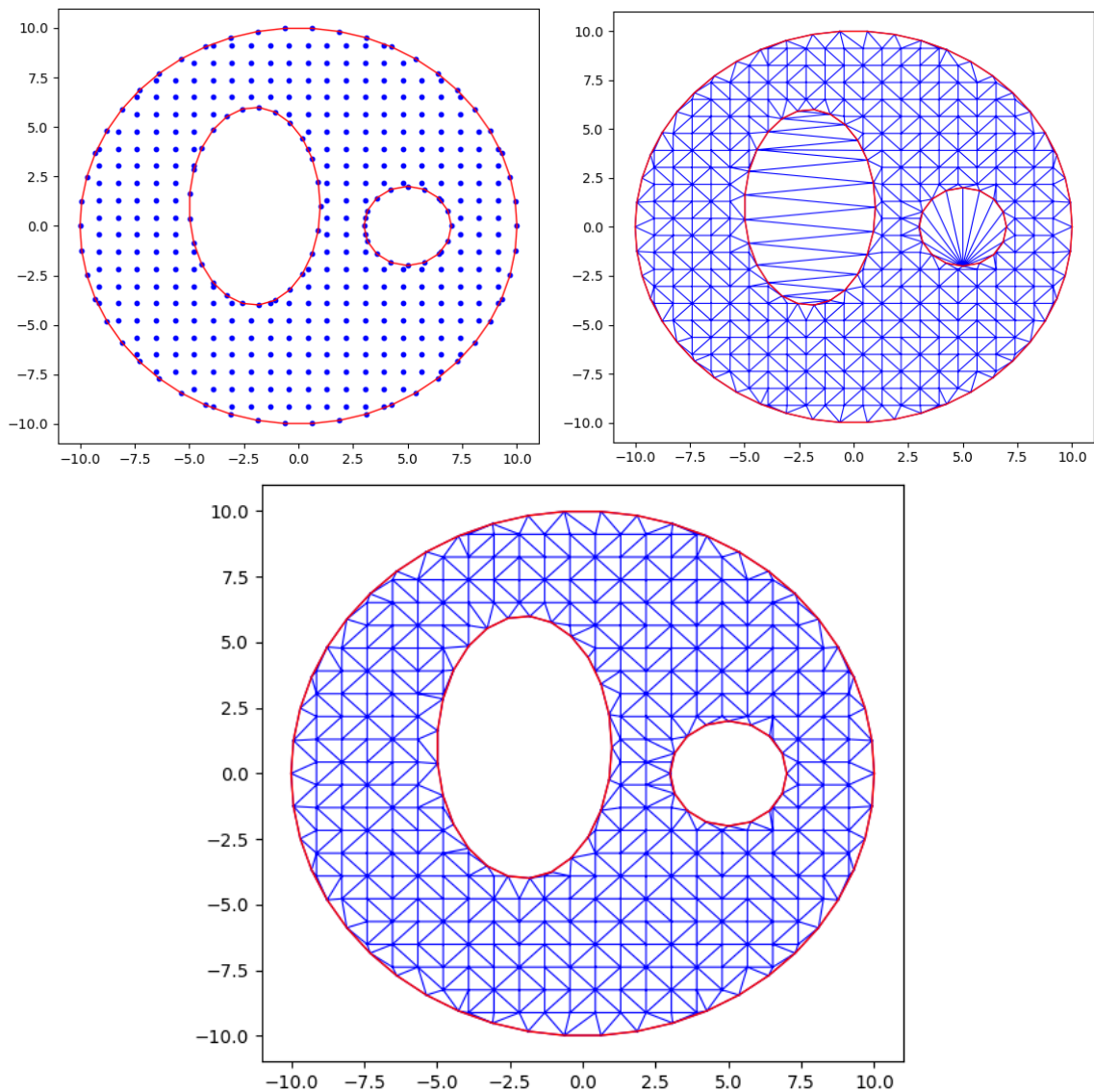


Рис. 7. Процесс построения триангуляции области (2)

Далее рассмотрим область, заданную неравенствами:

$$\left\{ \begin{array}{l} x > -3 \\ x < 3 \\ y > -3 \\ y < 3 \\ (x - 3)^2 + (y - 3)^2 > 1 \\ (x + 3)^2 + (y - 3)^2 > 1 \\ (x - 3)^2 + (y + 3)^2 > 1 \\ (x + 3)^2 + (y + 3)^2 > 1 \\ x^2 + y^2 > 2.25. \end{array} \right. \quad (3)$$

На рисунке 8 продемонстрирован результат работы описанного алгоритма.



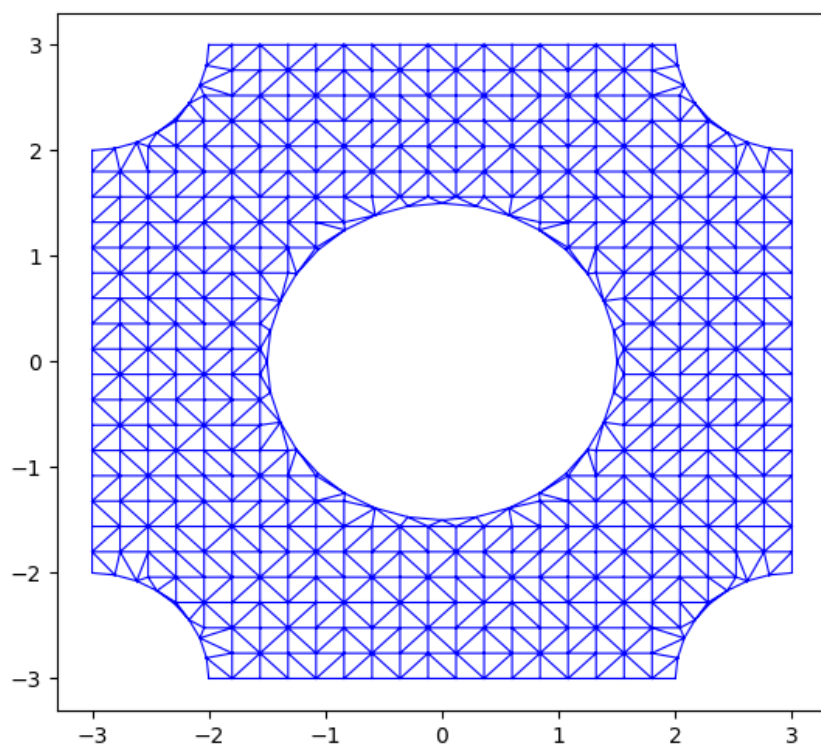


Рис. 8. Результат построения триангуляции области (3)

Еще один пример многосвязной области вида (1) был получен с помощью окружностей, эллипсов и параболических секторов (рис. 9).

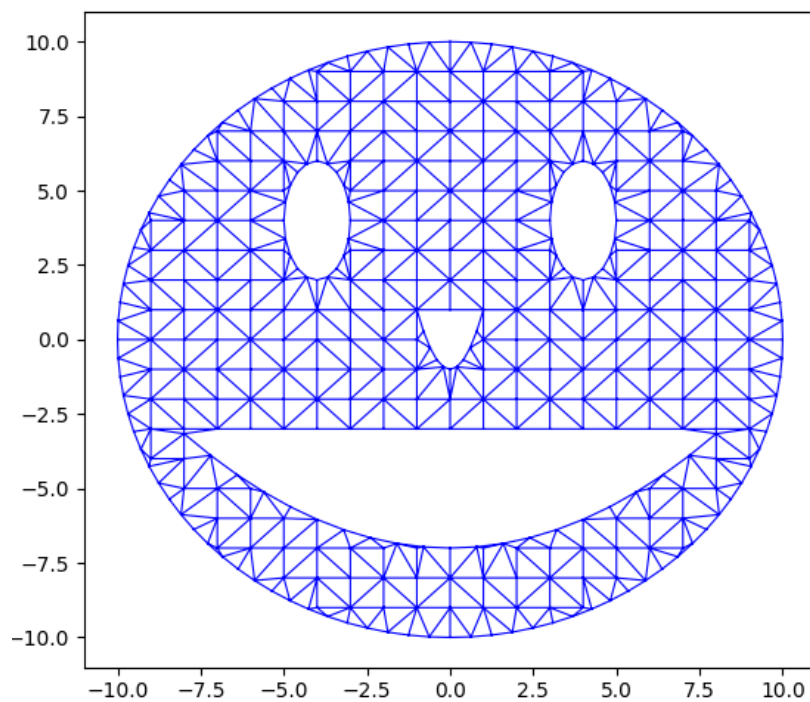


Рис. 9. Триангуляция многосвязной области



Далее можно рассмотреть область, ограниченную следующими кривыми, заданным параметрически:

$$\begin{cases} x(t) = 3 \cos t + \sin t \\ y(t) = 4 \sin(t - 1) - \cos^2 t, \end{cases}$$

$$\begin{cases} x(t) = 1,5 \cos t - \frac{1}{2} \\ y(t) = 2 \sin t + \cos t, \end{cases} \quad (4)$$

$$\begin{cases} x(t) = \cos(t - 1) + 1 \\ y(t) = \sin t - 2, \end{cases}$$

где  $t \in [0, 2\pi]$ .

На рисунке 10 продемонстрирован результат работы описанного алгоритма.

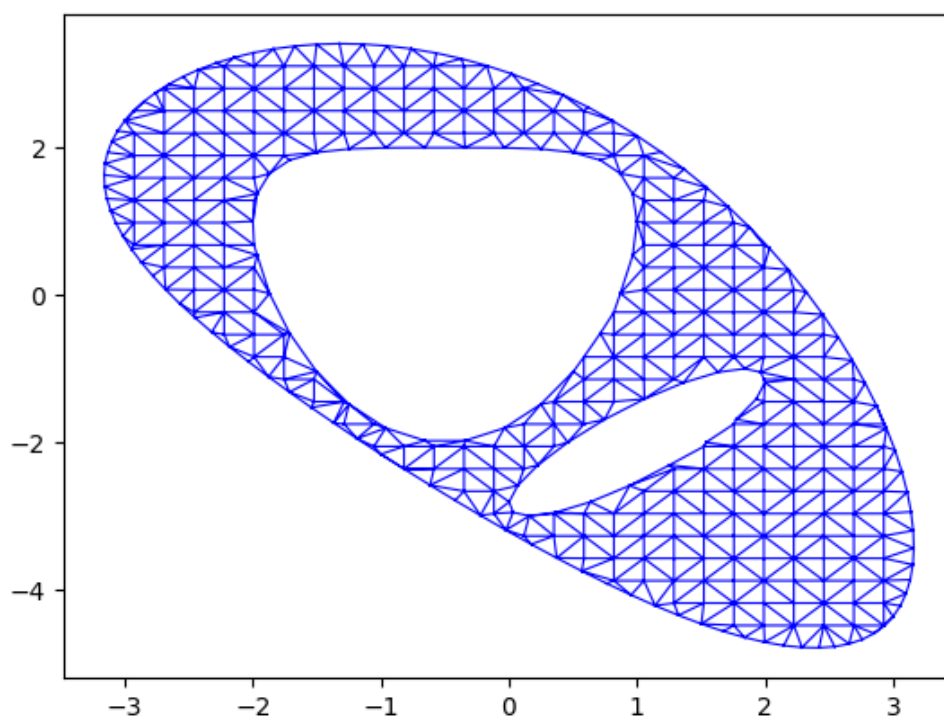


Рис. 10. Результат построения триангуляции области (4)

Область  $\Omega$  так же можно задавать с помощью файла формата .obj, в котором хранятся названия геометрических объектов (строки, начинающиеся с символа 'o'), их вершины (строки, начинающиеся с символа 'v') и ребра (строки, начинающиеся с символа 'l'). Такой файл можно получить, например, построив область в программе для создания трехмерной графики Blender и экспортировав результат в нужном формате. На языке программирования Python была реализована функция Read\_my\_obj, считывающая файл в формате .obj и возвращающая в качестве результата массив Figures, содержащий массивы вершин всех контуров. Ниже приведен листинг описанной программы:

```
def Read_my_obj(file_obj):
    f = open(file_obj)
```

```

Figures = []
Pb = []
for line in f:
    st = line.split()
    if len(st) > 0:
        if st[0] == 'v':
            Pb.append([float(st[1]),float(st[3])])
        if st[0] == 'o' and len(Pb) != 0:
            Figures.append(Pb)
            Pb = []
            Figures.append(Pb)
f.close()
return Figures

```

Таким образом, были получены триангуляции следующих областей (рис. 11):

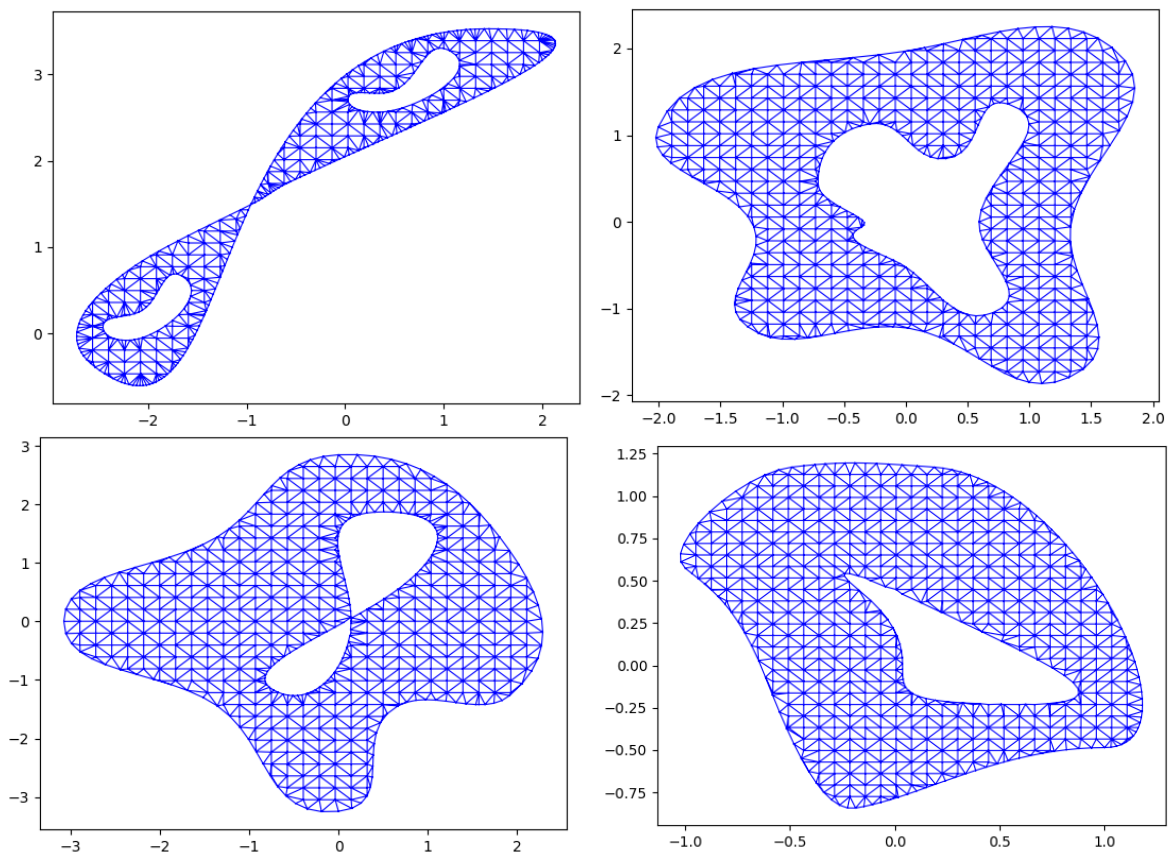


Рис. 11. Результат построения триангуляции областей из .obj файла

### Заключение

В данной работе был рассмотрен алгоритм триангуляции многосвязной области на основе триангуляции Делоне. Предложенный метод триангуляции отличается простотой реализации, что является несомненным преимуществом в сравнении с другими аналогичными алгоритмами. Помимо этого, разработанный метод подходит как для

многосвязных областей, заданных математическим методом (неравенствами), так и для областей, передаваемых с помощью .obj файлов. Еще одним преимуществом представленного метода является то, что данный алгоритм возвращает не просто триангуляцию, а триангуляцию, удовлетворяющую условию Делоне. Представленный в работе алгоритм был протестирован на различных областях. Полученные результаты доказывают работоспособность исследуемого метода и потенциал применения его на практике. Стоит так же сказать, что описанный в статье алгоритм достаточно легко переносится на случай пространственных многосвязных областей вида (1).

### ПРИМЕЧАНИЕ

<sup>1</sup> Автор выражает благодарность своему научному руководителю А. А. Клячину за постоянное внимание к работе и многочисленные полезные обсуждения.

### СПИСОК ЛИТЕРАТУРЫ

1. Алейников, С. М. Алгоритм генерации сетки в методе граничных элементов для плоских областей / С. М. Алейников, А. А. Седаев // Математическое моделирование. — 1995. — Т. 7, № 7. — С. 81–93.
2. 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.
3. Галанин, М. П. Разработка и реализация алгоритмов трехмерной триангуляции сложных пространственных областей: итерационные методы / М. П. Галанин, И. А. Щеглов // Препринт ИПМ им. М.В. Келдыша РАН. — 2006. — № 9. — С. 32.
4. Галанин, М. П. Разработка и реализация алгоритмов трехмерной триангуляции сложных пространственных областей: прямые методы / М. П. Галанин, И. А. Щеглов // Препринт ИПМ им. М.В. Келдыша РАН. — 2006. — № 10. — С. 32.
5. Гаранжа, В. А. Билипшицевы параметризации негладких поверхностей и построение поверхностных расчетных сеток / В. А. Гаранжа // Журнал вычислительной математики и математической физики. — 2005. — Т. 45, № 8. — С. 1383–1398.
6. Гаранжа, В. А. Регуляризация барьерного вариационного метода построения расчетных сеток / В. А. Гаранжа, И. Е. Капорин // Журнал вычислительной математики и математической физики. — 1999. — Т. 39, № 9. — С. 1489–1503.
7. Гилева, Л. В. Обоснование асимптотической устойчивости алгоритма триангуляции трехмерной области / Л. В. Гилева, В. В. Шайдуров // Сибирский журнал вычислительной математики. — 2000. — Т. 3, № 2. — С. 123–136.
8. Годунов, С. К. Об идеях, используемых при построении разностных сеток (навеяно обсуждением на заседании редколлегии 17 января 2002 г. данного номера журнала, посвященного разностным сеткам) / С. К. Годунов // Журнал вычислительной математики и математической физики. — 2003. — Т. 43, № 6. — С. 787–789.
9. Годунов, С. К. О расчетах конформных отображений и построений разностных сеток / С. К. Годунов, Г. П. Прокопов // Журнал вычислительной математики и математической физики. — 1967. — Т. 7, № 5. — С. 1031–1059.
10. Иваненко, С. А. Методы построения адаптивно-гармонических сеток / С. А. Иваненко, Г. П. Прокопов // Журнал вычислительной математики и математической физики. — 1997. — Т. 37, № 6. — С. 643–662.
11. Игумнов, А. Ю. Вычисление локальной характеристики качества элементов тетраэдральной сетки как решение экстремальной задачи / А. Ю. Игумнов // Математическая физика и компьютерное моделирование. — 2024. — Т. 27, № 4. —

С. 5–16. — DOI: <https://doi.org/10.15688/mpcm.jvolsu.2024.4.1>

12. Клячин, В. А. Алгоритм триангуляции, основанный на условии пустого выпуклого множества / В. А. Клячин // Вестник Волгоградского государственного университета. Серия 1, Математика. Физика. — 2015. — № 3 (28). — С. 27–33.

13. Клячин, В. А. Об одном обобщении условия Делоне / В. А. Клячин // Вестн. Томск. гос. ун-та. Матем. и механика. — 2008. — № 1 (2). — С. 48–50.

14. 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.

15. Клячин, А. А. Построение  $C1$ -гладких кусочно-квадратичных функций при решении краевых задач уравнений 4-го порядка на треугольной сетке / А. А. Клячин, И. Ю. Веревкин // Математическая физика и компьютерное моделирование. — 2023. — Т. 26, № 2. — С. 5–15. — DOI: <https://doi.org/10.15688/mpcm.jvolsu.2023.2.1>

16. Немировский, Ю. В. Автоматизированная триангуляция многосвязных областей со сгущением и разрежением узлов / Ю. В. Немировский, С. Ф. Пятаев // Вычислительные технологии. — 2000. — Т. 5, № 2. — С. 82–91.

17. 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.

18. Сакало, В. И. Универсальная программа триангуляции двумерной области произвольной формы со сгущениями сетки / В. И. Сакало, А. А. Шкурин // Проблемы прочности. — 1985. — № 1. — С. 106–108.

19. Скворцов, А. В. Алгоритмы построения триангуляции с ограничениями / А. В. Скворцов // Вычислительные методы и программирование. — 2002. — Т. 3, № 1. — С. 82–92.

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

21. Хижнякова, Е. В. NP-полнота задачи построения графа с минимальным коэффициентом непрямолинейности / Е. В. Хижнякова // Математическая физика и компьютерное моделирование. — 2023. — Т. 26, № 2. — С. 43–51. — DOI: <https://doi.org/10.15688/mpcm.jvolsu.2023.1.3>

**CONSTRUCTION OF TRIANGULAR MESHES OF MULTIPLY CONNECTED DOMAINS BASED ON DELAUNAY TRIANGULATION****Olga S. Bicherakhova**

Assistant Lecturer, Department of Mathematical Analysis and Function Theory,  
Volgograd State University  
mm-231\_965942@volsu.ru  
<https://orcid.org/0009-0000-3089-6029>  
Prosp. Universitetsky, 100, 400062 Volgograd, Russian Federation

**Abstract.** When numerically solving boundary value problems of mathematical physics in areas of complex shape, an important issue that must be studied separately is the construction of computational grids. The most suitable for this are grids with triangular cells, since triangles (tetrahedra in the spatial case) are the simplest linear figures, the geometric characteristics of which are quite easy to calculate. This article considers the problem of constructing triangulation of multiply connected areas. Existing methods for constructing triangular grids are either designed for convex areas or are very difficult to implement in software. In this article, a method for constructing triangular computational grids of multiply connected regions is proposed, based on the classical Delaunay triangulation of a finite system of points. A distinctive feature of the proposed method is that the algorithm returns a triangulation of a multi-connected region that satisfies the Delaunay condition. The idea to use Delaunay triangulation as the basis for the algorithm arose due to its property of maximizing the minimum angle in triangles, which leads to a reduction in too sharp angles in the grid, which negatively affect the accuracy of calculations. Thus, the algorithm returns a triangulation of sufficiently good quality, suitable for further mathematical calculations. A more detailed study of the quality of the resulting grid is a separate interesting task, the study of which is planned in future articles. The algorithm for constructing a triangulation of a multi-connected region presented in this article is simple to implement and is suitable for triangulating regions defined both mathematically and using an .obj file. The article provides code in the Python programming language that implements the presented algorithm, and also shows examples of triangulation of various flat multi-connected regions that are defined in different ways.

**Key words:** triangular mesh, Delaunay triangulation, multiply connected domain, non-convex triangulation, Delaunay condition.