

**ПАРАЛЛЕЛЬНЫЙ СПОСОБ ПОСТРОЕНИЯ СЕТОК ТРЕУГОЛЬНЫХ ЭЛЕМЕНТОВ ПРИ ФУНКЦИОНАЛЬНОМ ПРЕДСТАВЛЕНИИ**

*В статье рассмотрена проблема построения дискретных моделей на базе треугольных элементов для геометрических областей сложной формы. Для представления геометрических областей использован функциональный подход на базе неявных функций. При конструировании функций для сложных геометрических объектов использованы R-функции. Предложен параллельный метод триангуляции на базе фоновой сетки. Проведены вычислительные эксперименты для оценки эффективности разработанного метода. Полученные сетки близки к равномерным при использовании равномерных фоновых сеток.*

*Ключевые слова:* дискретная модель, сетка, треугольник, R-функция, параллельные вычисления.

С.В. ЧОПОРОВ, С.И. ГОМЕНЮК  
Запорізький національний університет**ПАРАЛЛЕЛЬНЫЙ СПОСІБ ПОБУДОВИ СІТОК ТРИКУТНИХ ЕЛЕМЕНТІВ ПРИ ФУНКЦІОНАЛЬНОМУ ПОДАННІ**

*У статті розглянута проблема побудови дискретних моделей на базі трикутних елементів для геометричних областей складної форми. Для подання геометричних областей використано функціональний підхід на базі неявних функцій. При конструюванні функцій для складних геометричних об'єктів використані R-функції. Запропоновано паралельний метод триангуляції на базі фонові сітки. Проведені обчислювальні експерименти для оцінки ефективності запропонованого методу. Отримані сітки близькі до рівномірних при використанні рівномірних фонових сіток.*

*Ключові слова:* дискретна модель, сітка, трикутник, R-функція, паралельні обчислення.

S.V. CHOPOROV, S.I. GOMENYUK  
Zaporizhzhya National University**PARALLEL METHOD FOR TRIANGULAR MESH GENERATION USING FUNCTIONAL REPRESENTATION**

*In herein paper is described the problem of triangular mesh generation for complex geometrical domain. Functional approach is used for a geometrical object representation. Functional approach is based on implicit functions. R-functions are used to construct implicit functions for complex geometric objects. New parallel method for triangulation using background mesh is proposed. The last one generates meshes that are close to uniform for structured background grids. Proposed method is tested using different number of cores.*

*Keywords:* discreet model, mesh, triangle, R-function, parallel computing.

**Постановка проблемы**

Параллельные вычисления являются одной из базовых компьютерных концепций. Идея одновременного использования нескольких параллельных процессов для решения одной задачи подобна идеи коллективного труда для достижения общей цели. Параллельные вычисления относятся к средствам уменьшения времени, необходимого для решения задачи. Следовательно, актуальной проблемой является разработка алгоритмов с использованием параллельных вычислений.

Важной частью проектирования является исследование эксплуатационных характеристик (например, прочности, устойчивости) проектируемых объектов. Для исследования сложных объектов зачастую на практике используют численные методы, в основе которых лежит идея перехода от непрерывной модели объекта к ее дискретному аналогу – дискретной модели. Для моделирования объектов постоянной толщины, как правило, используются двумерные дискретные модели на базе конечных элементов. Одной из наиболее распространенных форм двумерных элементов являются треугольники. При этом для автоматической генерации дискретных моделей необходимо представить информацию о геометрической структуре объекта в форме, «понятной» компьютеру. Весьма распространенным здесь является функциональный подход, в основе которого использование неявных функций. Для логического конструирования неявных функций, соответствующих сложным объектам, часто используется аппарат теории R-функций Рвачева В.Л.

Объект исследования – процесс проектирования и моделирования сложных объектов и систем.

Предмет исследования – методы автоматического построения дискретных моделей с

использованием технологий параллельных вычислений.

**Целью** данной работы является разработка метода построения дискретных моделей с использованием технологий параллельных вычислений для функциональных моделей геометрических объектов.

#### **Анализ последних исследований и публикаций**

Сегодня достаточно развиты методы построения триангуляции Делоне для исходного множества точек, детальный анализ которых проведен в работах А.В. Скворцова [1-2]. Эти методы получили свое применение в задачах компьютерной графики и геоинформационных системах. Однако их применение при функциональном подходе требует построения множества опорных точек, что является самостоятельной сложной задачей.

Также разработаны способы и методы построения триангуляции Делоне для областей, представленных многоугольниками или планарными графами [3-5]. В этой группе методов наибольшее распространение получили модификации метода Рапперта [4], которые по сути являются способами повышения качества сетки, сгенерированной для исходной области.

Методы подвижного фронта [6-7] основаны на идее послойного заполнения исходной области элементами треугольной формы. Исходной информацией для них является дискретная модель границы области. Данные методы весьма универсальны, однако необходимо отметить их высокую ресурсоемкость и относительно низкую скорость работы.

Другая группа методов генерации сеток базируется на использовании базовой или фоновой сетки с адаптацией приграничных узлов к границам области [8-10]. Фоновая сетка может быть относительно просто сгенерирована (например, равномерная), при этом накладывается единственное требование: размер ячеек на границе должен быть меньше минимальной геометрической особенности области. Для построения неравномерных фоновых сеток используется метод Quadtree [11]. На этапе адаптации здесь распространено использование граничной коррекции узлов [8]. Основным недостатком является возможность получения «плохих» или вырожденных элементов около границы области.

Также активно ведутся исследования в области разработки методов параллельного построения триангуляции области. Предложены алгоритмы для параллельного построения триангуляции Делоне для множества опорных точек или исходных планарных графов [12-16]. Проведены исследования в направлении создания параллельных методов на базе подвижного фронта [17] и метода Quadtree [18-19].

Исследования в области построения R-функций для представления сложных геометрических объектов проведены академиком В.Л. Рвачевым и его учениками [20-22]. Существующие работы [23-24] по решению задачи построения триангуляций геометрических объектов, представленных функционально, основаны на последовательном построении опорного множества точек на границе области и триангуляции Делоне на базе существующих методов.

#### **Формулирование задачи исследования**

Пусть исходная двумерная область  $\Omega$  представлена неявной действительной функцией  $F(x, y)$ . Причем  $F(x, y) > 0$  во внутренних точках  $\Omega$ ,  $F(x, y) < 0$  во внешних для  $\Omega$  точках и соответственно  $F(x, y) = 0$  на границе области. Задача состоит в разработке способа построения покрытия треугольниками области  $\Omega$  с учетом возможности одновременного использования нескольких вычислительных устройств (параллельных вычислений).

#### **Изложение основного материала исследования**

Функциональный подход предполагает использование неявных математических функций для представления геометрических объектов. Простейшие формы (примитивы) представляются элементарными функциями. Например, функция

$$\text{circle}(x, y, r) \equiv r^2 - x^2 - y^2$$

соответствует области, ограниченной окружностью радиуса  $r$  с центром в начале координат.

Для логического конструирования неявных функций, соответствующим более сложным областям, используются системы R-функций [20-22]. Наиболее распространенная в практике моделирования геометрических объектов система R-функций  $\mathfrak{R}_0$  имеет вид

$$\begin{aligned} \neg x &\equiv -x; \\ x_1 \wedge x_2 &\equiv x_1 + x_2 - \sqrt{x_1^2 + x_2^2}; \\ x_1 \vee x_2 &\equiv x_1 + x_2 + \sqrt{x_1^2 + x_2^2}, \end{aligned}$$

где  $x_1$  и  $x_2$  – значения неявных функций.

При функциональном подходе исходная область  $\Omega$  представляется неявной функцией  $F(x, y)$ . Неявное представление дает правило для проверки принадлежности точки телу, но не позволяет явно перечислить внутренние и граничные узлы. Для получения граничных узлов, как правило, сканируют область с некоторым шагом и затем в ячейках, содержащих и внутренние, и внешние точки, использует

методы поиска корней функции  $F(x, y)$ . Следовательно, с вычислительной точки зрения перспективным представляется разработка методов на базе фоновой сетки, что позволит строить элементы уже на этапе сканирования области.

Общая идея метода заключается в использовании некоторой фоновой сетки, полностью покрывающей исходную область  $D$  (пример исходной области приведен на рис. 1, а). Фоновая сетка может быть сгенерирована относительно просто (например, равномерная сетка), но размер ячеек должен быть меньше соответствующих им геометрических особенностей объекта  $\Omega$ . Затем все внешние или пересекаемые границей ячейки удаляются, а оставшиеся ячейки формируют начальную сетку (рис., 1, б). Заключительным этапом является построение элементов в приграничном слое (адаптация к границам, рис. 1, в). Этот этап является наиболее сложным. Здесь с учетом особенностей функционального подхода к моделированию перспективным видится техника изоморфизма, предложенная в работах [25-26] для построения четырехугольников и шестигранников на основе метода суперпозиции.

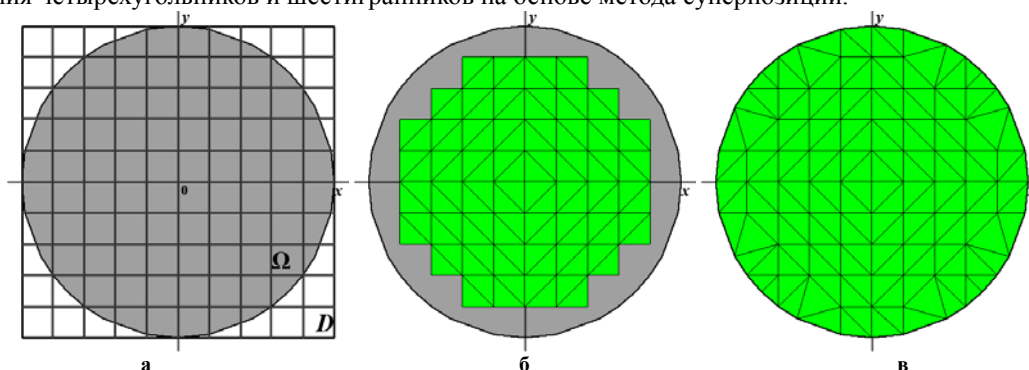


Рис. 1. Исходная область, начальная сетка и адаптированная сетка

Пусть фоновая сетка треугольных элементов представлена множеством

$$M_f = \{t_i\}, \quad i = \overline{1, m},$$

где  $m$  – количество элементов,  $t_i = (p_{i,0}, p_{i,1}, p_{i,2})$  – элемент с номером  $i$ ,  $p_{i,j} = (x_{i,j}, y_{i,j})$ ,  $j = 0, 1, 2$  – координаты  $j$ -го узла  $i$ -го элемента.

Первым шагом является формирование начальной сетки. Начальную сетку  $M_{initial}$  образуют внутренние элементы:

$$M_{initial} = \{t_i : F(p_{i,j}) > 0, \forall j\}.$$

На втором шаге для каждого узла начальной сетки строится нормаль. Нормаль к узлу начальной сетки определим, как среднее арифметическое нормалей к ребрам, инцидентным этому узлу:

$$n(p_l) = \frac{1}{k} \sum_{p_l \in e_{k,l}} \vec{n}(e_{k,l}),$$

где  $e_{k,l}$  – ребро, инцидентное узлу  $p_l$ ,  $\vec{n}(e)$  – операция вычисления нормали к ребру с учетом направления обхода узлов треугольника, которому принадлежит ребро.

На третьем шаге путем нахождения проекции нормали на границу области строится множество изоморфных точек. Необходимо отметить, что при функциональном представлении для нахождения изоморфных точек могут использоваться методы двоичного поиска, золотого сечения и т.п., осуществляя поиск на отрезке длины  $\sqrt{h_x^2 + h_y^2}$  ( $h_x, h_y$  – шаг или средние размеры ячеек фоновой сетки). На этом этапе учитываются характерные точки: если на расстоянии  $\Delta$  от изоморфной находится характерная точка, то в качестве изоморфной используется соответствующая характерная.

Четвертым шагом является построение элементов в приграничном слое. В результате третьего шага каждому узлу на границе начальной сетки соответствует изоморфный на границе области, каждому граничному ребру начальной сетки соответствует ребро на границе области. Следовательно, для каждого граничного ребра начальной сетки можно двумя способами построить элементы (меняя направление диагонали в четырехугольнике, который образуют смежные вершины и изоморфные к ним). Из двух возможных способов формирования треугольников выберем тот, при котором в треугольниках отношение значений минимального угла к максимальному будет наибольшим.

На четвертом шаге могут появляться вырожденные или «плохие» элементы, обусловленные возможностью совпадения или близкого (относительно размеров области) расположения изоморфных узлов. Для решения задачи устранения «плохих» элементов в приграничном слое для соседних узлов  $A$  и  $B$  определим операцию

$$\text{join}(A, B) = \{\text{deledge}(A, B), A = B\},$$

где операция  $\text{deledge}(A, B)$  удаляет ребро  $AB$ . То есть, в результате операции  $\text{join}(A, B)$  будут удалены треугольники, для которых ребро  $AB$  является общим, узел  $A$  будет совмещен с узлом  $B$ .

Учитывая, что «плохие» элементы при таком подходе могут возникать только в приграничном слое, операция  $\text{join}(A, B)$  может применяться только к элементам этого слоя, если расстояние между двумя соседними узлами меньше  $\Delta$ . Если оба узла  $A$  и  $B$  принадлежат границе области, в качестве положения нового узла может быть использованы координаты репроекции середины ребра  $AB$  на границу, осуществляя таким образом локальное сглаживание.

Параметр  $\Delta$  является ограничением на минимальный размер ребра. Эту величину можно интерпретировать как параметр качества сетки. В общем случае  $\Delta$  может зависеть от координат узлов фоновой сетки. Например, если в качестве фоновой используется равномерная сетка, то можно принять

$$\Delta = \delta \sqrt{h_x^2 + h_y^2}, \text{ где } \delta \text{ принимает значения от } 0,2 \text{ до } 0,5.$$

Пусть в вычислительной системе  $m$  функциональных устройств (процессоров, вычислительных ядер и т.п.). Разобьем фоновую сетку на  $m$  подмножеств. Такое разбиение можно осуществлять различными способами: например, разбив область  $D$  на  $m$  горизонтальных или вертикальных полос, или рекурсивно разбив на  $m$  квадрантов (в общем случае разбиение может зависеть от плотности элементов фоновой сетки). Затем к каждому подмножеству можно одновременно применить описанные выше действия. Критическими секциями будут вставка новых узлов и элементов в сетку, а также операция  $\text{join}(A, B)$ . Таким образом алгоритм параллельной генерации дискретной модели примет вид:

1. Генерация фоновой сетки и разбиение ее на  $m$  подмножеств.
2. Для каждого из  $m$  подмножеств параллельно: формируется начальная сетка  $M_{initial}^k$  (первый шаг метода).
3. Синхронизация подмножеств начальной сетки  $M_{initial} = M_{initial}^1 \cup M_{initial}^2 \cup \dots \cup M_{initial}^k$ .
4. Для каждого  $M_{initial}^k$  параллельно:
  - а) вычисление нормалей к узлам начальной сетки (второй шаг метода);
  - б) поиск изоморфных точек (третий шаг метода).
5. Для каждого  $M_{initial}^k$  параллельно: формирование приграничного слоя с использованием операции  $\text{join}(A, B)$  (четвертый шаг метода).

Синхронизация на третьем шаге алгоритма необходима для обеспечения корректности определения направления нормалей. Пятый шаг выполняется последовательно для обеспечения корректности формирования приграничного слоя: при использовании операции  $\text{join}(A, B)$  модифицируется сетка, поэтому эта операция будет критической в системах с общей памятью и требует синхронизации в системах с раздельной памятью.

Предложенный алгоритм реализован в виде программы на языке C++ с использованием библиотеки OpenMP для организации параллельных вычислений. В качестве тестовой платформы использован компьютер с процессором Intel Core i5-3330 3.0GHz (4 ядра) и 8 гигабайтами оперативной памяти, находящийся под управлением операционной системой LinuxMint 17.1 (64 бита, ядро Linux 3.13, компилятор gcc 4.8.2).

1. Модель простой планки (рис. 2, а) может быть представлена формулой

$$\text{planch}(x, y, w, h, r) = \left[ \text{rect}(x, y, w, h) \vee \text{circle}\left(x - \frac{w}{2}, y, \frac{h}{2}\right) \right] \wedge \text{circle}(x, y, r),$$

где функция  $\text{rect}(x, y, w, h) = \left( \frac{w^2}{4} - x^2 \right) \wedge \left( \frac{h^2}{4} - y^2 \right)$  соответствует прямоугольной области

ширины  $w$  и высоты  $h$  (центр прямоугольника совпадает с началом координат).

Графики зависимости времени выполнения от количества узлов в начальной сетке представлены на рис. 2 б, в.

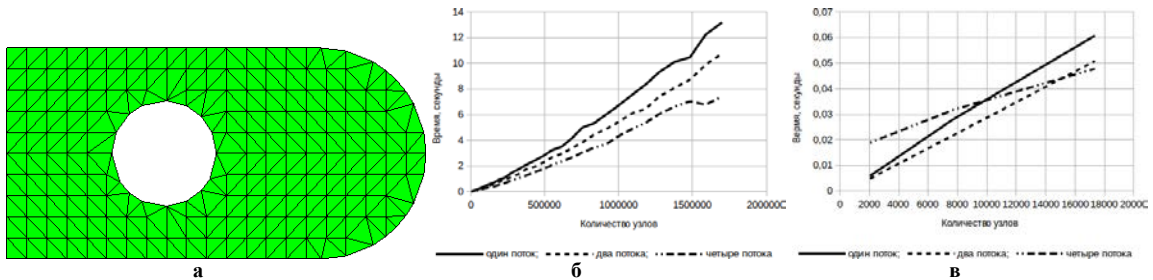


Рис. 2. Исходная область, зависимости времени построения от количества узлов

2. Модель прокладки (рис. 3, а) может быть представлена формулой

$$\text{layer}(x, y) = \left[ \text{circle}(x, y, r_1) \vee \text{circle}(x, y + h_2, r_5) \vee \text{rect}\left(x, y + \frac{h_2}{2}, w_2, h_2\right) \right] \wedge \\ \wedge \neg \text{circle}(x, y, r_2) \wedge \neg \text{rect}\left(x, y - r_2, w_1, 2h_1\right) \wedge \\ \wedge \neg \text{circle}(x - c_{1,x}, y - c_{1,y}, r_4) \wedge \neg \text{circle}(x - c_{2,x}, y - c_{2,y}, r_4) \wedge \\ \wedge \neg \text{circle}(x - c_{3,x}, y - c_{3,y}, r_4) \wedge \neg \text{circle}(x - c_{4,x}, y - c_{4,y}, r_4) \wedge \\ \neg \text{rect}\left(x, y + h_2 - \frac{h_3}{2}, 2r_6, h_3\right) \wedge \neg \text{circle}(x, y + h_2, r_6) \wedge \neg \text{circle}(x, y + h_2 - h_3, r_6),$$

где параметры  $r_1 = \frac{80}{2} = 40$ ,  $r_2 = \frac{40}{2} = 20$ ,  $r_3 = \frac{60}{2} = 30$ ,  $r_4 = \frac{10}{2} = 5$ ,  $r_5 = 24$ ,  $r_6 = 14$ ,  $w_1 = 14$ ,  $h_1 = 5$ ,  $w_2 = 2r_5$ ,  $h_2 = 100$ ,  $h_3 = 48$  соответствуют размерам на чертеже; параметры  $c_{1,x} = r_3 \cos(-\pi/4)$ ,  $c_{1,y} = r_3 \sin(-\pi/4)$ ,  $c_{2,x} = r_3 \cos(\pi/4)$ ,  $c_{2,y} = r_3 \sin(\pi/4)$ ,  $c_{3,x} = r_3 \cos(3\pi/4)$ ,  $c_{3,y} = r_3 \sin(3\pi/4)$ ,  $c_{4,x} = r_3 \cos(5\pi/4)$ ,  $c_{4,y} = r_3 \sin(5\pi/4)$  – центры четырех отверстий.

Графики зависимости времени выполнения от количества узлов в начальной сетке представлены на рис. 3 б, в.

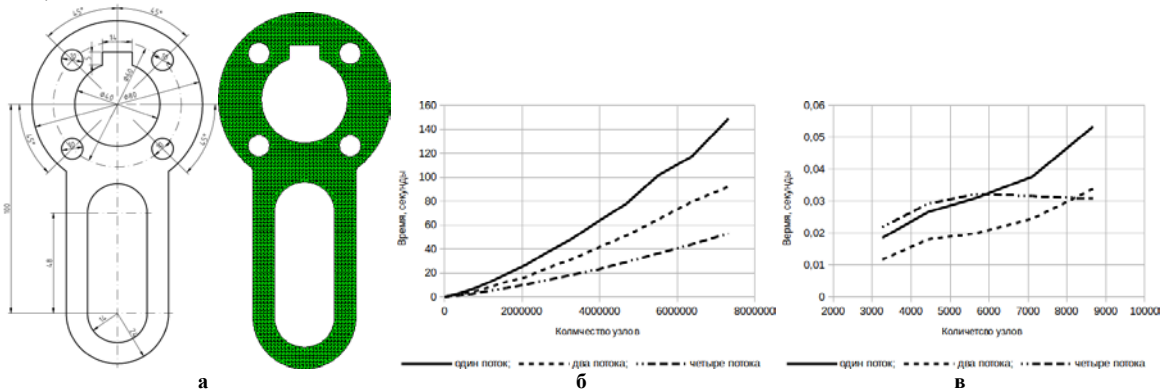


Рис. 3. Исходная область, зависимости времени построения от количества узлов

3. Модель перфорированной пластинки (рис. 4, а) может быть представлена формулой

$$\text{plate}(x, y, a, r) = \text{rect}(x, y, a, a) \wedge \neg \left[ \bigvee_{i=-2}^2 \bigvee_{j=-2}^2 \text{circle}(x - d_i, y - d_j, r) \right],$$

где  $d_i = 3ri$  – параметр сдвига отверстия.

Графики зависимости времени выполнения от количества узлов в начальной сетке представлены на рис. 4 б, в.

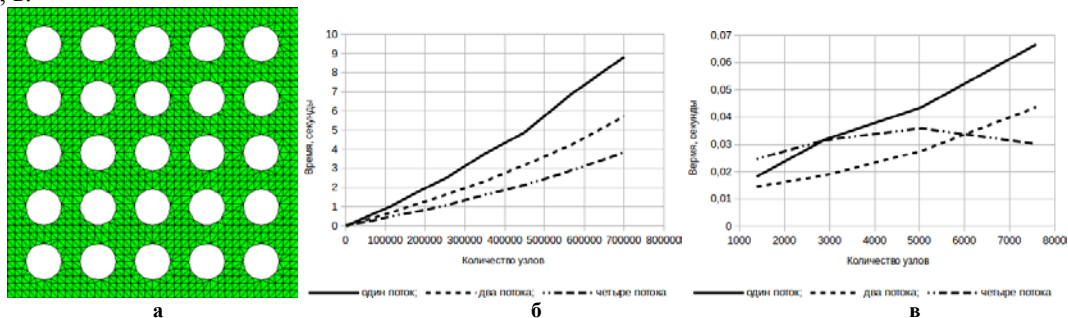


Рис. 4. Исходная область, зависимости времени построения от количества узлов

## Выводы

Таким образом, в статье решена задача автоматизации построения двумерных дискретных моделей для геометрических объектов, представленных функционально. Научная новизна результатов, полученных в статье, состоит в том, что впервые предложен метод построения дискретных моделей с использованием технологий параллельных вычислений для функциональных моделей геометрических объектов на базе фоновой сетки, что позволяет снизить требования к вычислительным ресурсам и уменьшить время, необходимое на моделирование. Практическая значимость полученных результатов заключается в том, что разработано программное обеспечение, реализующее предложенный метод, которое может использоваться в качестве препроцессора метода конечных элементов при исследовании напряженно-деформированного состояния или устойчивости двумерных объектов.

Полученные зависимости времени от количества узлов показывают, что при малом числе узлов (рис. 2, в, 3, в, 4, в) выигрыш во времени при использовании параллельных технологий может отсутствовать, что является подтверждением закона Амадала. Также из приведенных графиков видно, что для моделей, в которых использовано большее количество R-операций, выигрыш от использования параллельных технологий достигается на меньшем количестве узлов: в модели перфорированной пластинки использовано 37 R-операций (выигрыш от четырех потоков заметен, если узлов больше 6000), а в модели простой планки – 3 R-операции (выигрыш от четырех потоков заметен, если узлов больше 14500). Последнее утверждение является следствием того, что временные затраты на вычисление значения модели на базе теории R-функций вносят ощутимый вклад в общее время работы алгоритма.

Перспективой дальнейших исследований является обобщение предложенного метода для генерации дискретных моделей трехмерных тел на базе метода фоновой сетки и технологий параллельных вычислений.

## Список использованной 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. Триангуляция Делоне и её применение / А.В. Скворцов. – Томск: Издательство Томского университета, 2002. – 128 с.
3. Bern M. Mesh generation and optimal triangulation / M. Bern, D. Eppstein // Computing in Euclidean Geometry. – 1992. – No. 1. – P. 23-90.
4. Ruppert J. A Delaunay refinement algorithm for quality 2-dimensional mesh generation / Jim Ruppert //Journal of Algorithms. – 1995. – Volume 18, Issue 3. – P. 548-585.
5. Tournois J. Interleaving Delaunay Refinement and Optimization for 2D Triangle Mesh Generation / Jane Tournois, Pierre Alliez, Olivier Devillers // Proceedings of the 16th International Meshing Roundtable. – 2008. – P. 83-101.
6. Mavriplis D. An advancing front Delaunay triangulation algorithm designed for robustness / D.Mavriplis // Journal of Computational Physics. – 1995. – Volume 117, Issue 1. – P. 90-101.
7. Silva C. Greedy Cuts: An Advancing Front Terrain Triangulation Algorithm / Cláudio T. Silva, Joseph S. B. Mitchell // Proceedings of the 6th ACM international symposium on Advances in geographic information systems. – 1998. – P. 137-144.
8. Галанин М.П. Разработка и реализация алгоритмов трехмерной триангуляции сложных пространственных областей: итерационные методы / М.П. Галанин, И.А. Щеглов. – М.: ИПМ им М.В. Келдыша РАН, 2006. – № 9. – 32 с. – (Препринт / РАН, ИПМ им. М.В. Келдыша ; 06-01-00421).
9. Naumann U. Combinatorial Scientific Computing / [edited by] Uwe Naumann, Olaf Schenk. – Boca Raton, Florida: CRC Press, 2012. – 600 p.
10. Cheng S.-W. Delaunay Mesh Generation / Siu-Wing Cheng, Tamal Krishna Dey, Jonathan Richard Shewchuk. – Boca Raton, Florida: CRC Press, 2012. – 387 p.
11. Samet H. The Quadtree and Related Hierarchical Data Structures / Hanan Samet // ACM Computing Surveys. – 1984. – Volume 16, Issue 2. – P. 187-260.
12. Chrisochoides N. Parallel Delaunay mesh generation kernel / Nikos Chrisochoides, Demian Nave //International Journal for Numerical Methods in Engineering. – 2003. – Volume 58. – P. 161-176.
13. Linardakis L. Delaunay Decoupling Method for Parallel Guaranteed Quality Planar Mesh Refinement /Leonidas Linardakis, Nikos Chrisochoides // SIAM Journal on Scientific Computing. – 2006. – Volume 27, No 4. – P. 1394-1423.
14. Chernikov A.N. Algorithm 872: Parallel 2D constrained Delaunay mesh generation / A.N. Chernikov, Nikos P. Chrisochoides // ACM Transactions on Mathematical Software. – 2008. – Volume 34, Issue 1. – P. 6:1-6:20.
15. Antonopoulos C.D. Algorithm, software, and hardware optimizations for Delaunay mesh generation on simultaneous multithreaded architectures / Christos D. Antonopoulos, Filip Blagojevic, A.N. Chernikov, Nikos P. Chrisochoides, Dimitrios S. Nikolopoulos // Journal of Parallel and Distributed Computing. – 2009. – Volume 69, Issue 7. – P. 601-612.

16. Chen R. Localizing the Delaunay Triangulation and its Parallel Implementation / Renjie Chen, Craig Gotsman // *Lecture Notes in Computer Science*. – 2013. – Volume 8110. – P. 39-55.
17. Ito Y. Parallel Unstructured Mesh Generation by an Advancing Front Method / Yasushi Ito, Alan M. Shih, Anil K. Erukala, Bharat K. Soni, Andrey Chernikov, Nikos P. Chrisochoides, Kazuhiro Nakahashi // *Mathematics and Computers in Simulation*. – 2007. – Volume 75, Issue 5-6. – P. 200-209.
18. Bern M. Parallel Construction of Quadrees and Quality Triangulations / M. Bern, D. Eppstein, and S.-H. Teng // *International Journal of Computational Geometry & Applications*. – 1999. – Volume 9(6). – P. 517-532.
19. Zhang J. Data Parallel Quadtree Indexing and Spatial Query Processing of Complex Polygon Data on GPUs // Jianting Zhang, Simin You, Le Gruenwald // *ADMS'14 Workshop*. – Hangzhou, 2014. – 12 P.
20. Рвачев В.Л. Введение в теорию R-функций / В.Л. Рвачев, Т.И. Шейко // *Проблемы машиностроения*. – 2001. – Т. 4, No 1–2. – С. 46-58.
21. Максименко-Шейко К.В. R-функции и обратная задача аналитической геометрии в трехмерном пространстве / К. Максименко-Шейко, А.М. Мацевитый, А.В. Толок, Т.И. Шейко // *Информационные технологии*. – 2007. – № 10. – С. 23-32.
22. Максименко-Шейко К.В. R-функции в математическом моделировании геометрических объектов в 3D по информации в 2D / К.В. Максименко-Шейко, Т.И. Шейко // *Вісник Запорізького національного університету: Збірник наукових праць. Фізико-математичні науки*. – 2010. – No 1. – С. 98-104.
23. Лисняк А.А. Способ построения дискретных математических геометрических объектов, заданных с помощью R-функций / А.А. Лисняк // *Вісник Запорізького національного університету: Збірник наукових статей. Фізико-математичні науки*. – 2013. – № 1. – С. 59-69.
24. Лисняк А.А. Дискретизация границы трехмерных моделей геометрических объектов, заданных с помощью R-функций / А.А. Лисняк // *Радіоелектроніка, інформатика, управління*. – 2014. – № 1. – С. 82-88.
25. Schneiders R. A Grid-based Algorithm for the Generation of Hexahedral Element Meshes / R. Schneiders // *Engineering with Computers*. – 1996. – № 12. – P. 168-177.
26. Schneiders R. Octree-based Generation of Hexahedral Element Element Meshes / R. Schneiders, R. Schindler, F. Weiler // *5th Annual International Meshing Roundtable*. – 1996. – P. 205-216.