ПОСТРОЕНИЕ СЕТКИ ДИРИХЛЕ НА ОСНОВЕ ТРИАНГУЛЯЦИИ ДЕЛОНЕ ДЛЯ МЕТОДИКИ ТИМ-2D

А. В. Шурыгин, А. А. Воропинов, А. К. Шмелева

ФГУП «РФЯЦ-ВНИИЭФ», г. Саров Нижегородской обл.

Методика ТИМ-2D [1] предназначена для решения задач механики сплошной среды в многообластной постановке на нерегулярных лагранжевых сетках. При этом важным этапом является построение начальной сетки. Одним из используемых в методике ТИМ-2D типов сеток является сетка, состоящая из ячеек Дирихле, которые в общем виде могут быть произвольными выпуклыми многоугольниками.

Построение сетки Дирихле, на основе триангуляции Делоне сводится к генерации набора узлов и построению по нему соответствующей триангуляции Делоне, а затем, формированию сетки из ячеек Дирихле на основе построенной триангуляции.

1. Инкрементный алгоритм построения триангуляции Делоне

Все известные алгоритмы для построения триангуляции Делоне можно разделить на два класса:

1. Инкременционные. Они выполняют триангуляцию, начиная с любой внутренней точки или ребра границы триангуляции, пошагово добавляя полученные новые треугольники по одному.

2. Рекурсивные. Эти алгоритмы рекурсивно разбивают заданное множество точек на поднаборы примерно одинакового размера с последующим слиянием полученных данных в необходимый результат.

Каждый класс содержит достаточно большое количество алгоритмов, описание которых можно найти в работе [2].

Рассмотрим более подробно инкрементный алгоритм, наиболее распространенный и достаточно удобный в программной реализации, описанный в работе [3]. Вначале текущая триангуляция состоит из единственного ребра границы области, по окончанию работы алгоритма текущая триангуляция становится триангуляцией Делоне. На каждом шаге алгоритм ищет новый треугольник, который подключается к границе триангуляции.

Определение границы зависит от следующей схемы классификации ребер относительно текущей триангуляции. Каждое ребро может быть *спящим*, *живым* или *мертвым*.

Спящие ребра – это ребра триангуляции Делоне не обнаруженные алгоритмом.

Живые ребра – это ребра, обнаруженные алгоритмом, однако, известна всего одна примыкающая к нему область.

Мертвые ребра – это ребра, обнаруженные алгоритмом, для которых известны обе примыкающие стороны.

Вначале живым является единственное ребро, принадлежащее границе области. С одной стороны к нему примыкает неограниченная область, вследствие чего одна область известна. Все остальные ребра считаются спящими. По мере работы алгоритма ребра из спящих становятся живыми, а затем мертвыми. Граница триангуляции на каждом этапе состоит из набора живых ребер.

На каждой итерации выбирается любое ребро e границы, и оно подвергается обработке, заключающейся в поиске неизвестной области, которой принадлежит это ребро. Если эта область окажется треугольником t, определяемым концевыми точками ребра e и некоторой третьей вершиной v то ребро e становиться мертвым, так как теперь известны обе примыкающие к нему области. Каждое из двух других ребер треугольника t переводятся в следующее состояние: из спящего в живое или из живого в мертвое. Здесь вершина v будет называться сопряженной с ребром e. В противном случае, если неизвестная область оказывается бесконечной плоскостью, то ребро e просто умирает. В этом случае оно не имеет сопряженной вершины. При этом направление для каждого ребра выбирается таким образом, что неизвестная область для него лежит справа от ребра. Процесс триангуляции Делоне показан на рис. 1, где действие происходит сверху вниз и слева направо. Граница на каждом этапе выделена жирной линией.

Для того чтобы построить первое ребро сначала необходимо найти крайнюю левую точку из всего набора точек. Далее вычисляем угловой коэффициент прямой, которая соединяет первую точку с очередной из всего набора точек по следующей формуле: $K = \dfrac{Y_i - Y_1}{X_i - X_1}$. Второй точкой для ребра будет, та для которой значение углового коэффициента прямой будет максимальным.

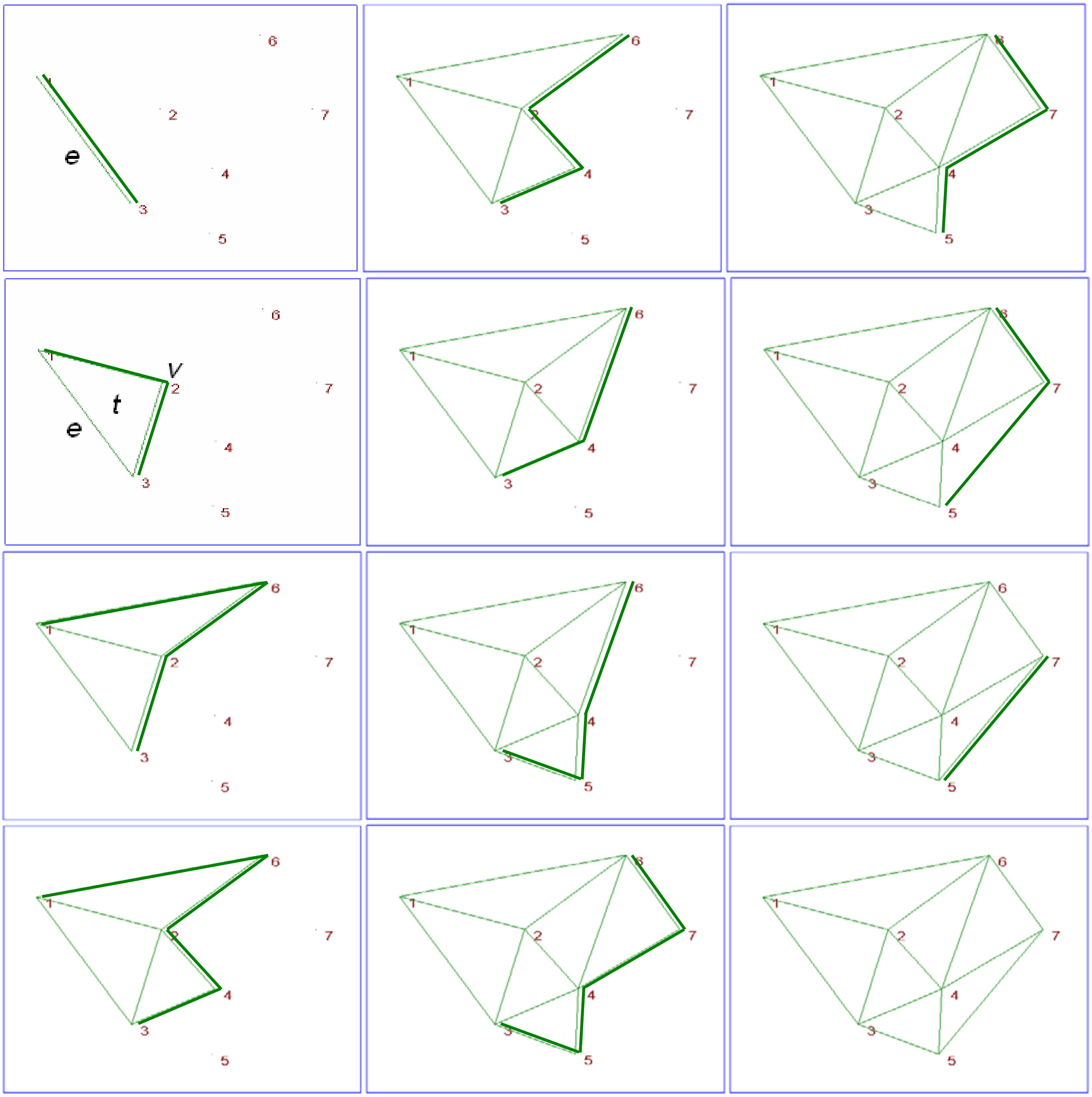


Рис. 1. Процесс построения триангуляции Делоне

В цикле по числу живых ребер происходит поиск сопряженной точки для текущего ребра. Цикл работает до тех пор, пока не останется ни одного живого ребра. Процедура поиска сопряженной точки для ребра выполняется следующим образом:

Сначала проверяется местоположение новой точки относительно текущего ребра:

$$Ax = X_2 - X_1,\ Ay = Y_2 - Y_1,\ Bx = X_i - X_1,\ By = Y_i - Y_1,$$
$$Sign = AxBy - BxAy,$$

где X_1, Y_1, X_2, Y_2 – координаты узлов для текущего ребра, а X_i, Y_i – координаты новой точки.

Если Sign > 0, то точка пропускается и далее не обрабатывается. Если Sign <= 0, то находим центр окружности, описанной вокруг этой точки и точек текущего ребра. Для этого необходимо, найти точку пересечения двух серединных перпендикуляров, восстановленных к текущему ребру и ребру, соединяющему один из концов текущего ребра с текущей точкой. Далее вычисляем расстояние t от центра текущего ребра до центра окружности. Наименьшее значение получаемого параметра t (рис. 2) при рассмотрении всех расположенных справа от ребра точек (и внутренних, и граничных) указывает на необходимую точку. Ей в данном случае является точка D.

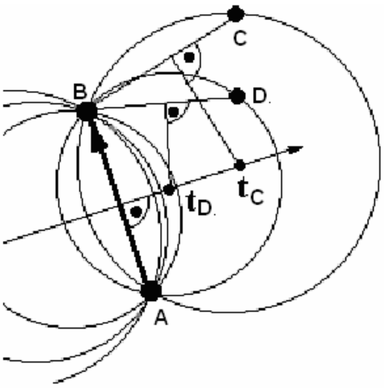


Рис. 2. К определению лучшей точки для триангуляции Делоне $(t_D < t_C)$

Если сопряженной точки для ребра не находим, то помечаем текущее ребро как мертвое и продолжаем цикл. Если находим сопряженную точку для ребра, то так же помечаем текущее ребро как мертвое, но при этом формируем два новых живых ребра и новый треугольник. При этом ориентация для нового ребра вводится следующим образом:

– для ребра, соединяющего начало ставшего на данной итерации мертвым ребра и полученной точкой – ориентация от начала к точке;

– для ребра, соединяющего точку с концом ставшего на данной итерации мертвым ребра – ориентация от точки к концу.

Для ребра записываем номера и координаты узлов, из которых оно состоит и номера соседних треугольников. Для треугольников записываем номера и координаты узлов, координаты центров описанных вокруг треугольников окружностей и номера ребер. Эта информация будет необходима при построении сетки Дирихле.

На рис. 3 показан пример построения триангуляции Делоне для 2500 точек, выбранных случайным образом в прямоугольной области. В результате построения образовано 4968 треугольников.

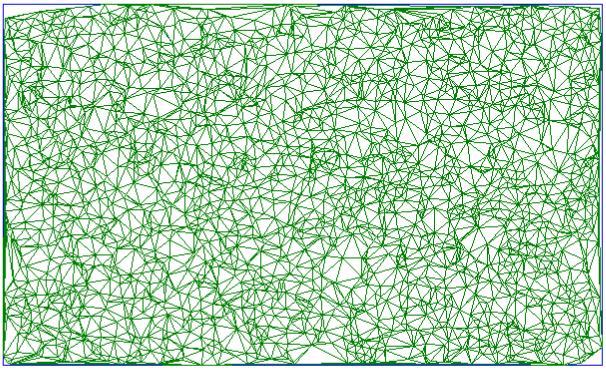


Рис. 3. Пример триангуляции Делоне для 2500 точек, выбранных случайным образом

2. Построение сетки Дирихле на основе триангуляции Делоне

Опишем алгоритм построения сетки Дирихле на основе триангуляции Делоне [4].

Шаг 1. По исходному множеству точек строим триангуляцию Делоне (см. рис. 4).

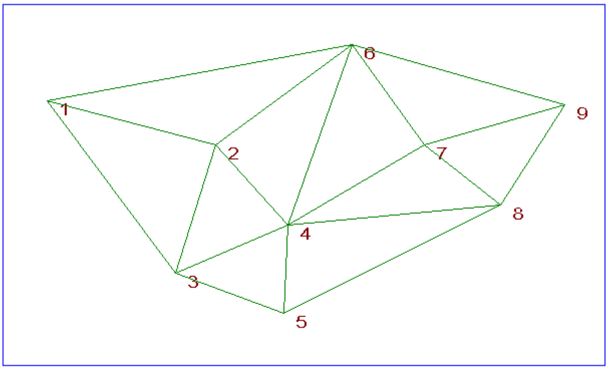


Рис. 4. Триангуляция Делоне, построенная по исходному множеству точек

Шаг 2. Для каждого треугольника триангуляции вычисляем координаты центра описанной вокруг него окружности.

Шаг. 3. Для каждого узла вычисляем центр ячейки Дирихле. Для этого совершаем обход вокруг текущего узла по смежным треугольникам и собираем центры их описанных окружностей. Если узел расположен не на границе триангуляции (узел с номером 4 рис. 5), то таким образом мы получим все координаты соответствующей ячейки Дирихле этого узла (см. рис. 5).

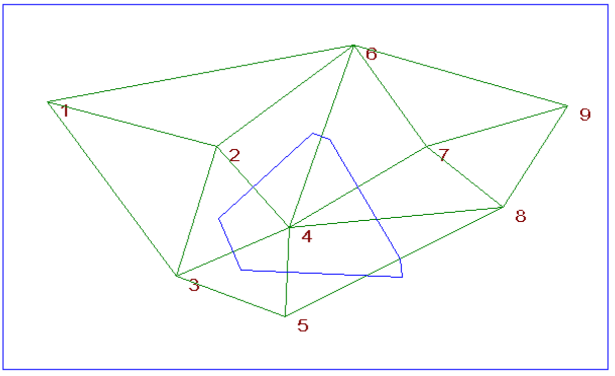


Рис. 5. Построение ячейки Дирихле для внутренних узлов триангуляции

Если узел находится на границе (узел с номером 3 рис. 5), значит, ячейка Дирихле является бесконечной фигурой. Поэтому необходимо в этом случае произвести отсечение двух ее бесконечных сторон границами области (см. рис. 6).

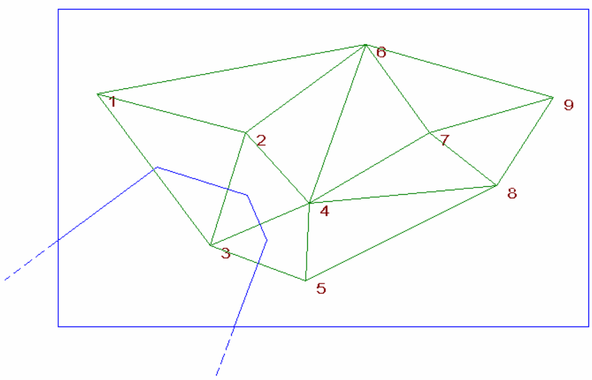


Рис. 6. Отсечение бесконечных сторон ячейки Дирихле границами области

В результате работы данного алгоритма получаем область, заполненную ячейками Дирихле (ячейки отрисованы синим цветом рис. 7).

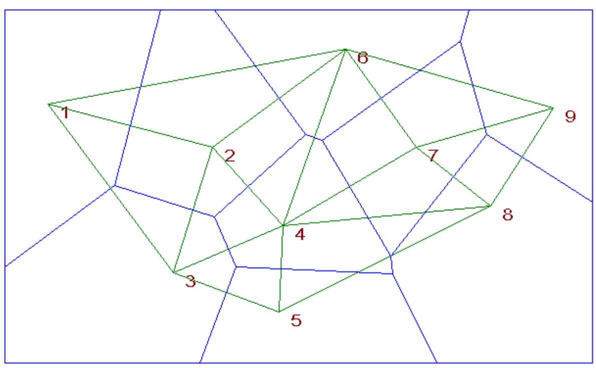


Рис. 7. Область, заполненная ячейками Дирихле

При работе с сеткой было принято использовать информацию об окружении центров окружностей, описанных вокруг треугольников, так как эти центры являются внутренними узлами сетки Дирихле. Причем в каждом узле (кроме угловых) этой сетки сходится ровно три ребра. Для каждого узла список окружающих ячеек и список соседних узлов формируется в порядке обхода против часовой стрелки.

При формировании списка соседних ячеек для узла используем номера вершин текущего треугольника. При формировании списка соседних узлов для узла используем номера соседних треугольников, если соседа не существует, то берем номер узла, получившегося в результате отсечения бесконечного ребра ячейки границей области. Это происходит следующим образом:

– сначала находим уравнение прямой, которая является серединным перпендикуляром к ребру, не имеющему соседнего треугольника;

– далее находим координаты точек пересечения этой прямой со сторонами области (см. рис. 8) и выбираем ту точку (P_1), которая расположена по другую сторону от вершины треугольника (вершина с номером 2), не принадлежащей ребру;

– заносим информацию о соседстве для нового узла;

– дополняем информацию о соседних узлах для текущего узла.

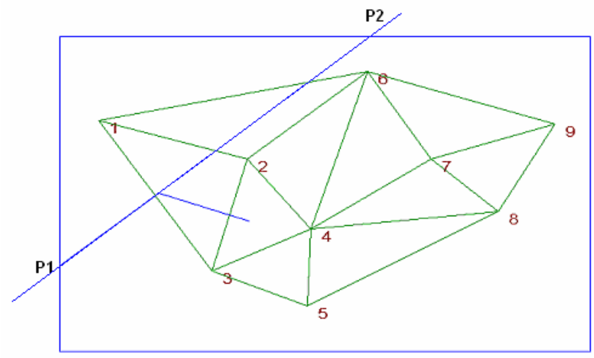


Рис. 8. Пересечение прямой со сторонами области

Процедуру отсечения ребра ячейки границей области так же выполняем, если сосед для треугольника существует, но центр описанной вокруг него окружности расположен вне области.

Далее в цикле по всем граничным узлам происходит дополнение информации о соседних узлах:

– сначала анализируется, какой из сторон области принадлежит текущий граничный узел;

– выполняется поиск ближайших узлов с обеих сторон по отношению к текущему, номера которых дополняют списки соседних узлов.

После того как списки соседства для всех узлов сформированы, выполняется уплотнение нумерации узлов, если некоторые их них не попали в область.

3. Алгоритмы формирования координат набора точек

В алгоритмах построения сетки Дирихле на основе триангуляции Делоне реализовано несколько методов формирования координат набора точек. Ниже представлены примеры формирования координат точек для прямоугольных областей, так как в методике ТИМ-2D один из способов построения сетки это обрамление любой геометрии прямоугольником, построение сетки в этом прямоугольнике, а затем из сеточного покрытия, путем разрезания ячеек вдоль границы, строится вписанная в границу подобласти сетка.

Первый – это возможность загружать координаты точек из файла. Благодаря этому в программу построения сетки Дирихле могут быть загружены координаты, сформированные отдельной подпрограммой генерации точек, либо введенные вручную пользователем. Пример такой сетки показан на рис. 10, где сначала выполнено построение триангуляции Делоне по заданному набору точек (см. рис. 10а), а на ее основе выполнено построение сетки Дирихле (см. рис. 10б).

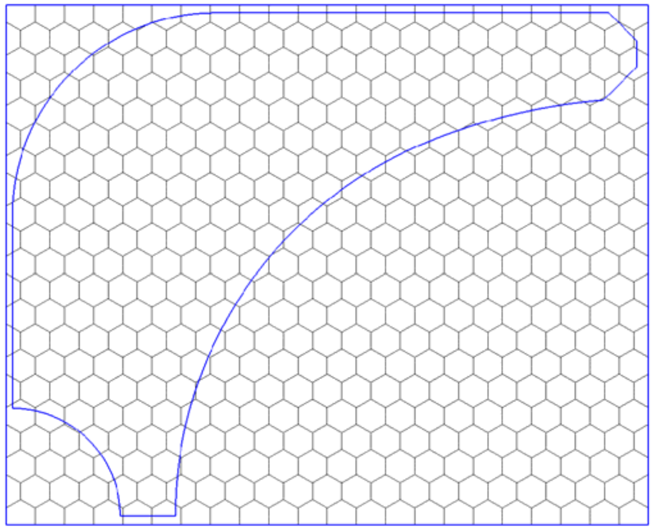
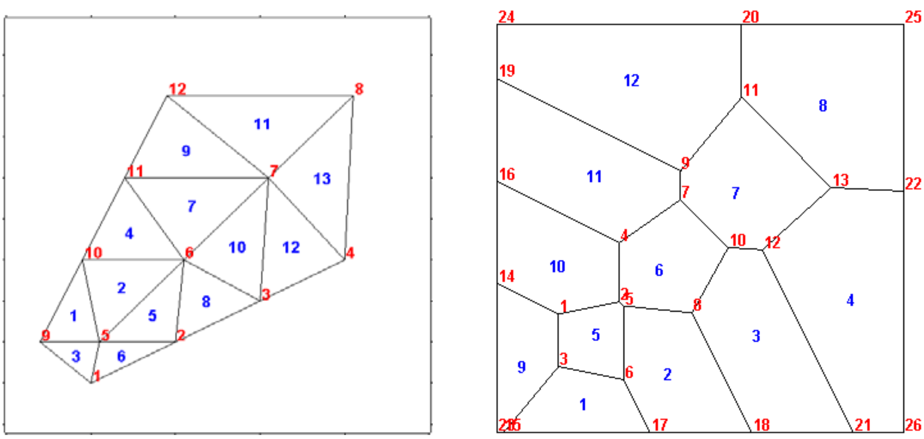


Рис. 9. Один из способов построения сетки в методике ТИМ-2D



а б

Рис. 10. Пример триангуляции Делоне (а), выполненной по заданному из файла набору точек и построенной на ее основе сетки Дирихле (б)

Введем следующие обозначения, которые далее будут применяться:

$X_{\max}$, $Y_{\max}$, $X_{\min}$, $Y_{\min}$ – максимальные и минимальные координаты границы области;

D_X – расстояние между центрами соседних ячеек по оси X;

D_Y – расстояние между центрами соседних ячеек по оси Y;

K_X, K_Y – число ячеек вдоль оси абсцисс и оси ординат;

K_C – количество точек участвующих в построении триангуляции Делоне;

X_C, Y_C – координаты этих точек.

Второй способ – это построение сетки с расстановкой точек случайным образом (см. рис. 11). Здесь с помощью функции *SetRange* значение координат точки из диапазона случайных чисел от 0 до 1, преобразовывается к диапазону возможных значений, в зависимости от размеров области. Это преобразование выполняется по следующим формулам:

$$X_C = X_{\min} + X_C\left(X_{\max} - X_{\min}\right),$$

$$Y_C = Y_{\min} + Y_C\left(Y_{\max} - Y_{\min}\right).$$

Третий способ – расстановка точек в шахматном порядке, таким образом, в результате чего получается сетка из шестиугольников. На рис. 12(б) показан пример построения такой сетки.

Координаты центров ячеек вычисляются следующим образом:

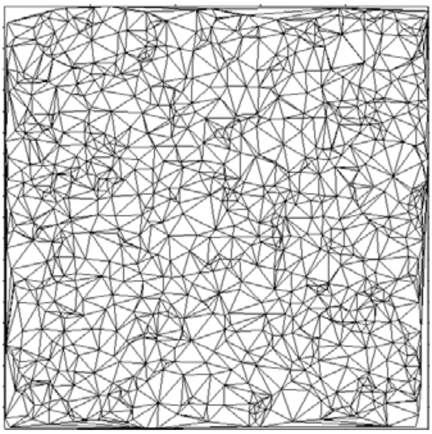
$$D_X = \frac{X_{\max} - X_{\min}}{K_X - 1}, \quad D_Y = \frac{X_{\max} - X_{\min}}{K_Y - 1};$$

$$X_C = X_{\min} + \frac{D_X}{2}K + (i-1)D_X,$$

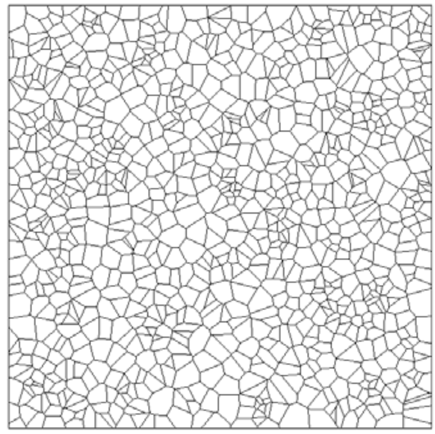
$i = 1,\ldots,K_X$, $K = 1$, если i – нечетное число и $K = 0$, если i – четное.

$$Y_C = Y_{\min} + (j-1)D_Y,$$

где $j = 1,\ldots,K_Y$

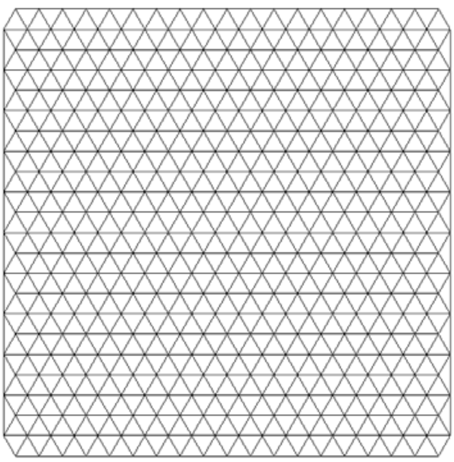


а

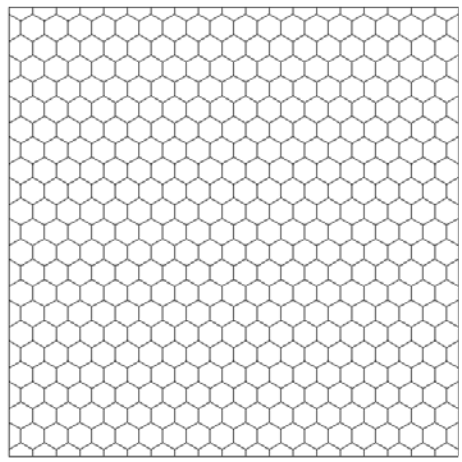


б

Рис. 11. Пример триангуляции Делоне (а), выполненной по набору точек, сгенерированных случайным образом, и построение на ее основе сетки Дирихле(б)



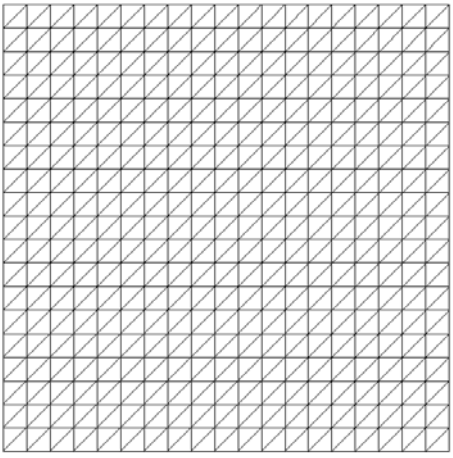
а



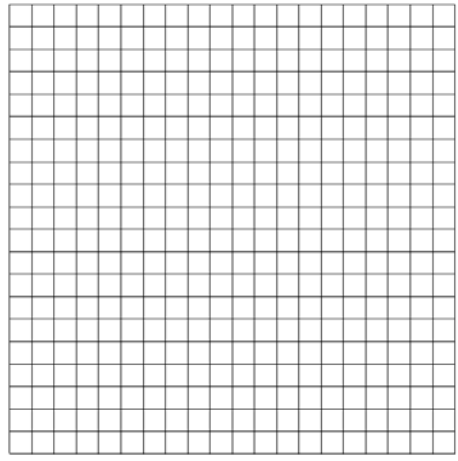
б

Рис. 12. Пример триангуляции Делоне, выполненной по набору точек (а), заданных в шахматном порядке, и построение на ее основе сетки Дирихле из шестиугольников (б)

Четвертый способ – равномерное покрытие области точками с определенным шагом по *X* и по *Y*. В результате получается четырехугольная сетка (см. рис. 13(б)).



а



б

Рис. 13. Пример триангуляции Делоне (а), выполненной по набору точек, равномерно распределенных по области, и построение на ее основе четырехугольной сетки Дирихле (б)

Координаты центров ячеек вычисляются следующим образом:

$$D_X = \frac{X_{\max} - X_{\min}}{K_X}, \quad D_Y = \frac{X\max - X_{\min}}{K_Y},$$

$$X_C = X_{\min} + iD_X - \frac{D_X}{2}, \quad Y_C = Y_{\min} + jD_Y - \frac{D_Y}{2},$$

где $i = 1,\ldots, K_X$, $j = 1,..., K_Y$.

4. Оценка производительности

Ниже представлены результаты теста на оценку производительности стандартного алгоритма построения сетки Дирихле [6] и алгоритма построения сетки Дирихле на основе триангуляции Делоне. Сравнительный график быстродействия вышеупомянутых алгоритмов представлен на рис. 14.

В качестве алгоритма расстановки центров ячеек для данного теста был выбран алгоритм расстановки центров для получения правильных шестиугольных ячеек (см. рис. 11).

Из графика видно, что производительность алгоритма построения сетки Дирихле на основе триангуляции Делоне много ниже производительности стандартного алгоритма. Так построение двумерной нерегулярной сетки Дирихле на основе триангуляции Делоне на 1000000 ячеек на персональном компьютере с частотой процессора порядка 2,4 ГГц и оперативной памятью 4 Гб заняло около 580 минут, это очень дорого для расчета начальных данных. В связи с этим возникла необходимость оптимизировать данный алгоритм.

Самой дорогой процедурой в данном алгоритме является поиск сопряженной точки для текущего ребра или ближайшего соседа Делоне для текущего треугольника. Ее трудоемкость составляет $O(N^2)$. Для ускорения данной процедуры было предложено использовать клеточный алгоритм [5] поиска сопряженной точки.

Сравнение производительности построения сетки Дирихле на основе триангуляции Делоне с использованием клеточного алгоритма поиска сопряженной точки для ребра со стандартным алгоритмом построения сетки Дирихле показано на рис. 15.

Из графика видно, что трудоемкость клеточного алгоритма является практически линейной O(N) и расчет сетки на 1000000 ячеек занимает всего 3 минуты, в то время как стандартный алгоритм построения сетки Дирихле справляется с этой операцией за 100 минут.

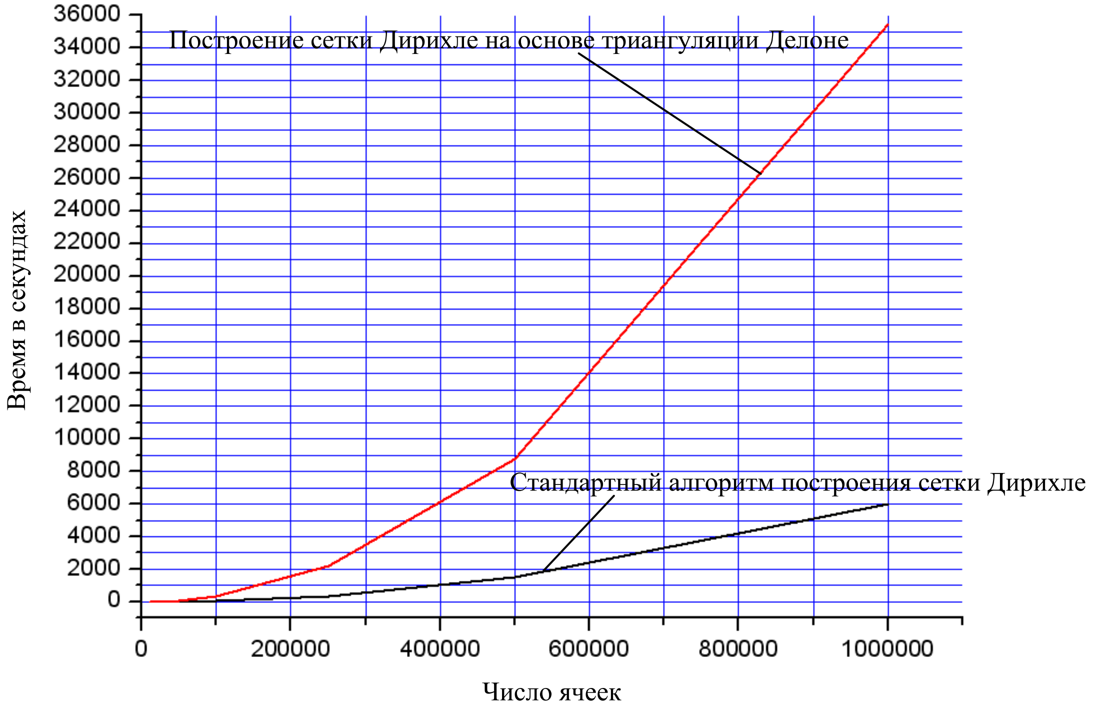


Рис. 14. Сравнение производительности

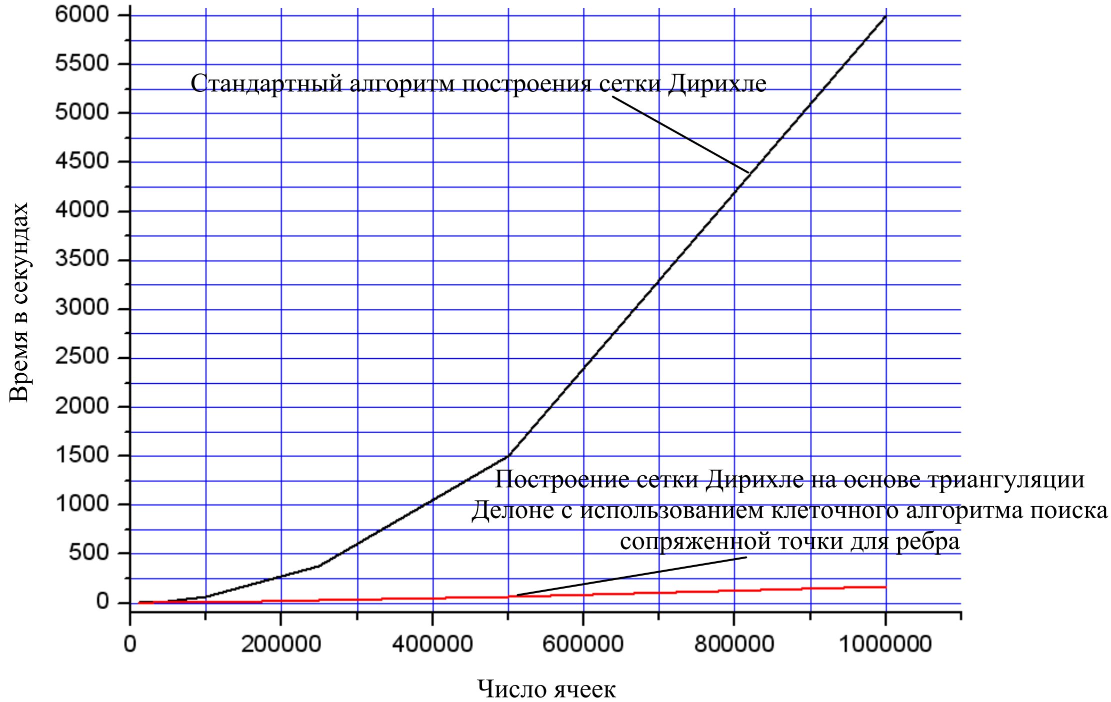


Рис. 15. Сравнение производительности

Заключение

В результате проделанной работы в программу расчета начальных методики ТИМ-2D внедрен метод построения сетки Дирихле на основе триангуляции Делоне. Проведена сравнительная оценка быстродействия стандартного метода построения сетки Дирихле и метода построения сетки Дирихле на основе триангуляции Делоне. В рамках данного метода был реализован клеточный алгоритм поиска ближайшего соседа Делоне для текущего треугольника. В результате процедура поиска стала работать быстрее в 190 раз. Благодаря этому общее ускорение метода построения сетки Дирихле на основе триангуляции Делоне составило более 30 раз по сравнению со стандартным методом построения сетки Дирихле.

Реализованные алгоритмы построения сетки Дирихле на основе триангуляции Делоне планируется использовать при построении начальной сетки, а так же в качестве одного из инструментов для локальных перестроек сетки в процессе счета.

Литература

1. Соколов С. С., Воропинов А. А., Новиков И. Г. и др. Методика ТИМ-2D для расчета задач механики сплошной среды на нерегулярных многоугольных сетках с произвольным количеством связей в узлах // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2006. Вып. 4. С. 29–43.

2. Скворцов А. В. Триангуляция Делоне и ее применение. Томск: Издательство Томского университета, 2002.

3. Ласло М. Вычислительная геометрия и компьютерная графика на С++: Пер. с англ. М.: «Издательство БИНОМ», 1997.

4. Скворцов А. В., Костюк Ю. Л. Применение триангуляции для решения задач вычислительной геометрии // Геоинформатика: Теория и практика. Вып. 1. Томск: Изд-во Томск. ун-та, 1998. С. 127–138.

5. Препарата Ф., Шеймос М. Вычислительная геометрия: Введение / Пер. с англ. М.: Мир, 1989.

6. Соловьев А. В., Соловьева Е. В., Тишкин В. Ф., и др. Об одном алгоритме построения ячеек Дирихле. М., 1985.