

# МОДИФИКАЦИЯ АЛГОРИТМА БОУЭРА – УОТСОНА ГЕНЕРАЦИИ ДВУМЕРНЫХ НЕСТРУКТУРИРОВАННЫХ СЕТОК В ОБЛАСТЯХ ПРОИЗВОЛЬНОЙ ФОРМЫ

М. Л. Сидоров

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

## Введение

Генерация пространственных сеток является одной из основных задач при решении уравнений математической физики численными сеточными методами.

В работе предложен алгоритм построения двумерной нерегулярной сетки в области произвольной формы. В основе алгоритма лежит удовлетворение двух требований: итоговая сетка должна иметь как можно меньшую меру геометрической и топологической нерегулярности. Первое требование удовлетворяется с помощью модификации алгоритма Боуэра – Уотсона (Bowyer, Watson) [1, 2], основанном на улучшении грубой триангуляции области. Второе требование удовлетворяется с помощью фронтального подхода и комплекса мер, направленных на приведение многоугольников фронта к шестиугольному виду. Алгоритм триангуляции области использует функцию распределения характерного размера ячеек, являющуюся численным решением методом конечных элементов уравнения стационарной теплопроводности с заданной на границе области и во внутренних источниках температурой. Здесь под температурой понимаются длины граничных ребер, полученные после разбиения границы, заданного пользователем.

## Алгоритм генерации нерегулярной сетки

Описываемый алгоритм, основан на двух принципах. Первый, – это минимизация меры геометрической нерегулярности

$$MG = \frac{1}{N_{\Delta}} \sum_{\Delta_i}^{N_{\Delta}} \left( \frac{1}{2} - \frac{r_{\Delta_i}}{R_{\Delta_i}} \right),$$

где  $r$  – радиус вписанной в треугольник  $I$  окружности;  $R$  – радиус описанной около треугольника  $I$  окружности, суммирование ведется по всем треугольникам.

Второй, – это минимизация меры топологической нерегулярности

$$MT = \frac{\sum_{I \in \text{InternalCells}} |6 - \text{Valency}(I)|}{\text{NumberOfInternalCells}},$$

где под валентностью ( $\text{Valency}$ ) понимаем число треугольников вокруг точки, суммирование ведется по всем внутренним точкам. Для правильной треугольной сетки  $MG = 0$  и  $MT = 0$ .

В общем случае этот генератор должен уметь создавать анизотропные сетки, но на данный момент реализован случай изотропных сеток, допускающих изменение характерного размера ячеек.

В головной программе используются две основных процедуры – процедура **MININGEOME** и процедура **BLASTOME**. Входными данными являются два одномерных массива координат, определяющих контур. На выходе из процедуры выдается триангуляция области в треугольном формате (треугольник  $I$  ссылается на три точки  $P1, P2, P3$ , и три своих соседа  $M1, M2, M3$  против часовой стрелки от каждой из точек, рис. 1).

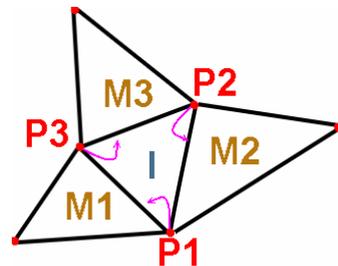


Рис. 1. Треугольный формат хранения топологии сетки

Реализованный генератор состоит из двух основных программ – стандартных этапов при генерации сетки. Первая программа – программа триангуляции области, ограниченной заданным контуром. При ее создании автор руководствовался тем, что программа должна быть проста в организации и реализации, без сложных структур и критериев. В результате анализа алгоритмов, описанных в различных источниках, был выдвинут проект генератора с неупорядоченным фронтом, работающим на принципе уменьшения меры геометрической нерегулярности исходной триангуляции области.

Вторая программа, программа уменьшения меры топологической нерегулярности, использует различные операции с сеткой, описанные ниже, для уменьшения числа не шестиугольников, и увеличения числа шестиугольников. Здесь и далее под многоугольником понимается медианный конечный объем – фигура, полученная соединением центром треугольников. Количество углов многоугольника для внутренней точки равно валентности данной точки.

Кратко опишем весь процесс получения сетки на примере одной области.

Дискретный контур области получается из непрерывного с помощью программ разбиения кривых. После того, как контур получен, производится начальная триангуляция области, ограниченной данным контуром. Мы не будем останавливаться на описании алгоритма триангуляции контура, скажем лишь, что для первых двух этапов подходит любая невырожденная сколь угодно плохая триангуляция контура. Пример триангуляции представлен на рис. 2.

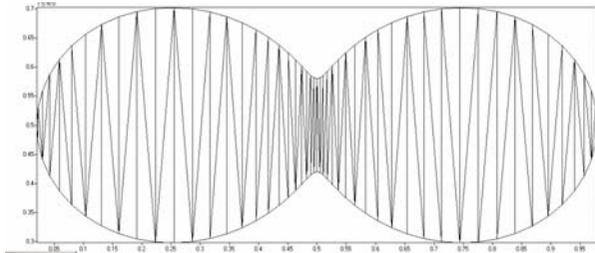


Рис. 2. Начальная триангуляция области, ограниченной контуром

Для построения сетки необходима функция распределения характерных размеров длин ребер ячеек –  $D(x, y)$ . С целью получения качественного результата желательно иметь гладкое распределение характерного размера на контуре (то есть гладкие граничные условия). В описываемом алгоритме функция распределения находится численно перед этапом построения сетки, но отличным от описанных выше способов. Данная функция в описываемом ниже алгоритме является решением уравнения стационарной теплопроводности с заданной на границе температурой  $(\Delta D = 0, D(x, y)|_{(x,y) \in \Gamma} = \ell(x, y))$ , характерный размер в точке граничного контура равен полусумме

длин граничных ребер, сходящихся в точку. После начальной триангуляции (рис. 2) области, ограниченной заданным контуром, производится анализ треугольников на наличие градиента характерного размера. Если градиент существенно отличен от нуля, то для улучшения аппроксимации в центр треугольника добавляется точка. После прохода по всем старым (то есть без новых треугольников, образованных при добавлении точки) треугольникам и перестройки полученных треугольников, мы получаем первую сетку (рис. 3) для нахождения функции  $D(x, y)$  с помощью метода конечных элементов. Полученную в результате систему линейных уравнений решаем в последовательном режиме с помощью библиотеки PMLP [2].

После нахождения искомой функции мы повторяем ту же последовательность действий: проход по треугольникам, добавление точек, перестройка, решение уравнения стационарной теплопроводности. Число итераций ограничено двумя, чтобы сетка для определения функции распределения была не слишком подробной в зоне сильных градиентов. Итерации были введены, чтобы улучшить аппроксимацию функции распределения в случае ее нелинейного характера. Найдя численное представление функции распределения, запоминаем сетку, на которой она была найдена, как фоновую и восстанавливаем первоначальную триангуляцию области. Найденное численное решение уравнение Лапласа с граничными условиями 1 рода для данной задачи представлено на рис. 4.

После восстановления первоначальной триангуляции запускаем итерационный процесс построения сетки. Опишем основные детали алгоритма.

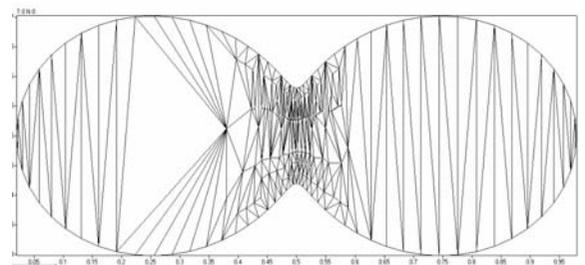
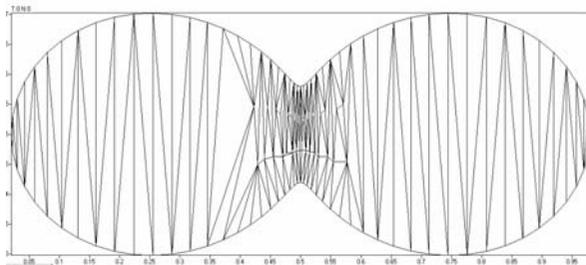


Рис. 3. Фоновая сетка для нахождения функции распределения. После 1-й и 2-й итерации

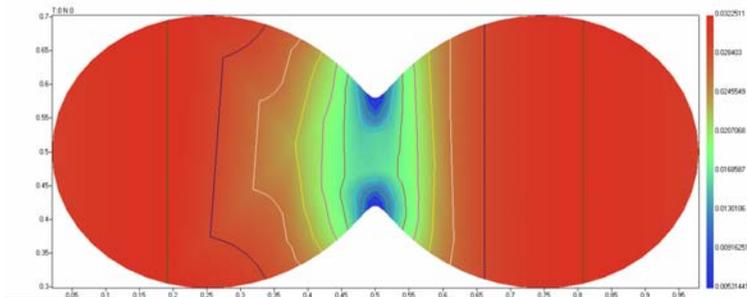


Рис. 4. Поле функции распределения

Набирается множество существующих треугольников, назовем его **O**. Из множества **O** выбираются треугольники, обладающие мерой искажения  $M < 0,9$

( $M$  вычисляется по формуле  $M = 4\sqrt{3} \frac{S_{\Delta}}{l_1^2 + l_2^2 + l_3^2}$ )

и удовлетворяющие условию: число сторон треугольника с длинами меньше 1.2 виртуальной длины не равно 3. Под виртуальной длиной ребра понимаем среднее арифметическое из двух характерных размеров  $D(i), D(j)$ , вычисленных на фоновой сетке. Далее выбирается ребро фронта из условия  $0,8 < L/D < 1,2$ , где  $L(i, j)$  – настоящая длина ребра. После этого от середины выбранного ребра **I, J** строится нормаль с длиной, равной минимуму из виртуальной длины ребра и предполагаемым характерным размером  $D^*$  новой точки **K**. Две эти ситуации показаны на рис. 5.

Производится нахождение треугольника **T**, в который попала предполагаемая новая точка. Это делается итерационно: от текущего треугольника набираются соседи 1-го, 2-го и т. д. порядков, пока не будет найден содержащий точку треугольник. Производится проверка на близость точки к точкам найденного треугольника по критерию: расстояние от точки до вершин треугольника меньше 1.5 от характерного размера, вычисленного в точке. Если проверка пройдена, то точка добавляется, а треугольники с **B** уровнем соседства к найденному треугольнику **T** подвергаются коррекции (перестройке в случае

необходимости). Характерный размер в точке вычисляется с помощью весов от треугольника фоновой сетки, куда она попала (поиск треугольника реализован так же, как это описано выше). **B** зависит от номера итерации нелинейно и может изменяться от 20 до 3. Зависимость может быть изменена пользователем в файле настройки.

После того, как все допустимые треугольники из множества **O** выбраны, образуется множество **O1** – множество старых **O** треугольников и множество **H** добавленных треугольников. Часть треугольников из **O** и **H** содержится в множестве **B**. Если множество **H** пусто, процесс построения сетки заканчивается, иначе множество **O = B** и процесс продолжается. В процессе выполнения происходит уменьшение меры геометрической нерегулярности. Результат работы данного алгоритма представлен на рис. 6.

Полученная сетка далее подвергается коррекции во втором блоке программ, отвечающем за минимизацию топологической нерегулярности. Это блок фронтальной коррекции. Фронт продвигается от границы внутрь области (неограниченно или на число уровней заданных пользователем), и на точках фронта работают специальные процедуры, корректирующие валентность точек. Цель всей топологической коррекции – уменьшить число точек с валентностью, не равной 6, а также постараться сделать так, чтобы остались только 5, 6, 7 угольники (внутри области).

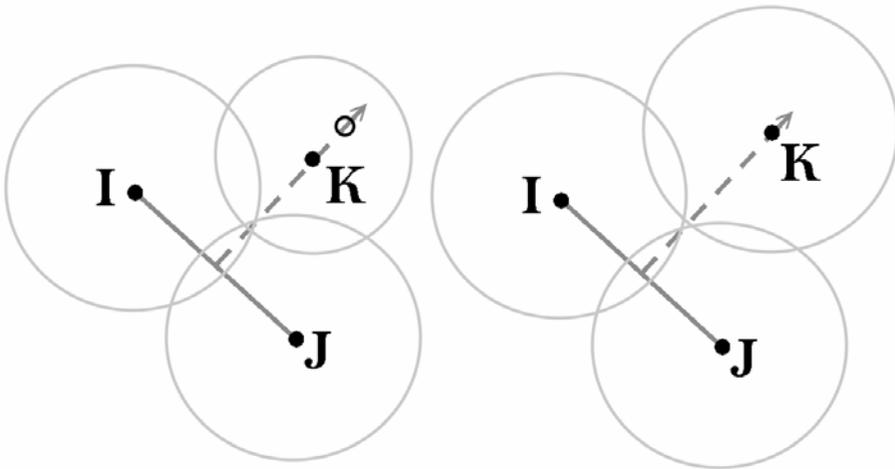


Рис. 5. Варианты построения новой точки **K** от ребра **I, J**

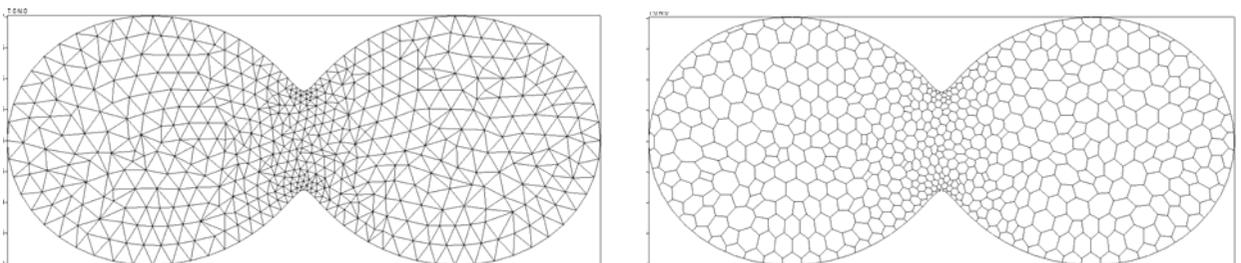


Рис. 6. Сетка после процесса уменьшения меры геометрической нерегулярности

В начале блока работает программа по коррекции граничных точек. После коррекции на границе области будет получен массив граничных точек (фронт). Затем производится коррекция фронта внутри области. В цикле по точкам фронта анализируется валентность точки и ее соседей по фронту. Если комбинация этих валентностей удовлетворяет одной из заложенной в алгоритм комбинации, то точка корректируется.

После окончания процедуры коррекции внутреннего фронта остаются ситуации, не исправленные в этой процедуре.

С целью улучшения топологической меры сетки на третьем этапе применяется набора процедур по обработке некоторых часто встречающихся случаев.

Ниже на рис. 7 представлена сетка до процесса топологической корректировки. На рис. 8 представлена сетка после топологической корректировки.

После топологической корректировки применяется сглаживание сетки с ограничением – не уменьшение минимума меры искажения треугольников вокруг точки.

В табл. 1 представлены данные для сравнения. Из данных в таблице видно, что второй и третий блок программ уменьшил число пятиугольников почти в 6 раз, а число семиугольников почти в 7 раз. Мера топологической нерегулярности уменьшилась почти в 50 раз, вместе с тем это положительно сказалось и на мере геометрической нерегулярности, она уменьшилась в 2 раза. Следует отметить, что такая процедура коррекции может приводить к появлению небольшой анизотропии в сетке.

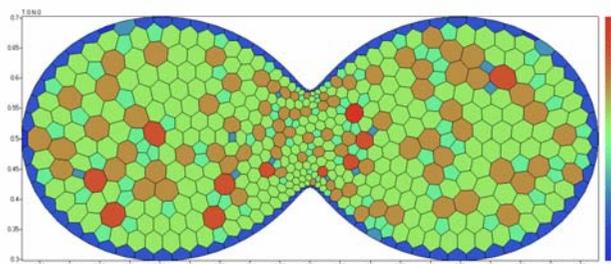


Рис. 7. Сетка до топологической корректировки

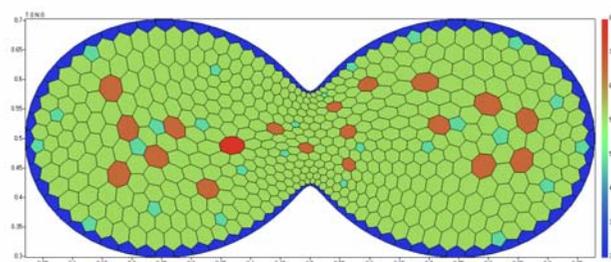


Рис. 8. Сетка до и после топологической корректировки

Таблица 1

Результаты топологической корректировки

Характеристика сетки	До коррекции	После коррекции
1-валентных	0	0
2-валентных	0	0
3-валентных	0	0
4-валентных	10	0
5-валентных	92	26
6-валентных	263	444
7-валентных	83	18
8-валентных	10	1
9-валентных	1	0
Общее число	457	489
Мера топологической нерегулярности	0,474	0,009
Мера геометрической нерегулярности	0,004	0,002
Процент пятиугольников	34,9	5,8
Процент семиугольников	31,5	4
Процент многоугольников, не являющихся шестиугольниками	4,57	0,2

После второго блока выполняется операция конвертирования базы из треугольного формата в реберный формат (ребро **I** ссылается на две точки **RV1**, **RV2**, и два ребра **RR1**, **RR2**, против часовой стрелки от каждой из точек, рис. 9), производится восполнение границы фиктивными ребрами, необходимыми для сшивки. Область считается обработанной после получения триангуляции в реберном формате.

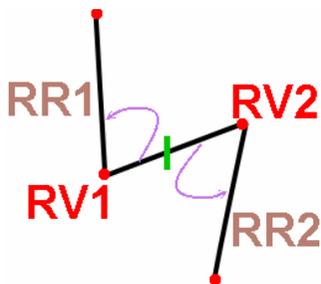


Рис. 9. Реберный формат хранения топологии сетки

В качестве дополнительной опции предусмотрено измельчение сетки и натяжение ее на контур с более подробной детализацией. Измельчение может производиться равномерно для всей сетки (сохраняя при этом отношение минимальной к максимальной длине ребра в сетке). Такое измельчение нужно

для получения сеток с большим числом точек и асимптотической топологической мерой нерегулярности  $O\left(\frac{1}{4^n}\right)$ , где  $n$  – число итераций измельчения (то есть мера стремится к нулю при измельчении сетки). Еще измельчение может производиться для достижения длинами ребер заданного пользователем размера. При этом возможны четыре ситуации: треугольник дроблению не подвергается, дробится одна, две или три стороны. Последние три варианта показаны на рис. 10.

Для оценки скорости программы MininGeoMe использовались две области (они будут показаны в примере 1 и примере 3 соответственно). Результаты представлены в табл. 2.

## Результаты применения алгоритмов

Описанные алгоритмы были реализованы в виде программ, которые тестировались на областях различной формы. Целью тестирования была проверка способности строить качественную сетку в невыпуклых областях с границей разбитой неравномерно и равномерно.

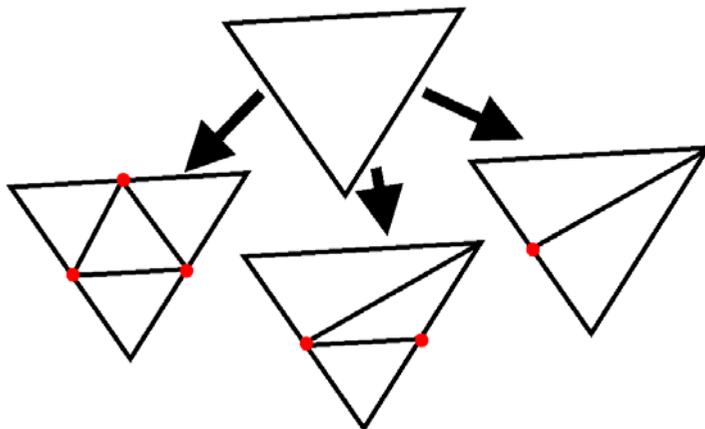


Рис. 10. Варианты дробления треугольника

Таблица 2

Времена построения сеток в зависимости от задачи

Число точек на границе	Разбиение границы	Количество треугольников	Количество точек	Время построения, с
Пример 1 – 3051	Равномерное	~ 154 000	~ 79 000	16
Пример 1 – 6107	Равномерное	~ 615 000	~ 310 000	55
Пример 1 – 6107*	Равномерное	~ 615 000	~ 310 000	27
Пример 2 – 3900	Не равномерное	~ 585 000	~ 294 000	83
Пример 2 – 5200	Не равномерное	~ 1 104 000	~ 554 000	161

Примечание. \* – это расчет без вычисления характерного размера точки в процессе построения, размер фиксировался один и тот же для всей задачи (так как разбиение границы равномерное).

## Пример 1

Область на рис. 11 представляет собой набор отрезков, разбитых неравномерно.

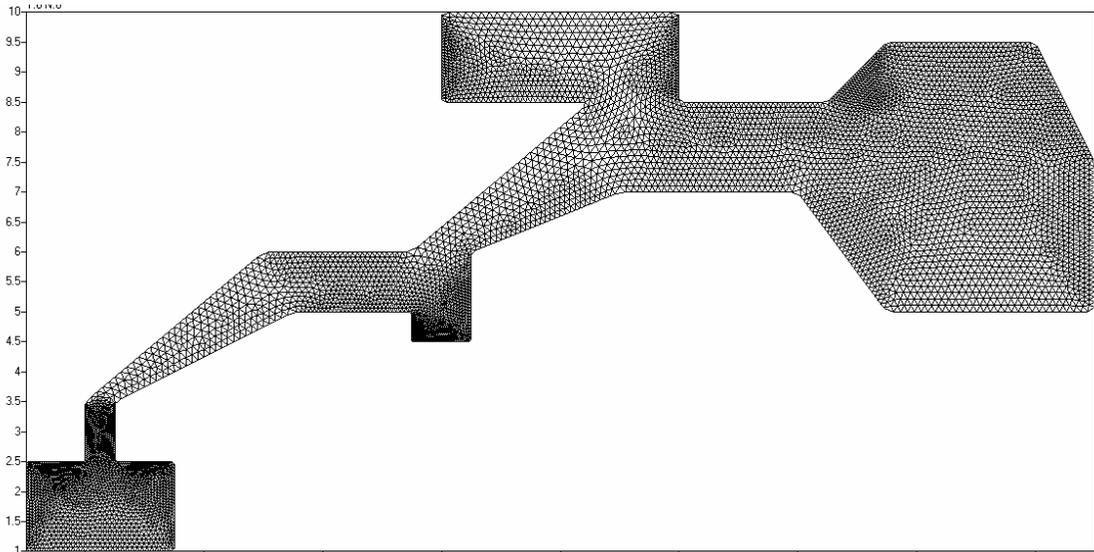


Рис. 11. Сетка из примера 1

На рис. 12 представлены фрагменты сетки.

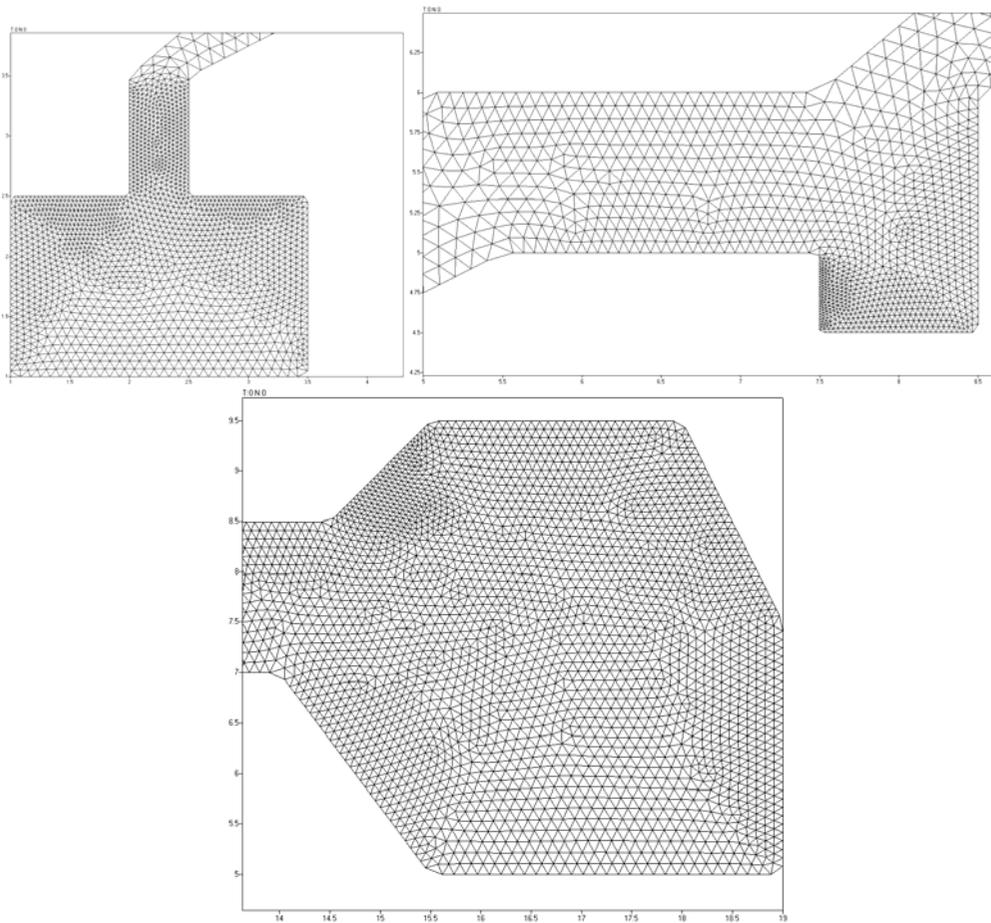


Рис. 12. Фрагменты сетки из примера 1

## Пример 2

Ниже на рис. 13 представлена сетка в области с границей разбитой неравномерно. На рисунке сетка без проведения фронтальной коррекции, на рисунке сетка с примененной фронтальной коррекцией.

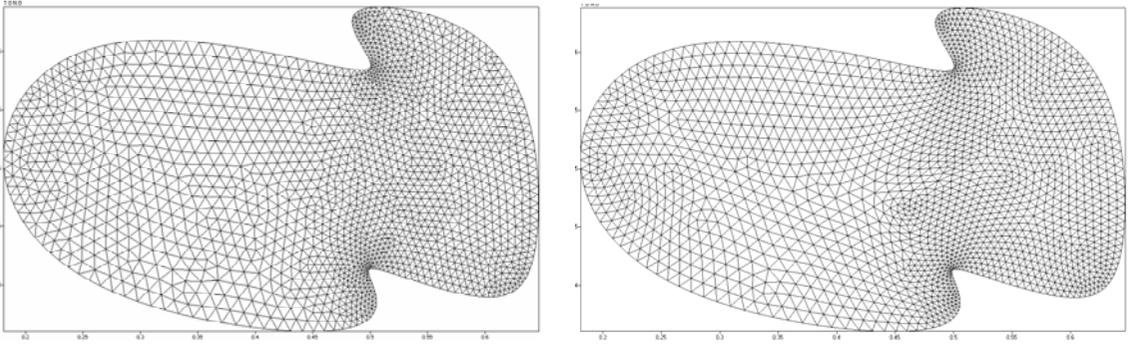


Рис. 13. Сетка из примера 2

## Пример 3

На рис. 14, 15 показана область с неравномерно разбитой границей.

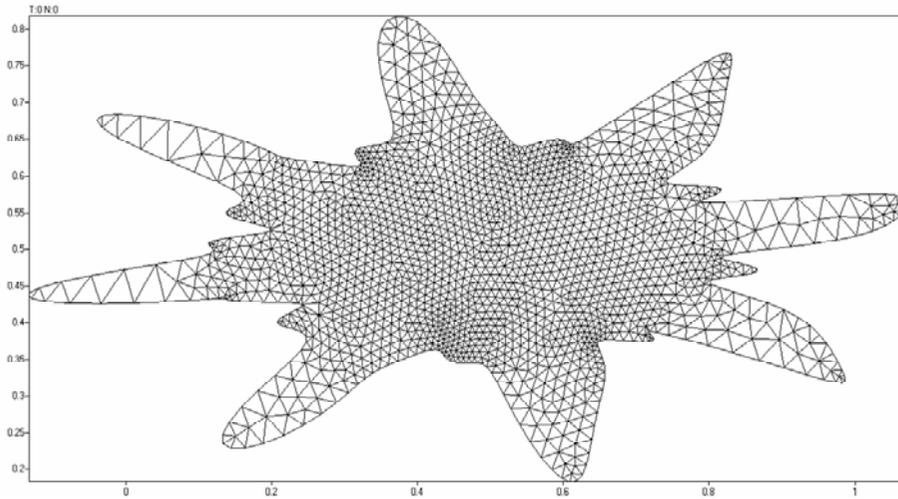


Рис. 14. Сетка из примера 3

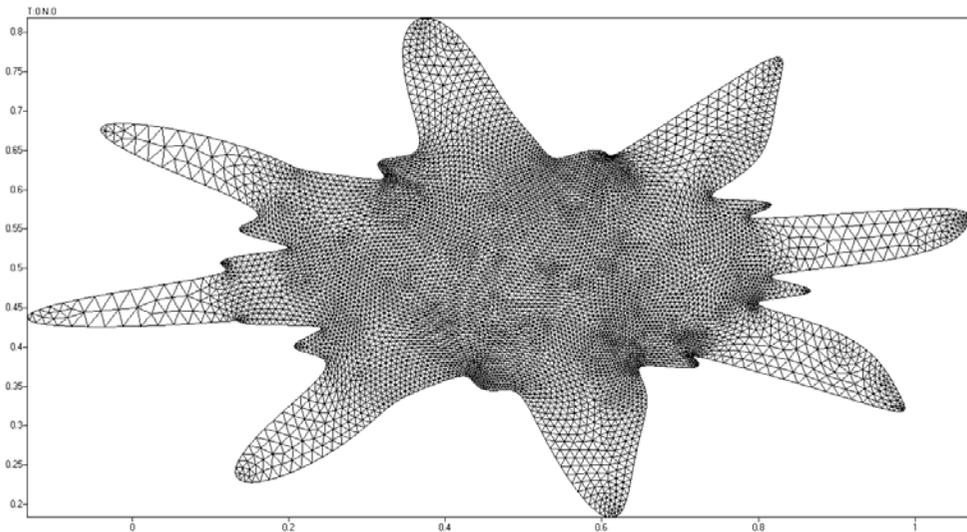


Рис. 15. Сетка из примера 3

## Пример 4

Рассмотрим фрагмент области с границей разбитой неравномерно на рисунке. Задача с сильным перепадом длин ребер является стандартным тестом при тестировании алгоритмов построения сеток. В левой части рис. 16 показано распределение характерного размера. В правой части рис. 16 показано распределение характерного размера. На рис. 17 показан фрагмент сетки в зоне сгущения сетки.

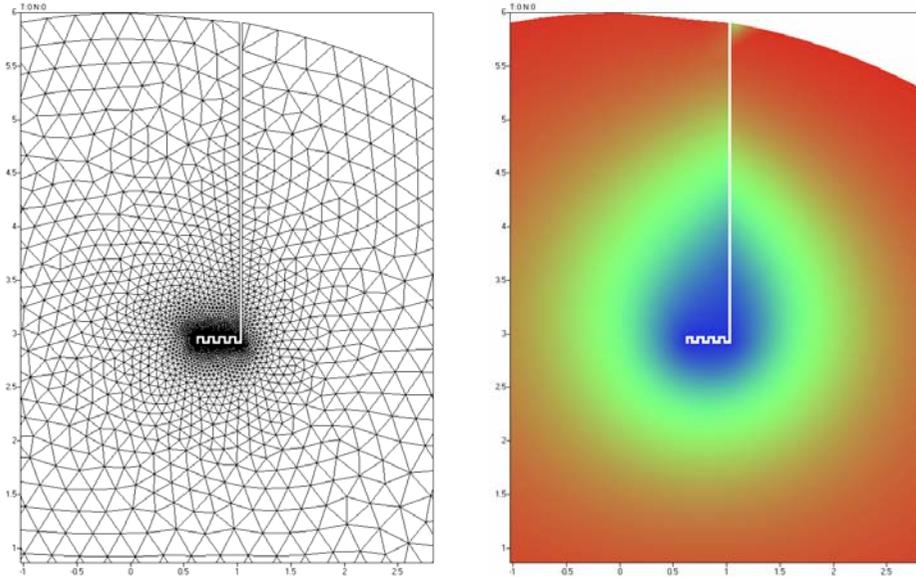


Рис. 16. Фрагмент сетки и поля распределения из примера 4

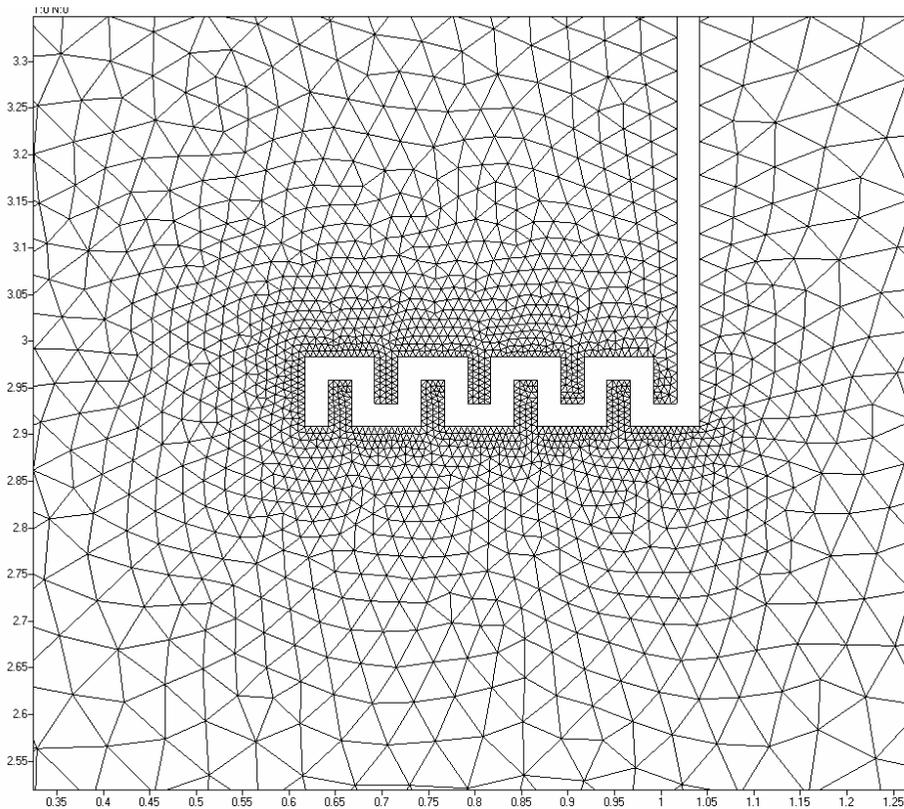


Рис. 17. Фрагмент сетки и поля распределения из примера 4

В статье предложена модификация алгоритма Боуэра – Уотсона (Bowyer, Watson), позволяющая строить качественную сетку в двумерных областях произвольной формы и произвольной связности. Алгоритм был опробован на множестве задач и показал свою работоспособность. В настоящее время программа, в которой реализован данный алгоритм, используется при генерации сетки для решения сложных задач газовой динамики, теплопроводности, подземной гидродинамики. В будущем планируется реализовать возможность учета анизотропии, и реализовать аналогичный алгоритм для трехмерного случая.

1. Bowyer A. Computing Dirichlet tessellations// The Computer Journal. 1981. Vol. 24, N 2. P. 162–166.
2. Watson D. Computing the n-dimensional Delaunay tessellation with applications to Voronoi polytopes// The Computer Journal. 1981. Vol. 24, N 2. P. 167–172.
3. Артемьев А. Ю., Бартенев Ю. Г., Басалов В. Г., Бондаренко Ю. А., Варгин А. М., Голубев А. А., Ерзунов В. А., Ломтев А. В., Максимов А. С., Панов А. И., Прокофьев А. И., Романова М. Д., Фролова Н. В., Щаникова Е. Б. Библиотека решателей разреженных линейных систем // Труды РФЯЦ-ВНИИЭФ, Научно-исследовательское издание. Вып. 7. Саров: РФЯЦ-ВНИИЭФ, 2004. С. 80–95.