



ИПМ им.М.В.Келдыша РАН • Электронная библиотека

Препринты ИПМ • Препринт № 60 за 2008 г.



Щеглов И.А.

Программа для
триангуляции сложных
двумерных областей
Gridder2D

Рекомендуемая форма библиографической ссылки: Щеглов И.А. Программа для триангуляции сложных двумерных областей Gridder2D // Препринты ИПМ им. М.В.Келдыша. 2008. № 60. 32 с. URL: <http://library.keldysh.ru/preprint.asp?id=2008-60>

РОССИЙСКАЯ АКАДЕМИЯ НАУК
Ордена Ленина Институт прикладной математики
им. М.В. Келдыша

Щеглов И.А.

**Программа для триангуляции
сложных двумерных областей
Gridder2D**

Москва – 2008

Аннотация

Программное обеспечение Gridder2D предназначено для построения треугольных сеток в сложных двумерных областях. В данной работе приведены описание ПО, его особенности, спецификации форматов входных и выходных данных, а также ряд примеров задания входных данных.

Abstract

Software Gridder2D is intended for triangulation of complex 2D volumes. This paper contains the software description, input and output data format specifications and a number of examples.

Содержание

1. Описание	3
1.1 Назначение ПО Gridder2D	3
1.2 Характеристики и ограничения ПО	3
1.3 Системные требования	5
1.4 Краткое описание алгоритма дискретизации	6
2. Интерфейс ПО	7
2.1 Основная (консольная) часть	7
2.2 Визуальная надстройка Gridder2D-VI	7
3. Формат входных и выходных данных	14
3.1 Формат входных данных	15
3.2 Формат выходных данных	17
4. Примеры	20
4.1. Примеры файлов с описателями областей	20
4.2. Примеры файлов проекта	28
5. Рекомендации и советы	29
6. Возможные ошибки и проблемы	30
6.1 Коды возврата Gridder2D	30
6.2 Известные проблемы Gridder2D-VI	31
7. Список литературы	32

1. Описание

Программное обеспечение (ПО) Gridder2D состоит из двух утилит - самой программы Gridder2D.exe, вызываемой из командной строки, и графической надстройки Gridder2D-VI.exe, которая представляет собой визуальный интерфейс к основной программе, а также обладает рядом дополнительных функций, таких как оценка качества сетки и сбор статистических сведений об элементах построенной сетки. Входные и выходные данные для программы записываются в виде текстовых файлов.

Работа выполнена при частичной финансовой поддержке Российского фонда фундаментальных исследований (проект № 06-01-00421).

1.1 Назначение ПО Gridder2D

ПО Gridder2D предназначено для построения **равномерных** треугольных сеток в сложных двумерных (плоских) областях. В данном контексте сложными называются области, которые состоят из нескольких подобластей; причем сетку в области необходимо построить так, чтобы ребра триангуляции не пересекали границ между подобластями, т.е. каждый элемент (треугольник) сетки должен целиком и полностью принадлежать только одной подобласти.

Сетки, построенные с помощью Gridder2D, могут быть использованы в расчетах конечно-элементными и интегро-интерполяционными методами. На рис. 1 и 2 приведены характерные примеры сеток, построенных в Gridder2D.

1.2 Характеристики и ограничения ПО

- Максимальный размер сеток - до 2 млн. элементов.
- Линейные размеры элементов должны находиться в пределах 10^{-6} - 10^6 .
- Данные записываются в файлы с точностью до 14 знаков после запятой.
- Точность построения сетки - 10^{-10} .

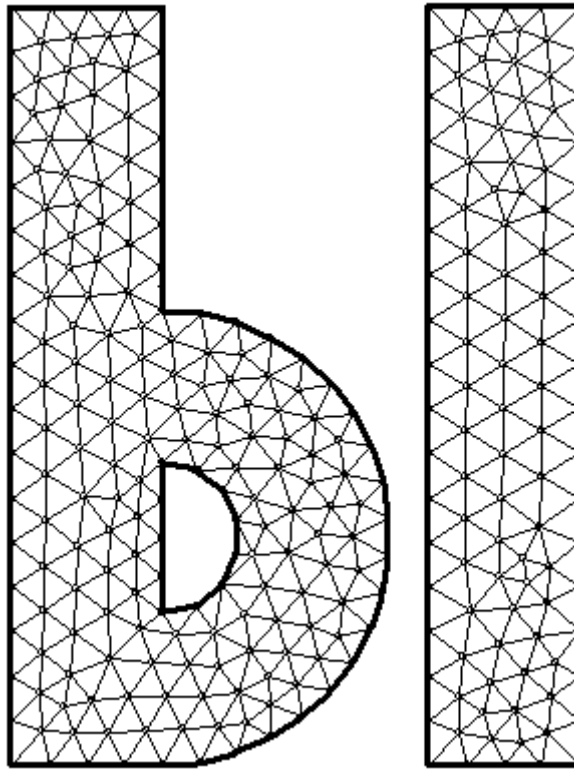


Рис. 1. Пример сетки, построенной с помощью ПО Gridder2D - несвязная область с отверстием в одном из фрагментов.

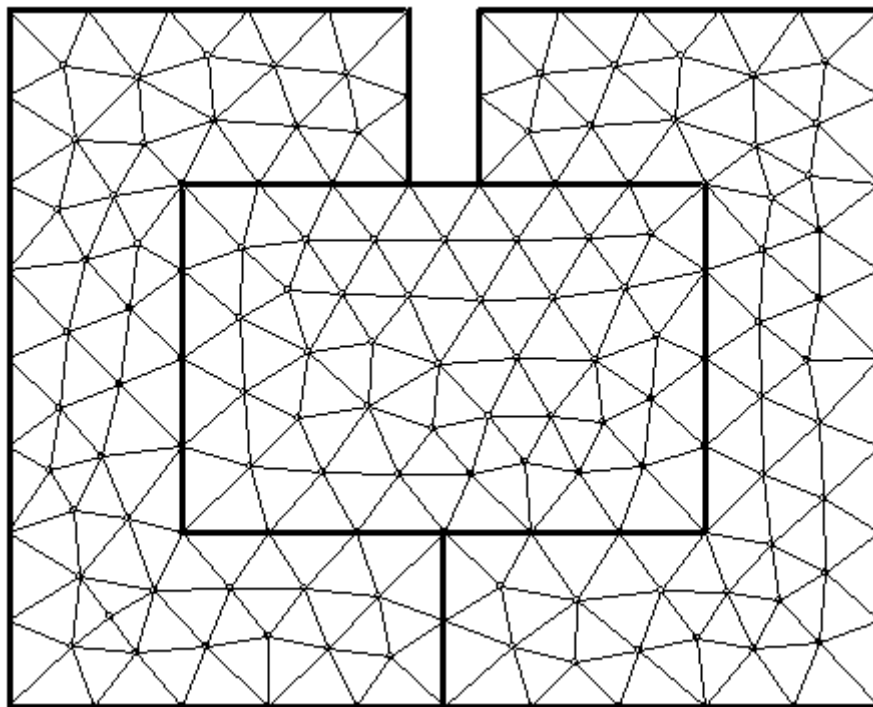


Рис. 2. Пример сетки, построенной с помощью ПО Gridder2D - область, состоящая из нескольких подобластей.

- Все вычисления производятся с двойной точностью.
- Построенные сетки, как правило, будут удовлетворять критерию Делоне, однако это свойство не гарантируется.
- С помощью Gridder2D можно триангулировать любую область, границу и внутренние ограничения которой можно описать конечным числом замкнутых непересекающихся контуров, представляющих собой ломаные линии из произвольного числа отрезков. Иными словами, область может быть несвязной, невыпуклой, иметь отверстия и внутренние ограничения в виде линий.

1.3 Системные требования

ПО предназначено для работы под операционными системами семейства Windows. Скорость работы ПО напрямую зависит от тактовой частоты процессора и объема оперативной памяти. Высокая скорость работы ПО достигается только в том случае, если все необходимые данные могут быть размещены в оперативной памяти компьютера. Следует исходить из того, что 1 МБ оперативной памяти в 32-разрядной системе хватает на хранение примерно 5-7 тыс. элементов сетки. Если оперативной памяти для хранения всей сетки окажется недостаточно, Windows будет использовать виртуальную память, что существенно снизит скорость работы программы.

Следует также учитывать то, что часть оперативной памяти занята самой операционной системой и другими запущенными программами.

При расчете места на жестком диске, которое потребуется для записи сеток, можно использовать следующие оценки:

Для сетки из 1 000 элементов: ~50 кБ

Для сетки из 10 000 элементов: ~600 кБ

Для сетки из 100 000 элементов: ~7 МБ

Для сетки из 1 000 000 элементов: ~80 МБ

1.4 Краткое описание алгоритма дискретизации

В ПО Gridder2D используется вариант метода исчерпывания [2], разработанный автором и носящий условное название "алгоритм от угла". Построение сетки состоит из следующих этапов:

1) Исходные контуры области разбиваются на отрезки так, чтобы длина отрезков не превосходила заданного параметра h (шаг триангуляции).

2) Каждый из контуров последовательно преобразуется в начальный фронт и передается в качестве начальных данных алгоритму "от угла" (описание алгоритма приведено ниже).

3) Окончательно сетки в построенных подобластях оптимизируются методом сглаживания по Лапласиану. Суть данного метода заключается в том, что каждый внутренний узел сетки помещается в центр тяжести системы своих узлов-соседей. Этот процесс повторяется итерационно заданное количество раз. Такая простая оптимизация не требует много времени, но позволяет существенно улучшить качество сетки.

Входными данными каждой итерации алгоритма "от угла" является фронт триангуляции - замкнутая цепь ребер, охватывающая еще не триангулированную область. На первом шаге определяется узел фронта, обладающий наименьшим углом. Если этот угол меньше определенного значения, то ребра, прилегающие к углу, соединяются третьим ребром, в результате чего получается новый элемент сетки. Если угол слишком велик, чтобы в результате получился приемлемый элемент, то угол разбивается новым отрезком в отношении 1:1 или 1:2 (в зависимости от величины угла) и этот отрезок соединяется с одним из ребер для образования нового элемента. Также дополнительно проводится проверка, нет ли в секторе радиуса h , вырезаемом углом в исчерпываемой области, уже существующих узлов сетки, которые можно использовать для образования нового элемента.

Каждый новый треугольник сетки проверяется на корректность (лежит ли он внутри исчерпываемой области и не пересекается ли уже существующими элементами). Если треугольник оказывается некорректным, он отбрасывается, а алгоритм ищет другие варианты. Заметим, что ситуация, когда других вариантов может не оказаться, исключена.

На каждой итерации образуется 1 новый элемент (треугольник) сетки, после чего фронт обновляется и алгоритм продолжает работу, пока длина фронта не станет равной 0. Заметим, что фронт в процессе работы может стать несвязным (т.е. разбиться на несколько замкнутых контуров), однако это никоим образом на алгоритм не повлияет.

2. Интерфейс ПО

2.1 Основная (консольная) часть

Утилита Gridder2D является консольной программой, т.е. вызывается из командной строки. Программа вызывается с единственным параметром - указанием на файл проекта. Если имя файла проекта или путь к нему содержат пробелы, буквы кириллицы или спецсимволы, его необходимо заключать в кавычки. Формат файла проекта описан ниже, в разделе 3.1.2. Примеры вызова утилиты Gridder2D из командной строки:

```
Gridder2D.exe working.txt
Gridder2D.exe c:\temp\working.txt
Gridder2D.exe strong_beam.txt
Gridder2D.exe "my project.txt"
Gridder2D.exe "c:\Мои Документы\project1.txt"
```

2.2 Визуальная надстройка Gridder2D-VI

Надстройка Gridder2D-VI обеспечивает утилите Gridder2D визуальным интерфейсом. Сама по себе она не содержит процедур триангуляции, однако

позволяет создавать файлы проекта, визуализировать файлы с описателями области и построенные сетки, а также позволяет проводить их анализ. Перед тем, как использовать Gridder2D для автоматического построения сеток, рекомендуется сначала с помощью Gridder2D-VI построить сетку в нужной области и, проанализировав ее, убедиться, что область задана корректно, сетка строится правильно и обладает необходимым качеством при заданных параметрах триангуляции.

Программу Gridder2D.exe можно использовать отдельно, однако для работы Gridder2D-VI.exe необходимо наличие Gridder2D.exe (файлы Gridder2D.exe и Gridder2D-VI.exe должны находиться в одной папке).

Ниже дано описание элементов всех вкладок программы Gridder2D-VI.exe.

2.2.1 Вкладка "Проект"

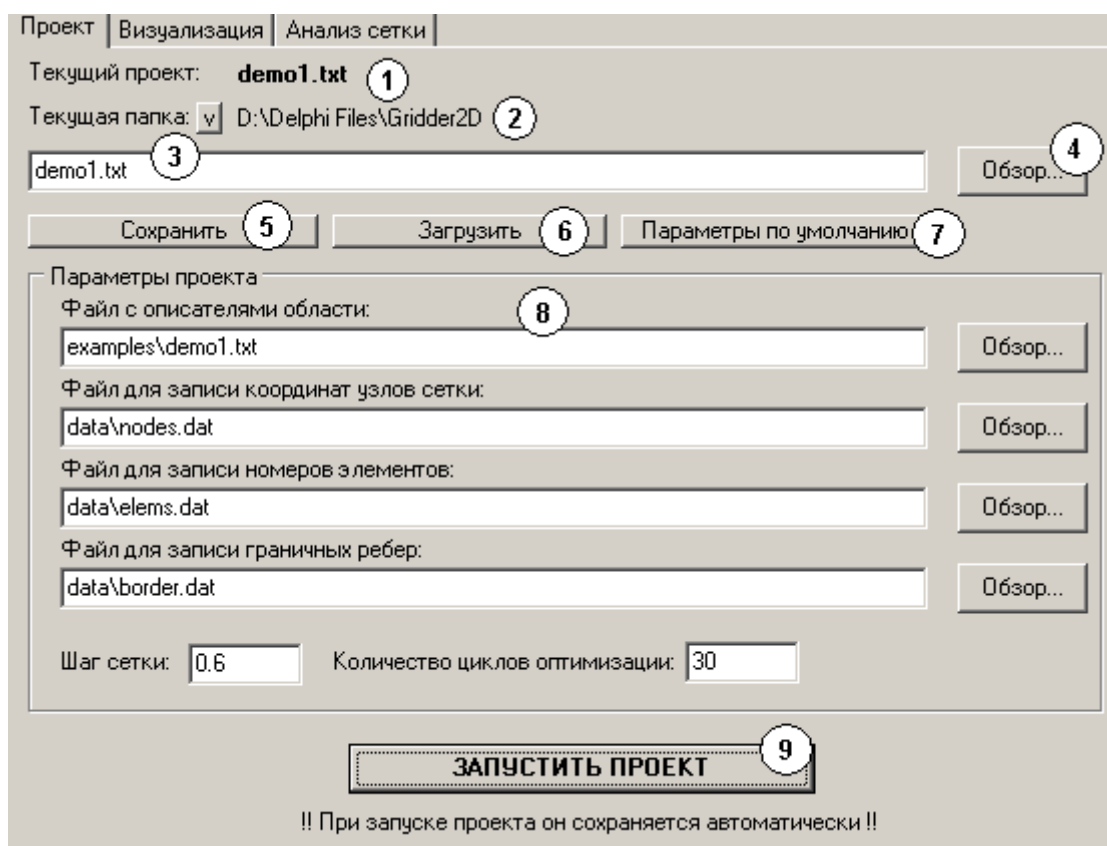


Рис. 3. Вкладка "Проект".

Эта вкладка предназначена для управления файлами проекта (см. раздел 3.1.2). Она позволяет создавать, загружать, редактировать и сохранять файлы проектов.

В поле (1) выводится название текущего проекта. Если никакого проекта не загружено, в поле будет указано слово "Нет". При закрытии программы Gridder2D-VI запоминает текущий файл проекта (его название записывается в текстовый файл recent.dat в папке программы) и автоматически загружает его при следующем запуске.

В поле (2) выводится адрес текущей рабочей папки (его можно оттуда скопировать). При нажатии кнопки "v", расположенной рядом с полем, адрес рабочей папки будет добавлен ко всем именам файлов на панели параметров проекта (8).

Текстовое поле (3) служит для работы с файлом проекта. Чтобы сохранить текущий проект под новым именем, измените название файла по своему усмотрению и нажмите кнопку (5). Нажатие кнопки (6) загрузит проект с именем, указанным в поле (3). Воспользуйтесь кнопкой (4), чтобы внести в поле имя файла с помощью стандартного диалога Windows. Внимание: файлы перезаписываются без предупреждения!

Нажмите кнопку (7), чтобы заполнить все поля значениями по умолчанию.

В поля на панели (8) заносятся параметры проекта. Их можно редактировать по своему усмотрению. Внимание: внесенные изменения вступят в силу только после сохранения проекта!

Нажмите кнопку (9), чтобы запустить текущий проект. При этом проект будет сохранен автоматически.

2.2.2 Вкладка "Визуализация"

Эта вкладка служит для визуализации построенной сетки или файла с описателями области.

Поле (1) служит для вывода изображения. При этом используются стандартные средства GUI Windows, т.е. для работы программы не требуется

видеокарта с 3D-ускорителем или какие-либо кодеки. Размеры окна для вывода графики привязаны к размерам окна программы. Чтобы максимально увеличить размер рисунка, разверните окно программы на весь экран.

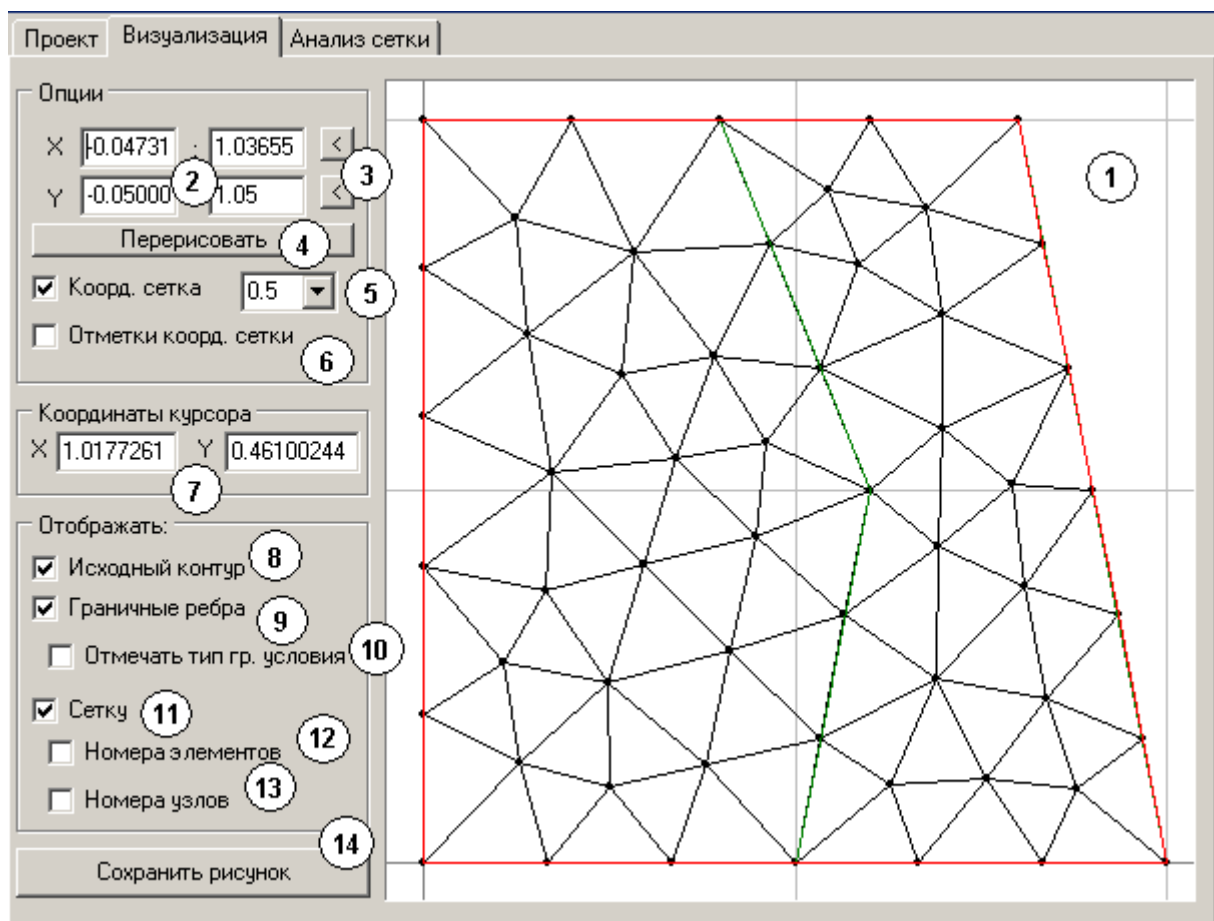


Рис. 4. Вкладка "Визуализация".

Изображением в окне (1) можно манипулировать с помощью мыши. Нажимая левую кнопку мыши, можно перемещать изображение вправо/влево и вверх/вниз. Нажимая правую кнопку мыши, можно изменять масштаб изображения: чтобы увеличить масштаб (приблизить изображение), нужно нажать правую кнопку мыши и повести курсор вверх, чтобы уменьшить - нажать правую кнопку мыши и повести курсор вниз.

Размеры изображаемой области также можно задавать напрямую, вводя нужные значения в полях (2), при этом масштабы по осям могут не совпадать. Чтобы уравнивать масштабы по осям, можно воспользоваться кнопками с галочками (3). Верхняя кнопка устанавливает для оси OX такой же масштаб,

какой в текущий момент установлен для оси ОУ. Нижняя кнопка делает то же самое для оси ОУ.

Чтобы любые изменения, задаваемые в правой части окна, начали действовать, следует нажать кнопку (4), которая обновляет изображение в окне (1).

Поле (5) позволяет включать/отключать вывод координатной сетки и задавать частоту проведения координатных линий.

При установленном флажке (6) у краев координатных линий будут выводиться отметки со значениями координат. Координатные оси рисуются более жирными линиями и отметки для них никогда не отображаются.

В полях (7) выводятся текущие координаты курсора в системе координат отображаемой области. Точность данных значений зависит от размеров окна. Чтобы получить более точные значения, следует максимально увеличить фрагмент области, координаты которого нужно узнать.

При отмеченном флажке (8) в поле (1) будет отображаться исходный контур (зеленым цветом). Чтобы эта опция стала доступна, необходимо сохранить текущий проект (даже если в загруженный проект не было внесено никаких изменений). Таким образом можно проверить, насколько верно задана область в файле с описателями области.

При отмеченном флажке (9) в поле (1) будут выводиться данные из файла для записи граничных ребер, т.е. красным цветом будут отображены все граничные ребра области (ребра, лежащие на линиях ограничений внутри области, граничными не считаются). Эти данные отображаются поверх данных об исходном контуре. Эта опция доступна только после обработки проекта. Если также отмечен флажок (10), то рядом с ребрами будет указано число, показывающее тип граничного условия на данном ребре (задается в файле с описателями области, см. 3.1.1). Опция (10) имеет какое-то влияние, только если включена опция (9).

Если отмечен флажок (11), в окне (1) будет рисоваться построенная сетка. Эта опция доступна только после обработки проекта. Если отмечены флажки

(12) или (13), то в сетке будут отмечаться номера элементов и узлов соответственно. Обе эти опции имеют влияние только в том случае, если включена опция (11).

С помощью кнопки (14) можно сохранить текущее изображение из окна (1) в одном из стандартных графических форматов Windows.

2.2.3 Вкладка "Анализ сетки"

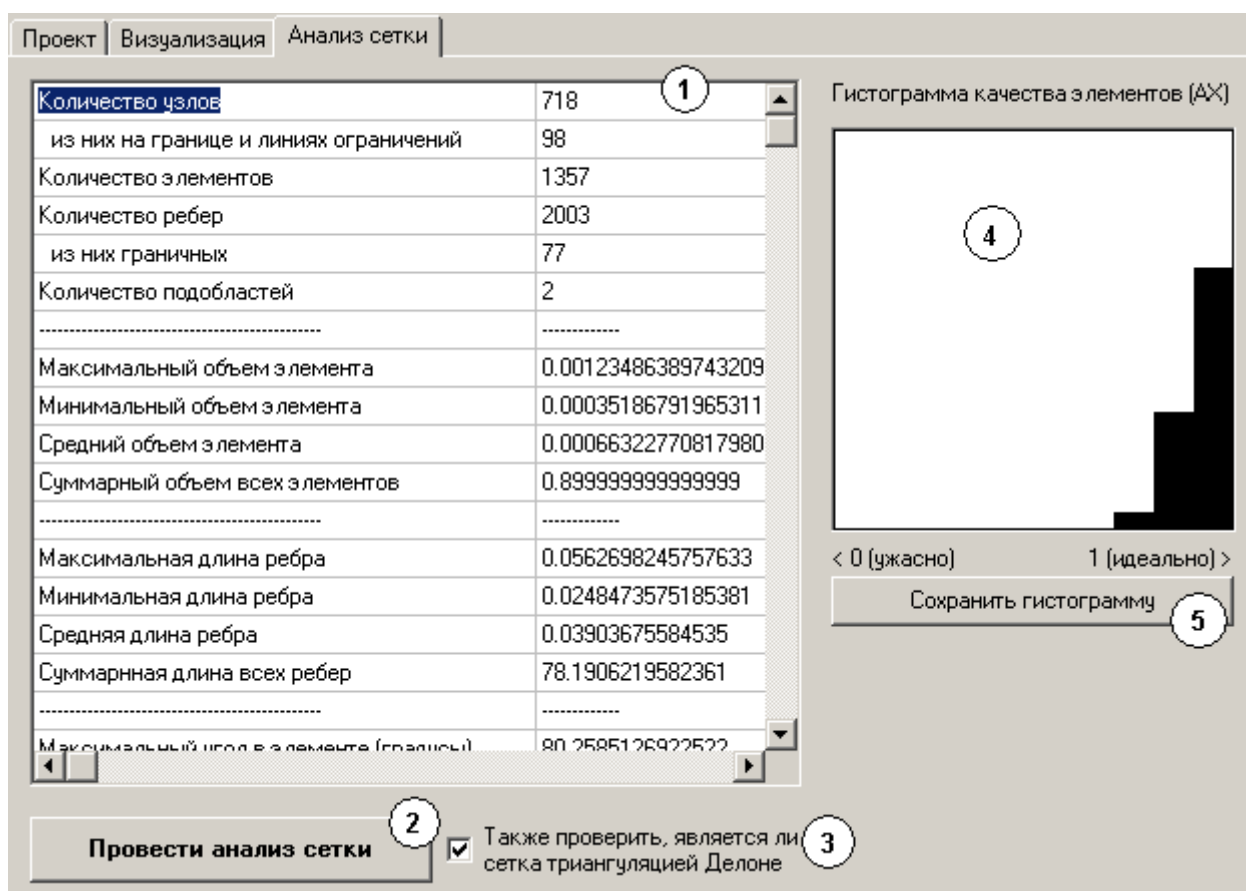


Рис. 5. Вкладка "Анализ сетки".

С помощью данной вкладки можно получить разнообразную информацию о построенной сетке, в том числе оценить ее качество.

Чтобы провести анализ сетки, необходимо нажать кнопку (2) (она доступна только после обработки проекта). В зависимости от размера сетки ее анализ может занять некоторое время (в любом случае существенно меньше времени, необходимого на построение). Если также требуется выяснить, является ли

построенная сетка триангуляцией Делоне с ограничениями, следует поставить флажок (3) (эта операция требует достаточно много трудоемких вычислений, поэтому вынесена отдельной опцией).

Пояснение: треугольник сетки считается удовлетворяющим критерию Делоне, если внутри описанной вокруг него окружности не попадают никакие другие узлы сетки. Если все элементы сетки удовлетворяют критерию Делоне, то такая сетка называется триангуляцией Делоне. Сетка называется триангуляцией Делоне с ограничениями, если внутри окружности, описанной вокруг любого элемента сетки, не попадают никакие другие *видимые* вершинам этого элемента узлы. Узел считается *видимым* другому узлу, если отрезок, соединяющий эти узлы, не пересекает никаких линий ограничений, наложенных на область. Триангуляции Делоне обладают рядом замечательных свойств, рассмотрение которых выходит за рамки данной работы [3].

По окончании анализа в список (1) будут выведены различные количественные характеристики сетки.

Примечания:

- Под объемом элемента понимается площадь треугольника, составляющего элемент.
- В сетках из большого числа элементов возможны небольшие (<1%) расхождения суммарной площади всех элементов и реальной площадью области триангуляции (вызвано накоплением ошибок округления).
- Максимальная длина ребра в сетке может оказаться немного больше шага триангуляции. Такое может произойти исключительно в результате оптимизации сетки. Чтобы избежать этого эффекта, следует либо отказаться от оптимизации, установив параметр "Количество циклов оптимизации" равным 0 (не рекомендуется), либо брать параметр "шаг триангуляции" чуть меньше желаемого.

- Если поле (3) отмечено, в конец списка (1) будет добавлен вердикт о том, является ли построенная сетка триангуляцией Делоне с ограничениями ("Да"/"Нет"). В случае отрицательного ответа в скобках также будет указан первый (по порядковому номеру) элемент, для которого не выполняется критерий Делоне.
- Качество сетки оценивается с помощью подсчета аппроксимационной характеристики (АХ) всех элементов. Аппроксимационной характеристикой называют нормализованную (т.е. отнесенную к идеальному случаю) численную оценку качества элемента [1]. В программе Gridder2D используется одна из наиболее релевантных (точных) оценок качества - отношение длины наименьшего ребра треугольника к радиусу описанной вокруг него окружности. АХ изменяется в пределах от 0 до 1; чем ближе к 1, тем лучше. Аппроксимационные качества всей сетки характеризуются наименьшим значением АХ среди всех элементов сетки (в частности, это значение явным образом входит в некоторые оценки ошибки аппроксимации методом конечных элементов [4]). Среднее значение АХ позволяет судить об общем качестве сетки.

В поле (4) выводится гистограмма, которая показывает распределение АХ среди элементов, что позволяет наглядно оценить качество сетки. С помощью кнопки (5) можно сохранить гистограмму в виде графического файла. Размеры рисунка фиксированы (200x200 пикселей).

3. Формат входных и выходных данных

Все данные для утилиты Gridder2D записываются в обычные текстовые файлы в кодировке ANSI. Расширение файлов значения не имеет. Числа в файлах должны отделяться друг от друга пробелами, запятыми или символом табуляции. В качестве разделителя десятичной части *необходимо* использовать точку. Дробные числа можно записывать как в десятичном, так и экспоненциальном формате.

3.1 Формат входных данных

3.1.1 Формат файла с описателями области

Файл с описателями области содержит данные о геометрии области и, при необходимости, данные о типах граничных условий.

Геометрия области описывается на основе т.н. "опорных точек", т.е. точек, на которых строятся контуры, охватывающие подобласти. Сами контуры представляют собой замкнутые ломаные линии, узлы которых являются опорными точками. (Примечание: во все опорные точки непременно будут помещены узлы сетки.) Контуры необходимо обходить против часовой стрелки относительно охватываемой ими области. Таким образом, чтобы описать геометрию области, нужно подобрать необходимое число опорных точек так, чтобы можно было обойти контуры всех подобластей, а все кривые сегменты границы аппроксимировать с достаточной точностью ломаными линиями.

Данные в файле с описателями области делятся на три логические секции: секция, описывающая опорные точки, секция, описывающая подобласти, и секция, описывающая граничные условия.

Данные следует указывать в следующем порядке (1-я секция):

Строка 1:

[количество опорных точек (NP)]

Следующие NP строк:

[координата X опорной точки][координата Y опорной точки]

Далее в файле описываются подобласти (2-я секция).

Строка NP+2:

[количество подобластей (NSV)][количество контуров, которые описывают каждую из подобластей, всего NSV чисел; сумма этих чисел равна общему количеству контуров (NC)]

Далее в этой секции описываются по порядку все контуры. Каждая запись имеет вид:

[количество узлов в контуре (NCP)][номера опорных точек, составляющих контур, всего NCP чисел]

Примечание: эти данные можно записывать в одну строку или разбивать на строки произвольным образом.

Следующая секция файла является необязательной и описывает типы граничных условий на данных сегментах контуров. Сегмент контура - это отрезок, соединяющий соседние опорные точки контура. Если для данного сегмента указан тип граничного условия (целое число), то для всех граничных ребер, которые попадут на этот сегмент, это число будет записано в файл для записи граничных ребер (см. раздел 3.2.3). Если ребро не попадает ни на один из описанных в этой секции сегментов, то для него будет указан тип граничного условия "0". Формат записи:

Первая строка в секции:

[количество сегментов с особым (ненулевым) типом граничного условия (NBT)]

Последующие NBT строк:

[n1][n2][Type]

n1 и n2 - номера опорных точек, Type - тип граничного условия, который будет записан для всех граничных ребер, попавших на этот сегмент (должно быть целым числом). Строка из этого списка будет иметь силу, только если указанная пара точек образует сегмент границы области или линии ограничения. Если какой-либо сегмент указывается более одного раза, учитывается только самое первое значение. Порядок указания опорных точек сегмента значения не имеет.

Примеры задания областей приведены в разделе 4.1.

3.1.2 Формат файла проекта

Файл проекта - это файл, который передается утилите Gridder2D.exe в качестве параметра и содержит ссылку на файл с описателями области, имена файлов для записи выходных данных и параметры построения сетки.

В файле проекта должно быть 6 строк (далее перечисляется содержание строк):

1) Файл с описателями области. Если файл находится в той же папке, что утилита, путь указывать необязательно.

2) Файл для записи координат узлов.

3) Файл для записи элементов.

4) Файл для записи граничных ребер.

5) Шаг триангуляции (желаемая максимальная длина ребра).

6) Количество циклов оптимизации.

Все последующие строки игнорируются (т.е. в них можно записать любые комментарии).

3.2 Формат выходных данных

Данные о сетке записываются в три текстовых файла в формате "узлы-элементы": первый файл содержит координаты узлов, второй - описатели элементов, третий - информацию о граничных ребрах. Ниже приведен формат всех трех файлов.

3.2.1 Файл для записи узлов элементов

1 строка:

[общее количество узлов (NN)][количество узлов, лежащих на границе области и линиях ограничений (NBN)]

Последующие строки (всего NN строк):

[Координата X узла][Координата Y узла]

Сначала перечисляются узлы, лежащие на границе и линиях ограничений (всего NBN строк); далее все остальные узлы. Запись в файл идет с двойной точностью (14 знаков после запятой).

3.2.2 Файл для записи номеров элементов

1 строка:

[количество элементов (NE)]

Последующие строки (всего NE строк):

[n1][n2][n3][e1][e2][e3][v]

Каждая строка содержит 7 целых чисел. Первые три (n_1, n_2, n_3) указывают на номера узлов, составляющих элемент; при этом узлы всегда обходятся против часовой стрелки. Следующие три (e_1, e_2, e_3) указывают на номера элементов, инцидентных текущему через ребро при соответствующей вершине. Т.е. соседний элемент e_1 имеет с текущим общее ребро n_1-n_2 , соседний элемент e_2 - ребро n_2-n_3 , а соседний элемент e_3 - ребро n_3-n_1 (см. рис. 6) Если соседний элемент имеет номер $>NE$, это означает, что элемент является фиктивным (т.е. данный элемент касается границы). Чтобы узнать номер соответствующего граничного ребра, достаточно из номера фиктивного элемента вычесть NE .

Последнее число обозначает номер подобласти, которой принадлежит элемент. Подобласти нумеруются согласно порядку их перечисления в файле с описателями области, начиная с 1.

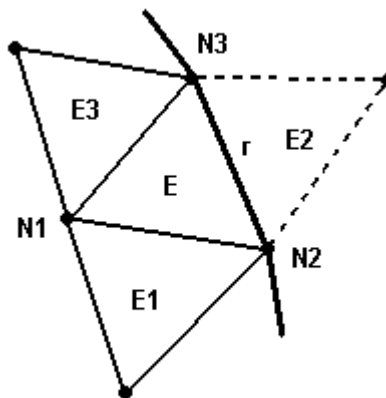


Рис. 6. Элемент с "соседями". Ребро n_2-n_3 - граничное, и элемент e_2 является фиктивным.

3.2.3 Файл для записи граничных ребер

В данном файле перечисляются все ребра сетки, лежащие на внешней границе триангулированной области и ее внутренних ограничениях. Файл содержит две секции; в первой описываются граничные ребра:

Строка 1:

[количество граничных ребер (NBR)]

Последующие строки (всего NBR строк) содержат по 4 целых числа:

[n1][n2][E][Type]

n1 и n2 - это номера узлов, составляющих граничное ребро. Номера узлов перечисляются в порядке обхода граничного контура против часовой стрелки.

E - номер элемента, которому принадлежит данное ребро.

Type - тип граничного условия. Берется из файла с описателями области. Если сегменту границы, которому принадлежит данное ребро, не назначен никакой тип граничного условия, будет записан 0.

Во второй секции перечисляются ребра, лежащие на линиях внутренних ограничений:

Строка NBR+2:

[количество ребер на ограничениях (NRR)]

Последующие строки (всего NRR строк) содержат по 5 целых чисел:

[n1][n2][E1] [E2] [Type]

n1 и n2 - это номера узлов, составляющих ребро.

E1 и E2 - номера элементов, которым принадлежит данное ребро.

Type - тип граничного условия на этом ребре. Берется из файла с описателями области. Если сегменту границы, которому принадлежит данное ребро, не назначен никакой тип граничного условия, будет записан 0.

Если ребер, лежащих на внутренних ограничениях, нет, то вместо NBR будет записан 0.

4. Примеры

ПО Gridder2D поставляется с рядом примеров файлов с описателями областей (папка "example") и файлов проекта (папка программы). Ниже описываются эти примеры и приводятся правила описания областей.

4.1. Примеры файлов с описателями областей

При описании областей необходимо понимать, как утилита Gridder2D производит триангуляцию области: сначала на равные отрезки разбиваются все сегменты контуров, затем отдельно и последовательно триангулируются все подобласти, охватываемые контурами. Таким образом триангуляция получается согласованной. Отсюда же следуют и требования к области:

- Область должна состоять из непересекающихся подобластей (но они могут соприкасаться друг с другом).
- Области могут быть несвязными (т.е. состоять из нескольких не соприкасающихся подобластей) и иметь отверстия.
- Объем подобласти должен быть больше нуля (также нельзя задавать ограничения в виде линий как отверстия нулевого объема).

Как правило, разбивка на подобласти определяется самой структурой исходной области триангуляции и/или решаемой в области задачей. В некоторых случаях, однако, имеет смысл разбить крупную подобласть на несколько более мелких (это позволяет несколько ускорить процесс триангуляции).

Таким образом, в файле необходимо описать каждую подобласть, в которой будет строиться сетка. Каждая подобласть может быть описана одним или несколькими контурами, т.е. замкнутыми ломаными линиями. Контур, в свою очередь, строится на заданном множестве опорных точек и определяется последовательным перечислением номеров опорных точек. Опорные точки необходимо перечислять так, чтобы описываемая подобласть всегда находилась

"слева" от направления обхода. Иными словами, внешнюю границу подобласти нужно обходить *против часовой стрелки*, а внутреннюю (в случае отверстия) - *по часовой стрелке*.

При описании контуров следует руководствоваться следующими правилами:

- Какая точка является начальной, значения не имеет.
- Пропускать опорные точки, лежащие на контуре, нельзя.
- Все заявленные опорные точки должны быть использованы.
- Все контуры должны быть замкнутыми (поскольку последний узел является и первым, указывать его не нужно).
- В случае, если две подобласти соприкасаются, общие сегменты их контуров должны быть описаны одинаковым образом, т.е. с указанием одних и тех же опорных точек (разумеется, в своем порядке для каждой подобласти).
- Контуры не должны пересекаться.
- Нежелательно также касание разных контуров только в одной точке (это может привести к ошибке алгоритма).
- Одна точка может использоваться при задании контура несколько раз.
- Каждый отрезок контура должен иметь ненулевую длину.

4.1.1 Пример №1 (файл "volume1.txt")

Данный файл описывает область, представляющую собой единичный квадрат. Первая строка указывает количество опорных точек (4), далее перечисляются координаты опорных точек (числа записаны с десятичной частью для наглядности). Следующие две строки указывают количество подобластей (1) и количество контуров в первой подобласти (1), затем определяется сам контур: первое число в строке указывает количество точек в контуре (4), далее следуют номера узлов в порядке обхода контура против часовой стрелки (рис. 7).

```

4
0.0 0.0
1.0 0.0
1.0 1.0
0.0 1.0
1
1
4 1 2 3 4

```

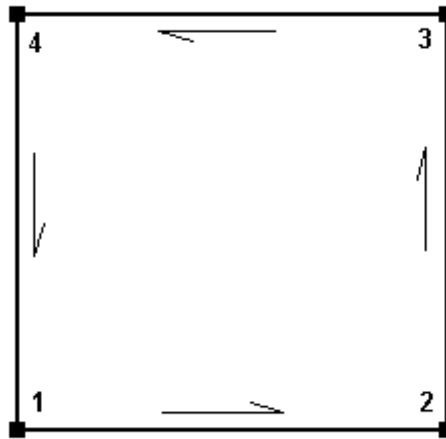


Рис. 7. Область, описываемая примером №1.

4.1.2 Пример №1а (файл "volume1a.txt")

В файл с описателями области также может быть внесена информация о граничных условиях, которые заданы на каком-либо сегменте контура. Эти данные представляют собой просто идентификатор (целое число), которое будет приписано всем граничным ребрам, принадлежащим указанному сегменту. Если сегменту не приписан никакой тип граничного условия, то ребру будет задан нулевой идентификатор.

Данные записываются в конце файла, после перечисления контуров, и имеют следующую структуру: 1-я строка - количество сегментов, на которых задано особое граничное условие, далее отдельными строками перечисляются (в произвольном порядке) сегменты в формате: узел 1, узел 2, тип граничного условия. Порядок перечисления узлов так же не важен.

```

4
0.0 0.0
1.0 0.0
1.0 1.0
0.0 1.0
1
1
4 1 2 3 4
2
1 2 1
4 3 2

```

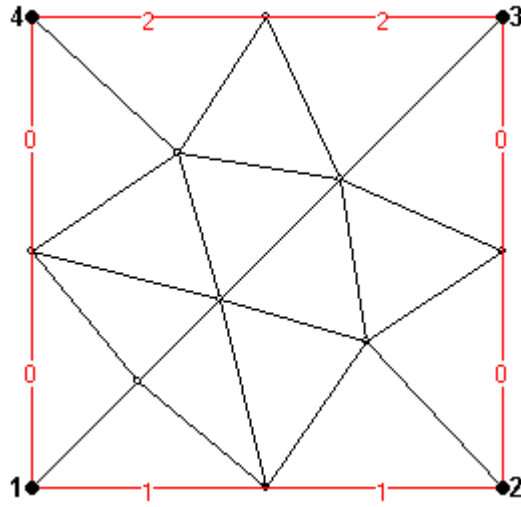


Рис. 8. Триангуляция с указанием типов граничных условий (пример №1а).

4.1.3 Пример №2 (файл "volume2.txt")

Этот пример описывает область, состоящую из двух соприкасающихся подобластей, каждая из которых описывается одним контуром.

```

7
0 0
0.5 0
1 0
0.6 0.5
0 1
0.4 1
0.8 1
2
1 1
5 1 2 4 6 5
5 2 3 7 6 4

```

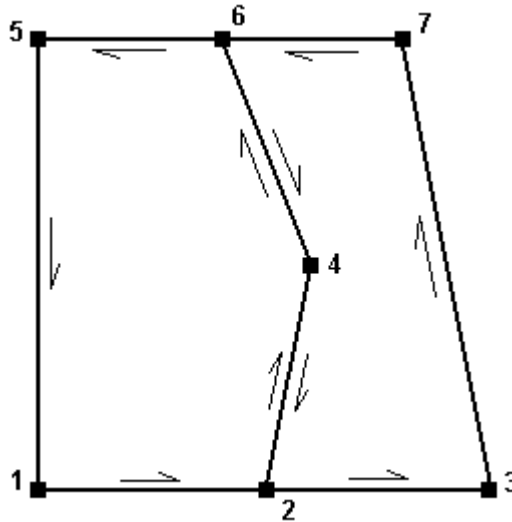


Рис. 9. Область, описываемая примером №2.

4.1.4 Пример №3 (файл "volume3.txt")

Этот пример описывает область, содержащую отверстие и внутреннее ограничение в виде линии (5-6). Вся область описана одним контуром, причем отрезок 5-6 проходится дважды.

```

9
0 0
1 0
1 1
0 1
0.4 0
0.4 0.2
0.75 0.5
0.5 0.9
0.2 0.45
1 1
11 1 5 6 9 8 7 6 5 2
3 4
    
```

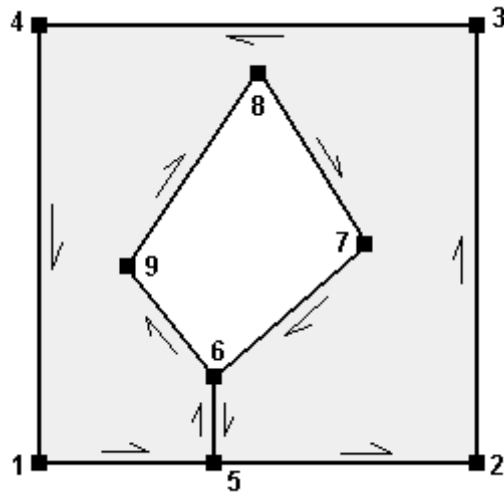


Рис. 10. Область, описываемая примером №3.

4.1.5 Пример №4 (файл "volume4.txt")

```

11
0 0
1 0
0 1
0.4 1
0.6 1
1 1
1 0.4
0.2 0.2
0.4 0.2
0.2 0.6
0.4 0.6
2 2 1
4 1 2 4 3
4 8 10 11 9
3 7 6 5
    
```

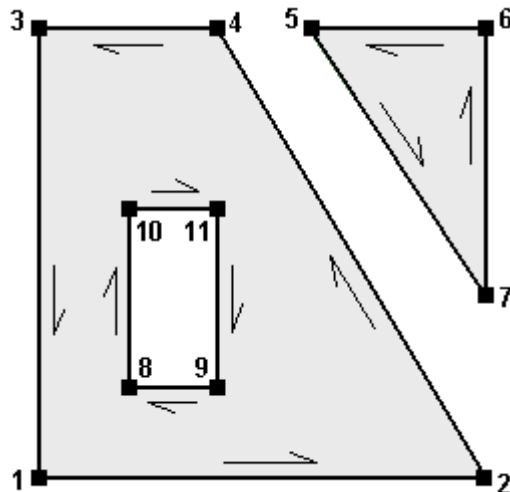


Рис. 11. Область, описываемая примером №4.

Этот пример описывает область, состоящую из двух подобластей, причем одна из областей содержит отверстие. Следует обратить внимание, что, в отличие от внешнего контура, отверстие обходится по часовой стрелке.

4.1.6 Пример №5 (файл "volume5.txt")

В некоторых случаях в области приходится вводить дополнительные (не обусловленные условиями задачи) ограничения. Например, в случае, когда необходимо аппроксимировать внутренние ограничения в виде отдельных линий. Линии невозможно описать замкнутым контуром как вырожденные отверстия, поскольку для них нельзя единственным образом определить направление обхода. Пусть область решения представляет собой квадрат с внутренней линией ограничения (рис. 12). Чтобы эту область можно было описать, необходимо дополнить ограничение, включив его во внешний контур (рис. 13).

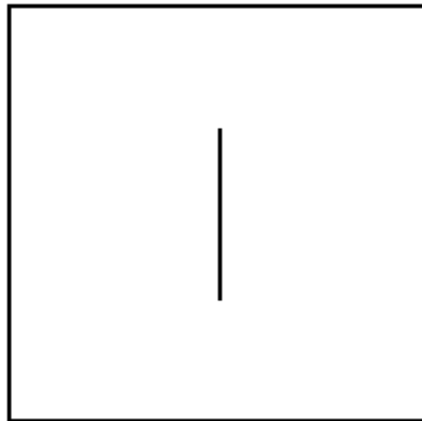


Рис. 12. Пример области с внутренним ограничением в виде линии. Такую область невозможно описать контурами напрямую.

```

7
0 0
1 0
1 1
0 1
0.5 0.3
0.5 0.7
0.5 1
1 1
9 1 2 3 7 6 5 6 7 4

```

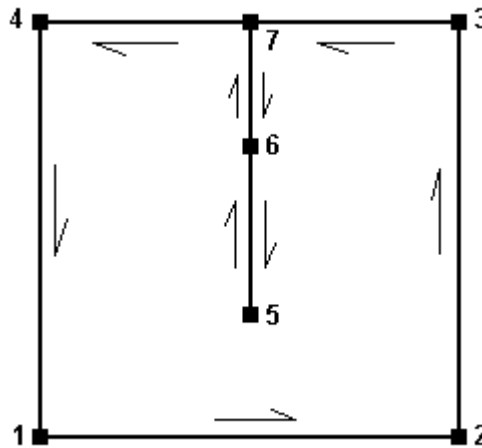


Рис. 13. Инкорпорация внутреннего ограничения во внешний контур.

4.1.5 Пример №6 (файл "volume6.txt")

В некоторых сеточных методах крайне нежелательно, чтобы ребра одного элемента принадлежали более чем одному сегменту границы. Однако, если триангулируется область, имеющая клиновидные выступы или иные острые внутренние углы (как, например, область на рис. 14), Gridder2D с большой долей вероятности поместит в такой угол только один элемент. Этого можно избежать, введя дополнительное ограничение в виде биссектрисы угла (рис. 15). Таким образом, в острый угол непременно будет помещено уже два элемента.

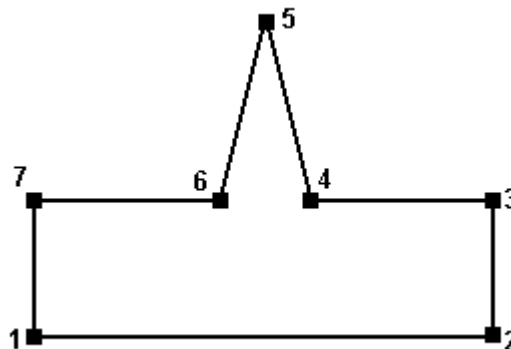


Рис. 14. Область с сильно острым внутренним углом.

```

8
0 0
1 0
1 0.3
0.6 0.3
0.5 0.7
0.4 0.3
0 0.3
0.5 0.4
1 1
9 1 2 3 4 5 8 5 6 7

```

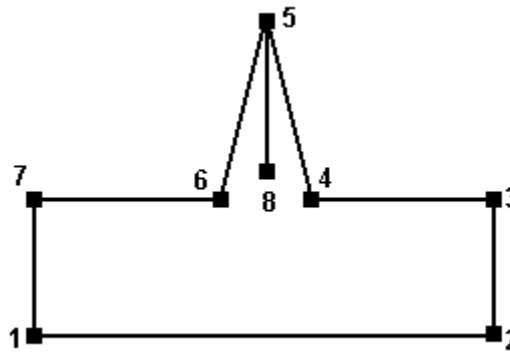


Рис. 15. Вставка дополнительного ограничения, чтобы вынудить Gridder2D поместить в угол 5 более одного элемента.

4.1.5 Пример №7 (файл "volume7.txt")

Если в триангулируемой области есть криволинейные фрагменты границы, их необходимо аппроксимировать ломаными линиями из достаточного количества отрезков (так, чтобы их длина не превосходила желаемого шага триангуляции). В этом случае для составления файлов с описателями области можно использовать программы, написанные на любом языке программирования, а также MATLAB или подобные ему программы.

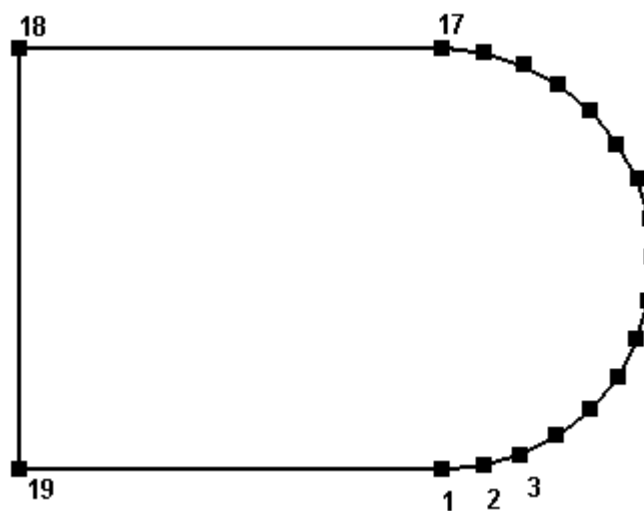


Рис. 16. Область, описываемая примером №7

```
19
2.0 0.0
2.195 0.019
2.382 0.076
2.555 0.168
2.707 0.292
2.831 0.444
2.923 0.617
2.980 0.805
3.0 1.0
2.980 1.195
```

```
2.923 1.382
2.831 1.555
2.707 1.707
2.555 1.831
2.382 1.923
2.195 1.980
2.0 2.0
0.0 2.0
0.0 0.0
1 1
19 1 2 3 4 5 6 7 8 9 10 11 12 13
14 15 16 17 18 19
```

4.2. Примеры файлов проекта

4.2.1 Пример №1 (файл "Project1.txt")

```
examples\volume1.txt
data\nodes.dat
data\elems.dat
data\border.dat
0.1
50
```

4.2.1 Пример №2 (файл "Project2.txt")

```
examples\volume7.txt
data\nodes.dat
data\elems.dat
data\border.dat
0.2
100
```

5. Рекомендации и советы

В данном разделе содержатся различные рекомендации и советы по работе с ПО Gridder2D, а также ответы на некоторые частые вопросы.

Номера узлов в опорных точках

Первые узлы сетки всегда помещаются в опорные точки, причем именно в таком порядке, в каком опорные точки перечисляются в файле с описателями области.

Номера узлов, лежащих на границе области

Чтобы узнать номера узлов, лежащих на границе области, необходимо обработать файл для записи граничных ребер. Для каждого ребра приводятся номера узлов, которые его составляют. В этом файле сначала перечисляются ребра, лежащие на внешней границе, а затем - лежащие на внутренних ограничениях.

Номера узлов, лежащих на линиях внутренних ограничений

Чтобы узнать номера узлов, лежащих на линиях внутренних ограничений, необходимо обработать файл для записи граничных ребер. Для каждого ребра приводятся номера узлов, которые его составляют. В этом файле сначала перечисляются ребра, лежащие на внешней границе, а затем - лежащие на внутренних ограничениях.

Определение граничных элементов

Для всех элементов сетки приводятся номера соседних с ними элементов. Если номер элемента-соседа больше, чем общее количество элементов (NE), значит, этот элемент - фиктивный, т.е. соответствующее ему ребро - граничное. При этом, чтобы получить номер сопоставленного ему граничного ребра, необходимо из номера фиктивного элемента вычесть NE.

Соотношение количества узлов, элементов и ребер

В двумерных треугольных сетках количество элементов обычно примерно вдвое больше количества узлов, а количество ребер - примерно втрое больше количества узлов.

Определение необходимого шага триангуляции исходя из желаемого количества элементов

Чтобы воспользоваться описанной ниже методикой, необходимо знать (точно или приближенно) площадь триангулируемой области.

Пусть площадь области равна S , а желаемое количество элементов - N . Тогда средняя площадь треугольника сетки будет равна S/N . Поскольку большая часть элементов сетки будет близка к правильному треугольнику, а для правильного треугольника длина ребра приблизительно равна $1.5\sqrt{S}$, получаем оценку: $h \approx 1.5\sqrt{S/N}$. Следует, однако, учитывать, что средняя длина ребра в сетке обычно в 1.3-1.6 раза меньше максимальной, поэтому в качестве окончательной следует использовать такую формулу: $h = 2\sqrt{S/N}$.

6. Возможные ошибки и проблемы

6.1 Коды возврата Gridder2D

По окончании работы Gridder2D можно стандартными методами Windows получить код возврата программы (errorcode), который позволяет определить, насколько удачно закончился процесс триангуляции. В таблице 1 приведены коды возврата и их значение.

Примечание. В силу особенностей обработки текстовых файлов ошибка задания файла с описателями области (P04) может не возникнуть, даже если в файле содержатся формальные ошибки (недостающие числа просто заменяются нулями). В этом случае программа обычно завершается ошибками исполнения (X01 или X02). Перед запуском крупных проектов убедитесь, что область задается корректно (Gridder2D-VI позволяет строить контуры областей, не строя саму сетку).

Таблица 1. Коды возврата Gridder2D

Код	Тип	Описание	Возможные причины
0	-	Триангуляция прошла успешно	
1	X01	"Не найдено ребро фронта"	Контур области задан некорректно.
2	X02	"Не удалось обновить ссылку на соседний элемент"	Подобласти пересекаются или иная ошибка задания области.
5	P01	"Параметр не найден или задан некорректно"	Утилита Gridder2D должна вызываться строго с одним параметром - ссылкой на файл проекта. См. раздел 2.1.
6	P02	"Не удалось считать файл проекта"	Файл проекта не найден или недоступен для чтения.
7	P03	"Ошибка в файле проекта"	Файл проекта содержит ошибки.
8	P04	"Не удалось считать данные области"	Файл с описателями области не найден или содержит ошибки.
9	P05	"Не удалось записать данные"	Диск защищен от записи или переполнен.

6.2 Известные проблемы Gridder2D-VI

По возможности в полях, где нужно указывать имена файлов, следует приводить полные имена файлов (с путем). Иначе при изменении рабочей директории (что может произойти, например, при использовании кнопок "Обзор...") весьма вероятно возникновение ошибок типа "файл не найден".

При задании имен файлов для записи сетки убедитесь, что указанный для них путь существует - новые папки программой автоматически не создаются.

7. Список литературы

1. Галанин М.П., Щеглов И.А. Разработка и реализация алгоритмов трехмерной триангуляции сложных пространственных областей: прямые методы. Препринт ИПМ им. М.В. Келдыша РАН, №10, 32с., 2006.
2. Галанин М.П., Щеглов И.А. Разработка и реализация алгоритмов трехмерной триангуляции сложных пространственных областей: итерационные методы. Препринт ИПМ им. М.В. Келдыша РАН, №9, 32с., 2006.
3. Скворцов А.В. Обзор алгоритмов построения триангуляции Делоне // *Вычислительные методы и программирование*, 2002, №3, с. 14-39.
4. Шайдуров В.В. Многосеточные методы конечных элементов. - М., Наука, 1989. - 288с.