УДК 519.6

doi 10.26089/NumMet.v16r449

МЕТОДЫ ПОСТРОЕНИЯ АДАПТИВНЫХ НЕСТРУКТУРИРОВАННЫХ СЕТОК ДЛЯ РЕШЕНИЯ ГИДРОГЕОЛОГИЧЕСКИХ ЗАДАЧ

А.В. Плёнкин¹, А.Ю. Чернышенко², В.Н. Чугунов³, И.В. Капырин⁴

Рассматриваются два сеточных генератора, внедренных в программный комплекс GeRa, предназначенный для решения задач геофильтрации и геомиграции радионуклидов. Это треугольно-призматический генератор с возможностью вырождения ячеек и генератор многогранных сеток с измельчением на основе восьмеричного дерева и возможностями "скалывания" ячеек. Генераторы позволяют автоматически строить конформные многогранные адаптивные сетки в трехмерных геологических областях с учетом сложной конфигурации внешних границ, кровель и подошв геологических пластов, выклинивания слоев и геологических неоднородностей.

Ключевые слова: генератор сеток, гидрогеологическое моделирование, треугольно-призматические сетки, сетки типа восьмеричное дерево.

Введение. Интенсивное развитие системы обращения с радиоактивными отходами (PAO), переход от отложенных решений (хранения) к захоронению РАО и возрастающие требования к безопасности пунктов захоронения в настоящее время обусловливают необходимость разработки расчетных кодов для трехмерного гидрогеологического моделирования. Их основной задачей является прогнозное моделирование процессов фильтрации и миграции радионуклидов в геологической среде. Отличительной особенностью разрабатываемых кодов должно стать использование новых вычислительных технологий, в частности адаптивных неструктурированных сеток, схем дискретизации высокой точности и параллельных вычислений.

В настоящей статье основное внимание уделено вопросам построения адаптивных неструктурированных сеток. Об актуальности данной тематики свидетельствует наличие различных методологий построения сеток, что нашло свое отражение в множестве различных разрабатываемых кодов. Один из характерных примеров — пакет FEFLOW [5] (разрабатывается германской компанией DHI-WASY GmbH) для моделирования подземных потоков, массообмена и теплообмена в пористых средах с трещинами. Эта система работает с призматическими сеточными элементами. Разработанный генератор производит сетку, которая обладает следующими свойствами: минимально возможное число тупых углов; мелкая сетка получается из грубой применением итерационного процесса; сетка учитывает внутренние ограничения (подземные объекты); представлена возможность локально сгущать сетку (например, вокруг скважин и в зонах загрязнений).

Другой известной разработкой, используемой для гидрогеологии, является naket MODFLOW для моделирования потока грунтовых вод через водоносные горизонты. В классической версии используются прямоугольные призмы с вертикальными боковыми гранями и скошенными основаниями. Модель позволяет задавать размер и количество ячеек по двум горизонтальным направлениям. Новая версия MODFLOW, называемая MODFLOW-USG [6] (разрабатывается геологической службой США), создана для поддержания широкого спектра структурированных и неструктурированных типов сеток, в том числе вложенных сеток на основе призматических треугольников, прямоугольников, шестиугольников и других форм ячеек. Гибкость при построении сетки может быть использована для изменения величины сеточных элементов вдоль рек и вокруг скважин.

Во Всероссийском научно-исследовательском институте экспериментальной физики (ВНИИЭФ, г. Саров) осуществлена параллельная реализация метода призматической дискретизации сложных геологических структур для гидрогеологического расчетного кода НИМФА [4]. Метод является двухэтапным.

(с) Научно-исследовательский вычислительный центр МГУ им. М. В. Ломоносова

¹ Институт проблем безопасного развития атомной энергетики РАН (ИБРАЭ РАН), Большая Тульская ул., д. 52, 115191, Москва; науч. сотр., e-mail: golden_dragon_84@mail.ru ²Институт вычислительной математики РАН (ИВМ РАН), ул. Губкина, д. 8, 119333, Москва; науч.

сотр., e-mail: chernyshenko.a@gmail.com

³ Институт вычислительной математики РАН (ИВМ РАН), ул. Губкина, д. 8, 119333, Москва; вед. науч. сотр., e-mail: chugunov.vadim@gmail.com

⁴ Институт проблем безопасного развития атомной энергетики РАН (ИБРАЭ РАН), Большая Тульская ул., д. 52, 115191, г. Москва; зав. лабораторией, e-mail: ivan.kapyrin@gmail.com

На первом этапе производится генерация двумерной неструктурированной сетки в параллельном режиме. Двумерный генератор предназначен для создания планарных сеток с числом точек более миллиона. После генерации двумерной неструктурированной сетки выполняется второй этап: построение трехмерной неструктурированной призматической сетки в параллельном режиме с возможностью адаптации сетки под различные типы объектов: скважины сложной траектории, геологические разломы, пласты и др.

Помимо указанных пакетов, среди различных открытых и коммерческих пакетов программ, моделирующих гидрогеологические и другие физические процессы, существуют коды, использующие генераторы гексаэдральных сеток. Во многих случаях используется блочная декомпозиция сетки (например, генератор ANSYS ICEM CFD [17]), сетки типа восьмеричное дерево [10] или вложенные сетки. Такие сетки не всегда строятся в автоматическом режиме; кроме того, остается открытым вопрос качественной аппроксимации сложной границы. Среди гексаэдральных генераторов широкого назначения можно отметить следующие коммерческие генераторы, позволяющие строить качественные гексаэдральные сетки в сложных областях на основе метода суперпозиции и сеток типа восьмеричное дерево: генератор HEXPRESS (разрабатывается компанией Numeca [14]), а также генератор Hexotic [16] (является частью французского комплекса MeshGems [15]). Наряду с этим следует отметить работу [13], в которой описан алгоритм построения и улучшения преимущественно гексаэдральной сетки для геологических областей.

С проблемой захоронения PAO связано и развитие задач экологии. Пакет Amanzi (разрабатывается в рамках программы ASCEM Министерства энергетики США) моделирует перенос радиоактивных загрязнений [9]. В этом пакете используются как структурированные, так и неструктурированные сетки. Данный пакет работает как с треугольными призмами, так и с произвольными многогранными ячейками.

Основные концепции указанных пакетов заложены и в российский код GeRa, сеточные генераторы которого и будут рассмотрены в настоящей работе. Использование адаптивных неструктурированных сеток призвано обеспечить высокую точность решения при малой вычислительной нагрузке.

Геометрической особенностью типичных гидрогеологических задач является слоистая структура расчетной области. Исходными данными для построения сетки являются контуры и области, отмеченные гидрогеологами на двумерных плановых картах различного назначения, и поверхности кровель и подошв геологических слоев. Этими контурами могут быть границы расчетной области, берега водоемов, линии геологических разломов, границы областей пространственной неоднородности параметров.

Сетки, используемые для гидрогеологических задач, должны отвечать следующим требованиям:

- отражение слоистой структуры расчетной области: аппроксимация сеткой кровель и подошв геологических слоев;
- наличие возможности локального сгущения сетки к скважинам или подземным объектам (например, к месту захоронения);
- учет на сетке выклиниваний пластов и геологических разломов;
- адаптация сетки под границы геологических неоднородностей;
- минимизация общего числа ячеек;
- аппроксимация сложных поверхностей гранями сетки со вторым порядком точности.

В нашей работе упор сделан именно на автоматизацию генерации на основе исходных данных сеток, удовлетворяющих вышеперечисленным требованиям. Основными инструментами в руках пользователягидрогеолога при построении сетки являются возможности задания и редактирования контуров, а также параметры сгущения и огрубления сетки в подобластях расчетной области.

Расчетный код GeRa (Геомиграция Радионуклидов) [1], разрабатываемый в ИБРАЭ РАН и ИВМ РАН, ориентирован на использование произвольных многогранных сеток. В коде выбор сделан в пользу двух сеточных генераторов. Первый генератор основан на построении треугольной сетки в двумерной проекции расчетной области и дальнейшем создании на ее основе треугольно-призматической сетки с возможностью получения также пирамидальных и тетраэдральных ячеек (например, при выклинивании пластов). Второй генератор создает иерархические сетки, ячейками которой преимущественно являются гексаэдры [3]. Иерархия ячеек построена на структуре восьмеричного дерева. Сетка адаптируется к границам области и внутренним границам путем "скалывания" гексаэдральных ячеек. Такая сетка являются конформной в обобщенном смысле: в случае, когда гексаэдральная (или сколотая многогранная) ячейка граничит по одной грани более, чем с одной ячейкой, эта грань условно разбивается на несколько отдельных граней так, чтобы каждой грани соответствовала только одна соседняя ячейка. Такой выбор

генераторов обусловлен несколькими соображениями. Во-первых, для прикладного гидрогеологического кода крайне важна автоматизация построения сетки — пользователь должен иметь только некоторые инструменты для адаптации сеток. Во-вторых, важно минимизировать количество ячеек. Это требование принуждает отказаться от хорошо развитых генераторов тетраэдральных сеток: при одинаковом шаге сетки число тетраэдров в 5–6 раз больше числа гексаэдров и приблизительно в 3 раза больше числа треугольных призм. В-третьих, такие сетки наилучшим образом подходят для типичных гидрогеологических задач, обладающих сильной геометрической анизотропией: обыкновенно мощность геологического слоя на порядки меньше его простирания. Получаемые ячейки способны повторить анизотропию расчетной области, при этом не имея двугранных углов, близких к 180 градусам.

Построение расчетных сеток производится в два этапа. На этапе предобработки заданные пользователем данные о гидрогеологических объектах преобразуются в формат данных сеточных генераторов. Затем производится собственно построение сетки. Кроме того, существует дополнительный этап постобработки, на котором в ячейки сетки записывается дополнительная информация о гидрогеологических объектах, не учтенных на первых двух этапах.

1. Преобразование гидрогеологических данных во входные данные сеточных генераторов. Существует ряд ограничений на входные данные сеточных генераторов. Так, например, в качестве входных данных не могут быть заданы перекрывающиеся подобласти. Кроме того, касание областей должно быть конформным (касание допускается только по целому ребру). Задаваемые пользователем объекты (водоемы, геологические неоднородности, ореолы существующего загрязнения) следует разбивать на логические группы и запрещать пересечения и перекрытия объектов внутри одной группы, в то время как объекты из разных групп могут пересекаться.

Для корректного учета задаваемых пользователем объектов разработан алгоритм, который на входе получает произвольные наборы ломаных. В результате его работы формируется набор непересекающихся контуров и конформно касающихся подобластей. При этом на ребрах построенных контуров должны быть заданы те же граничные условия, что и на соответствующих им ребрах исходного набора ломаных. Подобласти в результате работы алгоритма формируются без параметров. Наследование параметров исходных пользовательских областей разных типов полученными подобластями производится на втором этапе.

На вход алгоритма передаются границы всех областей, которые надо учитывать при построении сетки, и дополнительные ломаные (в частности, линии выклинивания геологических слоев). В таком случае любая полученная в результате работы алгоритма подобласть может только полностью лежать внутри исходной области. Может существовать несколько исходных областей разных типов, внутри которых лежит полученная подобласть. Параметры таких исходных областей будут определять параметры подобласти.

Итак, требуемый алгоритм состоит из предобработки (в ходе которой формируются дополнительные линии, к которым требуется проводить адаптацию сетки) и двух основных этапов.

На первом этапе по набору ломаных строится набор конформно касающихся подобластей и непересекающихся контуров (их границ), на которые перенесены граничные условия с исходных контуров. На втором этапе областям присваиваются параметры (сеточные, граничные условия, признак водоема и др.), соответствующие исходным областям, частями которых являются подобласти.

1.1. Геологические интерфейсы и формат хранения данных о положении границ геологических слоев. Положение геологических слоев принято задавать с помощью спецификации поверхностей кровли и подошвы геологического слоя в области, где этот слой залегает. В программном комплексе GeRa положение геологических слоев задается с помощью геологических интерфейсов — поверхностей, которые разделяют залегающие геологические слои или являются кровлей или подошвой расчетной области. Число геологических интерфейсов равно числу геологических слоев плюс один. Каждая поверхность геологического интерфейса задается для всей расчетной области. Каждый геологический слой ограничен двумя интерфейсами, которые совпадают соответственно с кровлей и подошвой этого слоя там, где он залегает, и совпадают между собой там, где слой вырождается.

При построении трехмерных сеток ключевую роль играет выбор правильного формата хранения данных о положении геологических интерфейсов, обеспечивающего максимальное быстродействие.

В настоящее время для задания границ геологических слоев в комплексе GeRa используется следующий формат. В плоскости Oxy задана равномерная прямоугольная сетка с ребрами, параллельными осям координат Ox и Oy. Для некоторого поднабора узлов этой сетки, определяемого положением границ расчетной области, на каждой из границ геологических слоев, включая кровлю и подошву, задается единственная соответствующая данной границе *z*-координата. При этом вследствие естественной неточности алгоритмов, используемых для интерполяции кровель и подошв, не исключены их пересечения в случае выклинивания слоя, которые должны быть должным образом обработаны при построении сетки.

Описанный формат весьма удобен как с точки зрения хранения данных, так и с точки зрения быстродействия при построении расчетной сетки. Действительно, при построении сетки часто возникает необходимость найти проекцию точки, заданной координатами (x, y), на каждую из границ геологических слоев. В общем случае этот процесс может требовать много времени, в особенности для больших сеток. При указанном формате задания границ геологических слоев для любой точки можно сразу же найти ближайший узел прямоугольной сетки, используемой для задания *z*-координат границ слоев. Затем находится прямоугольник, в который попадает точка, и производится интерполяция.

Следует отметить, что указанный подход поиска проекций точек на границы геологических слоев используется как в призматическом, так и в гексаэдральном сеточных генераторах.

1.2. Алгоритм формирования набора ребер, образующих линии выклинивания геологических слоев. В этом разделе опшпем алгоритм формирования набора ребер, образующих линии выклинивания или вырождения геологических слоев. Построение этих линий необходимо для адаптации сетки к выклиниваниям геологических слоев, которые задаются через поверхностные триангуляции кровли и подошвы. Этот алгоритм является универсальным в том смысле, что позволяет находить линии выклинивания геологических слоев даже в случае, когда кровля и подошва заданы произвольными пространственными триангуляциями.

Чтобы определить множество ребер, ограничивающих области совпадения кровли и подошвы геологического слоя, будем искать пересечение каждого треугольника из одной триангуляции с каждым сеточным элементом другой для произвольной пары триангуляций. При этом будем учитывать лишь пересечения двух пространственных треугольников по отрезку. В случае совпадения треугольников считаем, что они не пересекаются. Опишем алгоритм формально. Пусть eList — список ребер, каждое из которых определяется пространственными координатами концов. Обозначим через $n_t(k)$ число треугольников в k-й поверхностной триангуляции.

Алгоритм 1. Алгоритм вычисления границ совпадения k и m поверхностных триангуляций.

- 1: Положим *eList* пустой список.
- 2: для $i_1 = 1, n_t(k)$ начало цикла
- 3: для $i_2 = 1, n_t(m)$ начало цикла
- 4: Вычисляем пересечение треугольника *i*₁ из *k*-й поверхностной триангуляции с треугольником *i*₂ из *m*-й поверхностной триангуляции по алгоритму 2.
- 5: Добавляем новые ребра пересечения в список *eList*.
- 6: конец цикла
- 7: конец цикла

В алгоритме 1 нам важно лишь пересечение двух пространственных треугольников по отрезку. Это возможно, если эти два треугольника лежат в пересекающихся плоскостях. Рассмотрим треугольники *ABC* и *DEF* в трехмерном пространстве. Опишем в алгоритме 2 процедуру их пересечения, для простоты изложения не рассматривая случай, когда треугольники лежат в одной плоскости.

Алгоритм 2. Алгоритм пересечения пространственных треугольников ABC и DEF.

- 1: Определяем коэффициенты уравнений плоскостей z_1 , содержащей треугольник *ABC*, и z_2 , содержащей треугольник *DEF*.
- 2: Для каждого из отрезков DE, EF и FD проверяем пересечение с плоскостью z_1 . Если нет пересечений, то переходим к 7, иначе определяем две точки M_1 и N_1 , задающие отрезок пересечения треугольника DEF с плоскостью z_1 .
- 3: Для каждого из отрезков AB, BC и AC проверяем пересечение с плоскостью z₂. Если нет пересечений, то переходим к 7, иначе определяем две точки M₂ и N₂, задающие отрезок пересечения треугольника ABC с плоскостью z₂.
- 4: Если ни одна из точек M_1 и N_1 не принадлежит отрезку M_2N_2 или ни одна из точек M_2 и N_2 не принадлежит отрезку M_1N_1 , то переходим к 7.
- 5: Находим отрезок MN как пересечение отрезков M_1N_1 и M_2N_2 .
- 6: Треугольники ABC и DEF пересекаются по отрезку MN. Выходим.
- 7: Треугольники АВС и DEF не пересекаются.

Сделаем замечание по алгоритму 2. Чтобы определить, пересекает ли отрезок заданную плоскость, нужно сначала убедиться, что концы отрезка лежат в разных полуплоскостях, а после этого найти точку

пересечения отрезка с плоскостью.

В результате имеем набор ребер, образующих линии выклинивания геологических слоев.

1.3. Построение конформно касающихся неперекрывающихся областей на основе заданного набора подобластей и ломаных. Алгоритм ведет работу с наборами геометрических объектов. При этом постоянно происходит добавление и удаление элементов в массивы объектов. Для динамического хранения таких объектов используются двунаправленные списки. Используются следующие типы объектов: структура граничных условий, точка, ребро, контур, подобласть. Структура граничных условий хранит информацию о граничных условиях, которые задаются на ребрах. Объект "точка" хранит свои координаты и список инцидентных ей ребер. Объект "ребро" хранит ссылки на инцидентные ему узлы, граничные условия, а также несколько служебных параметров. Объект "контур" состоит из флага, определяющего, является ли этот контур внешней границей области, упорядоченного списка своих узлов (первый и последний совпадают) и упорядоченного списка граничных условий, соответствующих ребрам, соединяющим узлы. Объект "подобласть" состоит из списка контуров, ограничивающих данную область.

На входе алгоритм получает набор абсолютно произвольных ломаных (возможно, замкнутых) с заданными на их ребрах граничными условиями. По этим данным формируются начальные глобальные списки (pList) точек и ребер (eList).

В работе алгоритма можно выделить следующие шаги:

- списки узлов и ребер, сформированные по исходным данным, приводятся к виду, в котором они соответствуют правильному плоскому графу, в котором нет повторяющихся вершин, пересекающихся, совпадающих или касающихся ребер. Первый шаг является самым сложным, его можно разбить на три этапа:
 - а) объединение совпадающих узлов;
 - b) обработка пересечений ребер по узлу;
 - с) обработка остальных пересечений ребер;
- определяются вспомогательные параметры на ребрах (для каждой вершины ребра ищется крайний правый сосед (соседнее ребро), инцидентный данной вершине, и угол между ребром и соседом);
- 3) строится список контуров;
- 4) строится список подобластей.

Далее подробно опишем каждый из шагов.

1а. Перебираются все пары точек. Если расстояния между узлами меньше заданного малого порога, то вызывается операция объединения узлов. Для всех ребер, содержащихся в списках ребер, инцидентных первому и второму узлам, ссылки на второй узел заменяются ссылками на первый узел. После этого второй узел удаляется из списка узлов. Затем вызывается процедура очистки списков ребер. В ходе процедуры из списков ребер точек и глобального списка ребер удаляются вырожденные и повторяющиеся ребра.

1b. Для каждого из узлов проверяется, лежит ли он в малой окрестности какого-либо ребра. Если точка не является вершиной ребра и лежит в его малой окрестности, создаются два ребра, одной из вершин которых является проверяемая точка, а другими — вершины ребра. На созданных ребрах задаются те же граничные условия, что и на исходном ребре. Затем ребра добавляются в глобальный список ребер и в списки ребер инцидентных им узлов. В завершение исходное ребро удаляется из всех списков ребер.

1с. Перебираются все пары ребер. Если ребра не инцидентны (не имеют общего узла) и пересекаются, то точка пересечения добавляется в список узлов. Создаются четыре ребра, одной из вершин которых является точка пересечения исходных ребер, на ребра переносятся соответствующие граничные условия. Созданные ребра добавляются в глобальный список ребер и в списки ребер, инцидентных соответствующим узлам, включая точку пересечения. Затем исходные два ребра удаляются из всех списков ребер. В этот момент может нарушиться достигнутое на предыдущем этапе условие: отсутствие ребер, пересекающихся по вершине. Поскольку в одной точке могут пересекаться сразу несколько ребер, то созданная на этом шаге точка пересечения может лежать на других ребрах. Чтобы это исправить, для точки пересечения вызывается процедура из предыдущего этапа. Если в ходе перебора пар ребер найдено пересечение, то после его обработки вся процедура полностью перезапускается. После того как все пары ребер были проверены и пересечений не было найдено, вызывается процедура очистки списков ребер. 2. Для каждой из вершин ребра ищется крайний правый сосед (соседнее ребро) данного ребра, инцидентный данной вершине, и угол между данным ребром и соседом. Крайний правый сосед определяется по минимальному углу между ребром и его соседом, угол отсчитывается от данного ребра против часовой стрелки (рис. 1). Для этого перебираются ребра, инцидентные вершине, и вычисляются углы (как показано на рис. 1) между ними и текущим ребром. Если у ребра нет соседей через данную вершину, то соседом назначается это же ребро, а угол задается равным 2π радиан.

3. Строится список контуров (cList). На каждом ребре заводятся два дополнительных флага, которые показывают, было ли ребро использовано для построения контура при движении от первой точки ко второй и было ли оно использовано при движении от второй точки к первой. Изначально все ребра считаются неиспользованными. Запускается процесс перебора ребер, и если один из флагов использования ребра показывает, что ребро не использовано, то на основе этого ребра запускается процедура построения контура. При этом учитывается, какой из флагов сигнализировал о том, что ребро не использовано. Для одного и того же ребра процедура может быть запущена дважды. Если флаг указывает, что ребро не использовалось в направлении от первой точки ко второй, то в список узлов контура добавляется сначала первая точка, по-



Рис. 1. Положение крайних правых соседей

том вторая. В противном случае добавляется сначала вторая, потом первая. Для ребра устанавливается флаг, говорящий о том, что оно было использовано в указанном направлении. После этого на основе ребра и последней точки, добавленной в контур, ищется крайний правый сосед. Если крайний правый сосед при движении от последней добавленной в контур точки уже использовался, то процедура завершается, а первая точка контура добавляется в него еще раз, уже как последняя. Если же соседнее ребро в указанном направлении не использовалось, то направление помечается как использованное, а его вершина, отличная от последнего добавленного в контур узла, добавляется в контур. После этого процедура построения контура продолжается уже для крайнего правого соседа и последней точки контура. Для них ищется следующий крайний правый сосед, и так продолжается до тех пор, пока процедура не завершится по указанному выше признаку. Кроме того, в ходе процедуры при нахождении крайних правых соседей суммируются все углы между ребром и соседом, а также их дополнения до 2π радиан. Если сумма дополнений больше суммы углов, то контур помечается как "внешний".

4. Строится список подобластей (aList). Далее считается, что контур лежит внутри другого контура, если не существует вершин первого контура, лежащих вне второго контура, и хотя бы одна вершина первого контура лежит строго внутри (не на границе) второго контура.

Для каждого контура с признаком "внешний" создается область и в список ее границ добавляется указанный внешний контур, который будет внешней границей подобласти. Затем перебираются все контуры. Для контуров, не являющихся внешними и лежащими внутри внешней границы, выполняется следующая проверка. Если существует контур, не являющийся внешним, такой, что он расположен внутри внешней границы, на основе которой строится подобласть, и проверяемый контур лежит внутри него, то проверка провалена. Если же таких контуров не существует, то проверяемый контур добавляется в список границ подобласти.

Сложность алгоритма построения конформно касающихся неперекрывающихся подобластей пропорциональна $O(n_e^2 \cdot n_i)$, где n_e — число ребер в исходном наборе ломаных, а n_i — число точек пересечения ребер.

На этом процесс построения неперекрывающихся областей завершается.

1.4. Определение свойств подобластей на основе свойств исходных областей. Изначально все подобласти, полученные в результате разбиения, помечаются как неактивные (в таких областях сетка не строится), поскольку при разбиении могут быть сформированы подобласти, которые не принадлежат расчетной области.

В настоящий момент исходные области можно разделить на три группы: водоемы; локальные области геологических неоднородностей (область, определяющая неоднородность только в одном геологическом слое); прочие области (глобальные).

Сначала ищется произвольная точка-пробник, лежащая внутри подобласти. Поскольку подобласть

Алгоритм 3. Алгоритм построения подобластей.

- 1: Создаем пустой список областей aList.
- 2: для contour = cList.begin(), cList.end() начало цикла
- 3: если contour имеет признак "внешний" тогда
- 4: Создаем область area
- 5: Добавляем contour в area.cList
- 6: для $contour_1 = cList.begin(), cList.end()$ начало цикла
 - если (contour₁ не имеет признак "внешний") и (contour₁ лежит внутри contour) тогда для $contour_2 = cList.begin(), cList.end()$ начало цикла если (contour₂ не имеет признак "внешний") и (contour₂ лежит внутри contour) и
- $(contour_1$ лежит внутри $contour_2$) тогда
 - Переходим к 6 и проверяем следующий contour₁
- 11: конец если
- 12: конец цикла
- 13: Добавляем *contour*₁ в список контуров подобласти *area.cList*
- 14: конец если
- 15: конец цикла
- 16: Добавляем area в aList
- 17: конец если
- 18: конец цикла

может лежать внутри исходной области только целиком, то подобласть лежит внутри исходной области тогда и только тогда, когда точка-пробник подобласти лежит внутри исходной области. Следует отметить, что точки, лежащие на границе подобласти, для пробников не подходят. После того как найдена точка-пробник, начинается перебор всех исходных областей и проверяется, принадлежит ли пробник области. Если пробник принадлежит области, то, в зависимости от ее типа, выполняется одно из следующих действий. Если исходная область является водоемом, то подобласти присваивается идентификатор, соответствующий данному водоему. Если область является локальной неоднородностью, то в слое подобласти задается неоднородность (номер слоя и материала неоднородности определяются параметрами, заданными в исходной области). От глобальной области наследуются следующие параметры: набор граничных условий на кровле и подошве; флаг того, что область учитывается при построении сетки; скорость разгрубления сетки по мере удаления от границы области (для призматического генератора); средний размер сеточных элементов (для гексаэдрального генератора); флаг того, что размер сеточных элементов ограничен; величина ограничения размера сеточных элементов; данные неоднородностей. При переносе данных о неоднородностях из глобальной области проверяется, что для данного слоя в подобласть еще не была импортирована информация о неоднородности из локальной области неоднородности. Если данные уже импортировались, то информация из глобальной области игнорируется.

2. Схема работы генераторов. В этом разделе представлены два сеточных генератора, используемых в расчетном комплексе GeRa. Следует подчеркнуть, что задачей генераторов является не только построение сетки внутри расчетной области, такой, что грани ячеек аппроксимируют границы геологических слоев, неоднородностей и выклинивания, но и заполнение тегов (задаются в ячейках или на гранях), определяющих расположение водоемов, неоднородностей, свойства среды и граничные условия. Таким образом, в результате получается не только каркас сетки, но и информация о том, как по ней распределены заданные пользователем параметры.

2.1. Общая схема работы треугольно-призматического генератора. Построение сетки производится в два этапа. Сначала строится двумерная триангуляция расчетной области, адаптированная к заданным пользователем гидрогеологическим объектам. На треугольники и ребра переносится информация о граничных условиях и расположении объектов. На втором этапе строится треугольно-призматическая сетка. Пользователь может как сразу построить итоговую сетку, так и выполнять построение поэтапно, при этом построенная на первом этапе триангуляция будет наложена на карту и можно будет визуально оценить ее качество, корректность аппроксимации заданных гидрогеологических объектов и соответствие размеров полученных треугольников требованиям моделирования. В случае несоответствий можно внести изменения в модель и перестроить триангуляцию.

Исходными данными для построения двумерной триангуляции являются набор конформно касающихся неперекрывающихся подобластей с соответствующей информацией о гидрогеологических объектах, привязанной к ним (см. раздел 1).

На первом этапе производится независимая триангуляция каждой из подобластей. Рассматриваются

7:

8:

9:

10:

все скважины и точечные источники, попадающие в данную подобласть. Для каждого из объектов строится равносторонний треугольник определенного пользователем размера. Построенные треугольники не должны пересекать друг друга или границу области, поэтому на этапе задания скважин производятся проверки, позволяющие исключить подобного рода ошибки. Треугольники считаются частью будущей триангуляции области и вырезаются из подобласти. Триангуляция оставшейся части подобласти осуществляется методом продвигаемого фронта [2, 7, 8]. На треугольники и ребра построенной триангуляции переносится информация о гидрогеологических объектах, заданная в подобласти.

Наконец, триангуляции отдельных подобластей сливаются в единую триангуляцию расчетной области (при этом объединяются дублирующиеся вершины и ребра). Пример триангуляции расчетной области, состоящей из двух подобластей, одна из которых содержит скважину, представлен на рис. 2.

Сложность построения триангуляции подобласти методом продвигаемого фронта не превышает $O(n_e^3)$, а в среднем составляет $O(n_e ln(n_e))$, где n_e — число ребер в границе этой подобласти.

После того как построена триангуляция двумерной расчетной области, начинается построение треугольно-призматической сетки. Для каждой вершины триангуляции находятся точки пересечения вертикального луча, проходящего через эту точку, с кровлей и подошвой каждого геологического слоя. В силу описанного выше формата задания поверхностей границ слоев данная процедура является очень быстрой: определяется, в какой из квадратов равномерной прямоугольной сетки попадает точка и производится билинейная интерполяция.



Рис. 2. Триангуляция расчетной области со скважиной

После того как найдены точки пересечения с кровлей и подошвой, полученный отрезок разбивается в зависимости от того, на сколько сеточных подслоев пользователь желает разбить рассматриваемый геологический слой. Разбиение может быть как равномерным, так и сгущающимся к границам слоя. Поскольку слои могут вырождаться, предусмотрен ряд настроек построения сетки, позволяющих избежать построения слишком тонких ячеек, негодных для расчетов. Толщина слишком тонких ячеек может быть увеличена до заданной величины, при этом слои перестанут вырождаться, а все слои над вырожденным будут сдвинуты вверх на величину утолщения. Кроме того, слишком тонкие ячейки могут полностью вырождаться и исключаться из сетки, а слои над вырожденными сдвинутся вниз. Полученные точки разбиения отрезка будут являться вершинами призматической сетки, в генераторе сохраняются их *z*координаты.



Рис. 3. Триангуляция расчетной области с озером (a), призматическая сетка (b), гексаэдральная сетка (c)

Построенная таким образом сетка имеет строгую структуру. Это позволяет хранить, помимо двумерной триангуляции с заданными в треугольниках и на ребрах параметрами, только *z*-координаты вершин трехмерной сетки, что дает возможность генератору строить многомиллионные сетки даже на обычных персональных компьютерах. Кроме того, генерация может проводиться послойно с записью результатов в файл; в таком случае размер генерируемой сетки ограничен только возможностью построения двумерной триангуляции области.

После того как все *z*-координаты найдены, производится преобразование сетки из внутреннего формата в формат vtk. В окрестности выклинивания слоев призмы могут вырождаться в пирамиды и тетраэдры. Наконец, на основе данных, сохраненных в двумерной триангуляции, заполняются материалы ячеек, определяющие свойства среды, боковые граничные условия, а также граничные условия на кровле и подошве и теги, указывающие на наличие водоемов (рис. 3), и др.

Выбранный формат хранения границ геологических слоев позволяет сделать вывод о том, что сложность построения призматической сетки по найденной триангуляции имеет порядок $O(n_v)$, где n_v — число вершин призматической сетки.

2.2. Общая схема работы гексаэдрального генератора. В расчетном комплексе GeRa генератор сеток типа восьмеричное дерево со сколотыми ячейками представляет собой аппарат для построения многогранной сетки с преимущественно гексаэдральными ячейками для геологических областей, удовлетворяющей требованиям, описанным выше во введении. В частности:

- кровли и подошвы геологических слоев аппроксимируются гранями ячеек со вторым порядком точности;
- имеется возможность локального сгущения сетки;
- выклинивания пластов аппроксимируются гранями и ребрами сколотых ячеек;
- структурно сетка состоит из гексаэдров и многогранных ячеек, а общее число ячеек близко к числу вершин сетки;
- сетка адаптируется под границы геологических неоднородностей с помощью сколотых ячеек.

Использование сеток типа восьмеричное дерево позволяет строить гексаэдральные сетки, иерархически сгущающиеся к границам расчетной области, скважинам и другим объектам. Сколотая ячейка представляет собой часть гексаэдральной ячейки, полученную в результате ее среза поверхностной триангуляцией, и является многогранной ячейкой. Для построения сколотых ячеек используется алгоритм, основанный на методах Cubical marching squares [11] и Multiple material marching cubes [12]. Этот алгоритм подробно описан в работе [3].

В силу того, что геологические слои обычно имеют сильную вертикально-горизонтальную анизотропию, а кровли и подошвы слоев представляют собой сложные искривленные поверхности, допускающие пересечения, использование ортогональных гексаэдральных сеток становится весьма затруднительным. Для построения сеток в таких областях используется отображение ортогональных гексаэдральных сеток вдоль вертикальной оси. В результате получаются неортогональные анизотропные гексаэдральные ячейки (преимущественно). Грани, аппроксимирующие кровли и подошвы, могут стать неплоскими. При необходимости подобные грани могут быть разбиты на треугольники, однако для дискретизаций, используемых в GeRa, расчетные сетки могут иметь неплоские грани, поэтому они дополнительно не разбиваются.

Общая схема работы гексаэдрального генератора состоит из следующих этапов:

- получение входных данных из геологической модели;
- поиск набора отрезков пересечения геологических интерфейсов;
- формирование набора подобластей для каждого слоя из набора контуров; каждый набор состоит из входных данных геологической модели, а также набора контуров пересечений геологических интерфейсов;
- построение сетки типа восьмеричное дерево со сколотыми ячейками, представляющей собой набор из параллельных горизонтальных слоев для полученного набора подобластей;
- локальное измельчение гексаэдральной сетки;
- отображение сетки вдоль оси Oz в соответствии с координатами интерфейсов.

В качестве входных данных модели генератор использует: двумерные области, ограниченные контурами; геологические пласты, заданные интерфейсами; параметры скважин и точечных источников; области неоднородности. Каждый сеточный слой, соответствующий геологическому слою, может быть дополнительно разбит на подслои в количестве, указанном пользователем. Кроме того, пользователем задается параметр, определяющий средний размер ячейки. Большинство ячеек сетки будут иметь такой размер, остальные же будут дополнительно измельчены и будут иметь меньший размер.

Для гексаэдрального генератора также требуется подготовка данных, описанная в первом разделе. Для поиска линий пересечения геологических интерфейсов используется алгоритм 1. Эти линии необходимы генератору для того, чтобы можно было аппроксимировать линии выклинивания геологических слоев элементами сетки. В противном случае, после отображения сетки типа восьмеричное дерево вдоль оси Oz, ячейки, попавшие в область выклинивания, выродятся в ячейки с нарушенной топологией (т.е. могут иметь самопересечения, части нулевого объема и др.).

Для формирования набора подобластей используется алгоритм 3. Этот алгоритм применяется для каждого геологического слоя, так как линии выклиниваний слоев отличаются для разных слоев и соответственно получатся разные подобласти.

В результате исходная трехмерная область разбивается на непересекающиеся трехмерные подобласти. Теперь для каждой точки области можно определить, в какой подобласти она находится (или на границе каких подобластей). По этим данным может быть построена гексаэдральная сетка типа восьмеричное дерево со сколотыми ячейками в разнородной области (т.е. в области, разбитой на подобласти) [3]. Эта сетка автоматически локально измельчается для выполнения условия этого алгоритма: ребро гексаэдральной ячейки не может быть пересечено интерфейсами более, чем в одной точке. Кроме того, сетка дополнительно измельчается в соответствии с параметрами, заданными пользователем: измельчение к положению скважин, подземных объектов, контуров и др. Построенная сетка состоит из набора ровных горизонтальных слоев и интерфейсов, где каждый горизонтальный слой и интерфейс соответствует одному геологическому слою и интерфейсу исходной области. Эту сетку назовем *прообразом*.



Рис. 4. Прообраз сетки в двумерном случае (a) и отображенная сетка с выклиниванием (b)

Пусть расчетная область состоит из слоев \mathcal{L} и набора интерфейсов \mathcal{I} . Через $p\mathcal{L}$ и $p\mathcal{I}$ обозначим соответствующие элементы в сетке-прообразе. Набор граничных контуров подобластей обозначим через \mathcal{C} . Каждый слой $L_j \in \mathcal{L}$ находится между интерфейсами I_j и $I_{j+1} \in \mathcal{I}$. Рассмотрим двумерную область Γ на плоскости z = 0, представляющую собой объединение всех двумерных областей геологической модели. Каждый интерфейс $I_j \in \mathcal{I}$ может быть представлен в виде триангуляции, причем для любой точки $(x, y) \in \Gamma$ и вертикальной прямой, проходящей через эту точку, существует ровно одно пересечение с такой триангуляцией. Другими словами, для каждого интерфейса $I_j \in \mathcal{I}$ существует функция $fI_j : \Gamma \to \mathbb{R}$, однозначно определяющая z-координату соответствующей точки на интерфейсе: $\tilde{z} = fI_j(x, y)$ и $(x, y, \tilde{z}) \in I_j$. В сетке-прообразе каждый горизонтальный интерфейс pI_j лежит в плоскости $z = z_j$ для некоторого z_j . Для построения отображения сетки вдоль оси Oz для каждого слоя $pL_j \in p\mathcal{L}$ построим семейство аффинных отображений Φ_j , такое, что выполнены условия $\Phi_j(pI_j) = I_j$, $\Phi_j(pI_{j+1}) = I_{j+1}$. Элементами семейства Φ_j являются аффинные отображения ϕ_j^{xy} , переводящие отрезок $[(x, y, z_j), (x, y, z_{j+1})]$ в отрезок $[(x, y, \tilde{z}_j), (x, y, \tilde{z}_{j+1})]$. После применения таких семейств отображений, каждый слой сетки-прообраза, переходит в соответствующий слой итоговой сетки. При этом вершины сетки прообраза, которые соответствуют областям выклинивания слоев, начиная от линии выклинивания, после отображения совпадут с соседними вершинами по оси Oz. У всех ячеек, лежащих в области выклинивания (т.е. части слоя, имеющего нулевой объем), все вертикальные пары вершин совпадут, поэтому такие ячейки будут иметь нулевой объем. Эти ячейки могут быть легко удалены из сетки. У произвольной ячейки, лежащей по другую сторону от линии выклинивания, совпадут только те пары вершин, которые находятся на линии выклинивания. В результате общее количество вершин в ячейке станет меньше, но при этом такая ячейка не нарушит конформности сетки. Итоговая сетка с многогранными ячейками будет состоять преимущественно из гексаэдров. Для наглядности рассмотрим отображение сетки в двумерном случае (рис. 4). Сетка-прообраз состоит из трех слоев по 4 ячейки. Сколотые ячейки в среднем слое получены в результате разбиения области линией выклинивания (в двумерном случае — точкой). Различными цветами обозначены отдельные подобласти. После отображения голубая часть среднего слоя выродилась.



Рис. 5. Прообраз сетки в двумерном случае (a) и отображенная сетка с вывернутой ячейкой (b)

В случае сильно искривленных интерфейсов после отображения некоторые ячейки могут оказаться вывернутыми (рис. 5). Для исправления этого в генераторе предусмотрена постобработка ячеек, с помощью которой подобные ячейки дополнительно измельчаются.

Формальный алгоритм работы генератора может быть представлен в следующем виде.

Алгоритм 4. Алгоритм работы гексаэдрального генератора.

- 1: Получить входные данные: \mathcal{I}, \mathcal{C}
- 2: $nI = |\mathcal{I}|$
- 3: для i = 1, nI 1 начало цикла
- 4: Создать пустое множество отрезков C_i
- 5: Добавить элементы C в C_i
- 6: для j = i + 1, nI начало цикла
- 7: Найти C_{ij} пересечение интерфейсов I_i и I_j по алгоритму 1
- 8: Добавить C_{ij} во множество C_i
- 9: конец цикла
- 10: По множеству C_i найти подобласти A_i алгоритма 3
- 11: конец цикла
- 12: Построить сетку типа восьмеричное дерево со сколотыми ячейками из nI-1 горизонтальных слоев, разбитых на подобласти A_i
- 13: Разбить построенные слои на подслои в соответствии с параметрами, заданными пользователем
- 14: Измельчить сетку по заданным пользователем данным
- 15: для всех $V = (x_V, y_V, z_V)$ вершин сетки начало цикла
- 16: Найти pL_k , такой, что $V \in pL_k$
- 17: Построить аффинное отображение $\phi_k^{x_V y_V} \in \Phi_k$
- 18: Применить $\phi_k^{x_V y_V}(V)$
- 19: конец цикла
- 20: Удалить все ячейки нулевого размера

Проведем краткий анализ вычислительной сложности работы алгоритма 4, для этого оценим сложность его отдельных компонентов. Сложность алгоритма 1 пропорциональна $O(n_t^2)$, где n_t — число треугольников в триангуляции интерфейсов. Сложность алгоритма построения сетки типа восьмеричное дерево имеет порядок от $O(n_v)$ до $O(n_v \ln n_v)$ в зависимости от глубины дерева, где n_v — число вершин сетки [3]. Для отображения вершин сетки вдоль оси Oz требуется порядка $O(n_t n_v)$ операций. Таким образом, общую сложность алгоритма можно оценить как $O(n_t^2 + n_t n_v + n_v \ln n_v)$. При необходимости сложность алгоритма может быть уменьшена с помощью введения оптимальной структуры хранения триангуляций интерфейсов (например, kd-дерево).

Гексаэдральный генератор позволяет также учитывать наличие водоемов в модели. Водоем в модели представляет собой контур, который учитывается генератором на этапе алгоритма построения подобластей, однако этот контур учитывается только в верхнем геологическом интерфейсе. В результате верхние грани ячеек, пересекающихся с контуром водоема, разбиваются этим контуром на части. Для этого используется алгоритм Marching Squares, который лежит в основе алгоритма скалывания ячеек. Таким образом, часть контура водоема аппроксимируется полученными ребрами ячейки, а сама ячейка после этого имеет одну или несколько граней, аппроксимирующих поверхность водоема. Эти грани получают специальный тег водоема, который потом будет использован в дискретизации. Пример сетки с аппроксимированным контуром водоема изображен на рис. 3с.



Рис. 6. Сетка со сгущением к скважине (a) и срез сетки в области скважины (b)

Структура восьмеричного дерева позволяет легко измельчать гексаэдральную сетку локально. Это свойство используется при наличии скважин в модели. В расчетной области может быть задана одна или несколько скважин, для каждой из которых пользователем задается радиус. Эта величина (радиус) определяет уровень сгущения сетки к скважине как максимальный размер ячеек, в которые попадает фильтр скважины. Если радиус равен нулю, то сгущение проводиться не будет. Кроме координат (x, y), задающих положение скважины, генератор учитывает также положение фильтра скважины по оси Oz и сгущает сетку в соответствии с ним. Если координата (x, y) скважины совпадает с координатами грани или ребра сетки, то измельчаются одновременно все ячейки, разделяющие данный элемент сетки. В области может находиться произвольное количество скважин, для каждой из которых может производиться локальное сгущение сетки. На рис. 6 показана сетка с измельчением к скважине. Можно заметить, что измельчение производится не по всей оси Oz, а только в необходимой области, заданной положением фильтра скважины.

Гексаэдральный генератор позволяет учитывать поверхностные и подземные пустоты, а также подземные сооружения. В пустотах и подземных сооружениях сетку строить не нужно, однако сетка должна адаптироваться к их границе. Для генератора пустоты и подземные объекты представляют собой лишь подобласти с определенным индексом, поэтому построение сетки происходит без особых изменений. После того как сетка построена, из нее удаляются ячейки, принадлежащие подобласти с этим индексом. В результате границы пустот и подземных объектов аппроксимируются сколотыми ячейками сетки.

2.3. Численные эксперименты. Продемонстрируем работу генераторов на примере. Рассмотрим область, состоящую из трех геологических слоев, расположенных на высоте от 50 до 225 метров. Верхний геологический слой выклинивается. Триангуляции геологических интерфейсов имеют порядка 4 тысяч треугольников. Двумерная область ограничена контуром, изображенным на рис. 7а. Внутри области расположена скважина, фильтр которой расположен на высоте [150, 225] метров. Диаметр скважины — 5 метров. В каждом геологическом слое будем делать несколько сеточных подслоев, а именно 4, 4 и 3 начиная с нижнего слоя соответственно. Для данной области построим наборы сеток каждым генератором.



Рис. 7. Контур области со скважиной (a), пример гексаэдральной сетки при h = 40 (b)

Для построения сеток будем использовать персональный компьютер с 6 Гб оперативной памяти.

В табл. 1 приведены параметры сеток, построенных треугольно-призматическим генератором для данной модели. Через *h* обозначен максимальный шаг сетки, n_c, n_v — число ячеек и вершин сетки соответственно. В последних двух столбцах приводится время, затраченное генератором на построение сетки, и объем используемой памяти. Из таблицы можно заметить, что время построения сетки растет почти линейно.

Таблица 1

Параметры сеток, построенных треугольно-призматическим генератором, и время их построения

	Таблица	2
Параметры сеток, построенных		
гексаэдральным генератором, и		
время их построения		

h	n_c	n_v	Время, сек.	Память, Мб
40	25542	14688	0.99	105
20	93731	52620	3.33	388
10	367477	203424	16.5	1488
5	1560988	857340	58.65	4960

h	n_c	n_v	Время, сек.	Память, Мб
40	12715	14970	4.21	112
20	48022	54397	15.84	280
10	188 111	209337	57.39	874
5	745330	822236	227.40	3057

В табл. 2 приведены аналогичные параметры сеток, построенных гексаэдральным генератором для данной модели. Заметим, что для гексаэдральной сетки число вершин и число ячеек примерно эквивалентно. Из таблицы также можно заметить, что время построения сетки растет почти линейно. На рис. 7b изображена сетка при h = 40.

Заключение. Представленные результаты позволяют утверждать, что разработанные сеточные генераторы дают возможность строить адаптированные к положению различных гидрогеологических объектов сетки, состоящие из миллионов ячеек, на обычном персональном компьютере без применения параллелизации за приемлемое время. Разработанная методика обработки пользовательских данных, связанная с разбиением перекрывающихся областей на неперекрывающиеся подобласти, позволяет проводить адаптацию сеток к любым задаваемым пользователем двумерным объектам (областям и ломаным, включая самопересекающиеся). Тем не менее, подобная адаптация не всегда нужна, поскольку может приводить к чрезмерному измельчению сетки или, в случае сильно неравномерного разбиения границ объектов, к появлению ячеек "плохого" качества. Проблемы с качеством ячеек могут быть разрешены за счет ручного доразбиения границ объектов, но это может потребовать от пользователя серьезных усилий и приведет к измельчению сетки.

Кроме того, существует альтернативный подход, при котором сетка не адаптируется к части объектов. Это позволяет сократить число ячеек, но после того как сетка построена, возникает необходимость сохранить в ячейках сетки дополнительную информацию, характеризующую гидрогеологические объекты, не учтенные при построении сетки. В постобработке оценивается и сохраняется на сетке вклад пользовательского объекта в ячейку или грань. Так, для неоднородности, задаваемой в некотором слое, вычисляется, какая часть объема ячейки попадает внутрь области неоднородности. В итоге ячейке присваивается не один материал, характеризующий свойства среды, а список материалов и список соответствующих объемов. Аналогично для водоема: на поверхностной грани задаются списки попадающих в нее водоемов и площадей. Кроме неоднородностей и водоемов, к объектам, для которых можно проводить не только адаптацию сетки, но и учитывать их в постобработке, относятся реки и дренажи.

Работа была частично поддержана проектом "Прорыв" ГК Росатом и грантом РФФИ (код проекта 15–35–20991 мол а вед).

СПИСОК ЛИТЕРАТУРЫ

- 1. Капырин И.В., Уткин С.С., Василевский Ю.В. Концепция разработки и использования расчетного комплекса GeRa для обоснования безопасности пунктов захоронения радиоактивных отходов // Вопросы атомной науки и техники. Серия "Математическое моделирование физических процессов". 2014. № 4. 44–54.
- Василевский Ю., Вершинин А., Данилов А., Пленкин А. Технология построения тетраэдральных сеток для областей, заданных в САПР // Матричные методы и технологии решения больших задач. М.: ИВМ РАН, 2005. 21–32.
- Чернышенко А.Ю. Построение сеток типа восьмеричное дерево со сколотыми ячейками в неоднородных областях // Вычислительные методы и программирование. 2013. 14. 229–245.
- 4. Сидоров М.Л., Пронин В.А. Неструктурированная призматическая дискретизация сложных геологических структур в параллельном режиме // Вопросы атомной науки и техники. Серия "Математическое моделирование физических процессов". 2015. № 1. 47–55.
- 5. Diersch H.J.G. FEFLOW finite element subsurface flow and transport simulation system. Berlin: WASY GmbH, 2005.
- 6. Panday S., Langevin C.D., Niswonger R.G., Ibaraki M., Hughes J.D. MODFLOW-USG version 1: an unstructured grid version of MODFLOW for simulating groundwater flow and tightly coupled processes using a control volume finite-difference formulation: U.S. Geological Survey Techniques and Methods (http://water.usgs.gov/ogw/mfusg/#more).
- 7. George P.L. Automatic mesh generation and finite element computation // Handbook Numer. Anal. 1996. 4. 127–148.
- Данилов А.А. Технология построения неструктурированных тетраэдральных сеток // Ж. вычисл. матем. и матем. физ. 2010. 50, No 1. 146–163.
- Freedman V.L., Chen X., Finsterle S., et al. A high-performance workflow system for subsurface simulation // Environmental Modelling & Software. 2014. 55. 176–189.
- 10. Livnat Y., Shen H.-W., Johnson C.R. A near optimal isosurface extraction algorithm using the span space // IEEE Trans. Vis. Comp. Graphics. 1996. 2, N 1. 73–84.
- 11. Ho C.-C., Wu F.-C., Chen B.-Y., Chuang Y.-Y., Ouhyoung M. Cubical marching squares: adaptive feature preserving surface extraction from volume data // Comput. Graphics Forum 2005. 24, N 3, 537–545.
- Wu Z., Sullivan J.M. Multiple material marching cubes algorithm // Int. J. Numer. Meth. Eng. 2003. 58, N 2. 189–207.
- Yahiaoui B., Borouchaki H., Benali A. Hex-dominant mesh improving quality to tracking hydrocarbons in dynamic basins // Oil & Gas Science Technology. 2014. 69, N 4. 565–572.
- 14.Электронный ресурс: генератор HEXPRESS компании Numeca (http://www.numeca.com/).
- 15.Электронный pecypc: MeshGems: Suite of Reliable Meshing Software Components for High-Demanding CAD/CAE Developers (http://meshgems.com/volume-meshing-meshgems-hexa.html).
- 16.Электронный pecypc: Hexotic CAD/CAE Developers (http://www-roc.inria.fr/gamma/gamma/Membres/CIPD/Loic.Marechal/Research/Hexotic.html).
- 17. Электронный ресурс: ANSYS ICEM CFD

(http://www.ansys.com/Products/Other+Products/ANSYS+ICEM+CFD).

Поступила в редакцию 07.09.2015

Adaptive Unstructured Mesh Generation Methods for Hydrogeological Problems

A. V. Plenkin¹, A. Yu. Chernyshenko², V. N. Chugunov³, and I. V. Kapyrin⁴

¹ Nuclear Safety Institute, Russian Academy of Sciences; ulitsa Bol'shaya Tul'skaya 52, Moscow, 115191, Russia; Ph.D., Scientist, e-mail: golden_dragon_84@mail.ru

- ² Institute of Numerical Mathematics, Russian Academy of Sciences; ulitsa Gubkina 8, Moscow, 119333, Russia; Ph.D., Scientist, e-mail: chernyshenko.a@gmail.com
- ³ Institute of Numerical Mathematics, Russian Academy of Sciences; ulitsa Gubkina 8, Moscow, 119333, Russia; Dr. Sci., Leading Scientist, e-mail: chugunov.vadim@gmail.com
- ⁴ Nuclear Safety Institute, Russian Academy of Sciences; ulitsa Bol'shaya Tul'skaya 52, Moscow, 115191, Russia; Ph.D., Head of Laboratory, e-mail: ivan.kapyrin@gmail.com

Received September 7, 2015

Abstract: Two mesh generators embedded in the GeRa code for the solution of groundwater flow and radionuclide transport problems are discussed. The first one is a triangular-prismatic generator with the ability of cell degeneration; the second one is the polyhedral octree mesh generator implementing the cut-cell technology. These generators allow the user to automatically create conformal polyhedral adaptive grids in three-dimensional geological domains with consideration of complex boundaries, top and bottom geological layer surfaces, pinchouts, and geological heterogeneities.

Keywords: mesh generator, hydrogeological modeling, triangular-prismatic meshes, octree meshes.

References

1. I. V. Kapyrin, S. S. Utkin, and Yu. V. Vasьrilevskii, "Concept of the Design and Application of the GeRa Numerical Code for Radioactive Waste Disposal Safety Assessment," Voprosy Atom. Nauki Tekh. Ser.: Math. Model. Phys. Proc., No. 4, 44–54 (2014).

2. Yu. Vasilevskii, A. Vershinin, A. Danilov, and A. Plenkin, "A Technology of Constructing Tetrahedral Meshes for Domains Given in a CAD System," in *Matrix Methods and Technologies for Solving Large Problems* (Inst. Numer. Math. Russ. Acad. Sci., Moscow, 2005), pp. 21–32.

3. A. Yu. Chernyshenko, "Generation of Octree Meshes with Cut Cells in Multiple Material Domains," Vychisl. Metody Programm. 14, 229–245 (2013).

4. M. L. Sidorov and V. A. Pronin, "Unstructured Prismatic Discretization of Complex Geological Structures in the Parallel Mode," Voprosy Atom. Nauki Tekh. Ser.: Math. Model. Phys. Proc., No. 1, 47–55 (2015).

5. H.-J. G. Diersch, *FEFLOW Finite Element Subsurface Flow and Transport Simulation System* (WASY GmbH, Berlin, 2005).

6. S. Panday, C. D. Langevin, R. G. Niswonger, et al., MODFLOW-USG Version 1: An Unstructured Grid Version of MODFLOW for Simulating Groundwater Flow and Tightly Coupled Processes Using a Control Volume Finite-Difference Formulation, U.S. Geological Survey Techniques and Methods.

 $http://water.usgs.gov/ogw/mfusg/\#more.\ Cited\ October\ 5,\ 2015.$

7. P. L. George, "Automatic Mesh Generation and Finite Element Computation," Handb. Numer. Anal. 4, 127–148 (1996).

8. A. A. Danilov, "Unstructured Tetrahedral Mesh Generation Technology," Zh. Vychisl. Mat. Mat. Fiz. 50 (1), 146–163 (2010) [Comput. Math. Math. Phys. 50 (1), 139–156 (2010)].

9. V. L. Freedman, X. Chen, S. Finsterle, et al., "A High-Performance Workflow System for Subsurface Simulation," Environ. Model. Softw. 55, 176–189 (2014).

10. Y. Livnat, H.-W. Shen, and C. R. Johnson, "A Near Optimal Isosurface Extraction Algorithm Using the Span Space," IEEE Trans. Vis. Comput. Graphics 2 (1), 73–84 (1996).

11. C.-C. Ho, F.-C. Wu, B.-Y. Chen, et al., "Cubical Marching Squares: Adaptive Feature Preserving Surface Extraction from Volume Data," Comput. Graphics Forum **24** (3), 537–545 (2005).

12. Z. Wu and J. M. Sullivan, "Multiple Material Marching Cubes Algorithm," Int. J. Numer. Meth. Eng. 58 (2), 189–207 (2003).

13. B. Yahiaoui, H. Borouchaki, and A. Benali, "Hex-Dominant Mesh Improving Quality to Tracking Hydrocarbons in Dynamic Basins," Oil Gas Sci. Technol. **69** (4), 565–572 (2014).

14. HEXPRESS: Unstructured Full-Hexahedral Meshing. http://www.numeca.com/. Cited October 5, 2015.

15. MeshGems: Suite of Reliable Meshing Software Components for High-Demanding CAD/CAE Developers. http://meshgems.com/volume-meshing-meshgems-hexa.html. Cited October 5, 2015.

16. Hexotic CAD/CAE Developers.

http://www-roc.inria.fr/gamma/gamma/Membres/CIPD/Loic.Marechal/Research/Hexotic.html. Cited October 5, 2015.

 $17. \ ANSYS \ ICEM \ CFD. \ http://www.ansys.com/Products/Other+Products/ANSYS+ICEM+CFD. \ Cited \ October \ 5, \ 2015.$