

УДК 004.9:504:519.6

АЛГОРИТМ ПОСТРОЕНИЯ ЛИНИЙ УРОВНЯ В СИСТЕМЕ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ПРИ РАДИАЦИОННЫХ АВАРИЯХ РОДОС

Е.А. Евдин

Институт проблем математических машин и систем НАН Украины

e-mail: yewgen@env.com.ua

ВВЕДЕНИЕ

Европейская система поддержки принятия решений при радиационных авариях РОДОС разрабатывается с 1992 в рамках научных программ Европейской Комиссии [1]. Основное назначение системы – прогнозирование последствий потенциальных аварий на АЭС и планирование неотложных и долгосрочных контрмер. РОДОС – это современный программный комплекс, сочетающий математические модели, базы данных, ГИС функциональность и удобный пользовательский интерфейс [2–3]. Геоинформационный модуль системы обеспечивает базовую ГИС функциональность, используется для удобного представления результатов расчета пользователю, а также служит для сбора некоторых входных данных моделей [4].

Важной функциональностью ГИС модуля является возможность построения и отображения на карте линий уровня (изолиний) по сеточным данным как для рассчитанных с помощью математических моделей прогнозных величин (например, полей концентраций, доз), так и для метеорологических полей. В статье представлен алгоритм построения изолиний, а также приведены примеры его работы в СППР РОДОС.

РАВНОМЕРНАЯ СЕТКА

Опишем алгоритм построения линий уровня для величин заданных на прямоугольной сетке. При этом будем предполагать, что данные в ячейках сетки представляют собою значения в центрах ячеек. Это утверждение верно для расчетных и метеорологических полей, используемых в системе РОДОС.

Пусть c – значение линия уровня, которую необходимо построить. Для начала разделим сетку на ячейки, значения в которых больше или равны c и ячейки, значения в которых меньше c . Для этого поочередно будем перебирать грани ячеек находящихся на границе области. При этом будем рассматривать только те грани, которые выходят из границы в середину области. На рис. 1 граничные ячейки обозначены серым цветом, а интересующие нас грани выделены жирным.

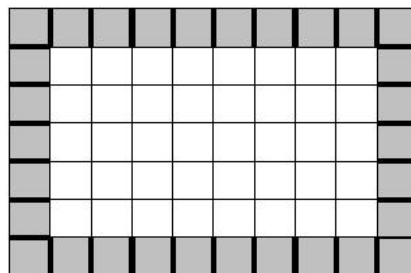


Рис. 1. Граничные ячейки (обозначенные серым) и рассматриваемые грани (жирные)

Последовательно перебирая грани, находим такую, которая будет разделять ячейки со значениями, находящимися по разные стороны числовой оси относительно c (то есть значение в одной ячейке будет меньше, а во второй больше или равно c). Эта грань будет принадлежать границе, разделяющей ячейки сетки относительно c . Запоминаем эту грань, как уже использованную. Для нахождения следующей грани рассмотрим четыре смежных

ячейки (рис. 2, рассмотренная грань выделена жирным, стрелкой обозначено направление движения по граням ячеек при построении границы).

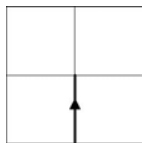


Рис. 2. Смежные ячейки, используемые для нахождения следующей грани

Существуют четыре варианта расположения значений в этих ячейках относительно s . В каждом из вариантов, новая грань, принадлежащая искомой границе, определяется однозначно по правилу, приведённому на рис 3 (ячейки со значением больше или равным s обозначены серым). Новую грань также запоминаем, как уже использованную. В случае рис. 3.4 можно выбирать направление, как варианте 3.1. Главное применять только одно правило во всех подобных случаях.

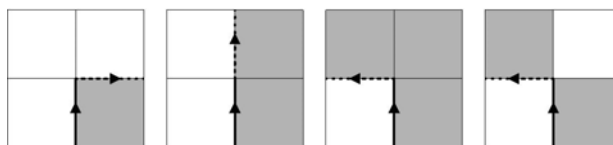


Рис. 3. Выбор новой грани (пунктирная стрелка) при различных вариантах расположения ячеек меньших (белые) и больше или равных s (серые) относительно последней использованной грани (жирная стрелка)

Описанный процесс повторяем для новой грани, поворачивая или отображая симметрично правило, описанное на рис. 3, в зависимости от направления грани и значений в ячейках. Работа этого этапа алгоритма закончится, когда новая грань будет принадлежать множеству граней, обозначенных на рис 1 жирным. Далее рассматриваем следующую грань из выделенных на рис 1 и неиспользованных ранее. В случае если она разделяет ячейки со значениями меньше и больше или равно s , применяем к ней описанный выше алгоритм. Таким образом, перебрав все грани, мы найдём незамкнутые границы, разделяющие области со значениями ячеек больше и меньше s .

Для нахождения замкнутых границ рассмотрим все вертикальные грани, которые не примыкают к границе сетки (рис. 4). Будем поочерёдно искать грань, которая ещё не была использована и разделяет ячейки со значениями меньше и больше или равно s . Для каждой такой грани повторим выше описанный алгоритм построения границы. Работа этого этапа алгоритма завершится тогда, когда новой гранью окажется уже использованная грань, то есть, граница замкнётся. После этого переходим к следующей вертикальной неиспользованной грани.

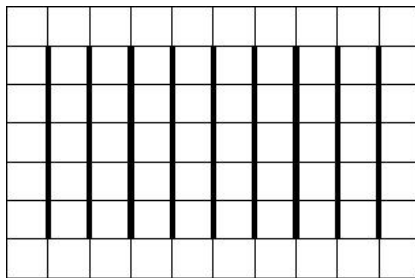


Рис. 4. Вертикальные грани, не примыкающие к границе сетки

Дополнительно рассматривать горизонтальные, не примыкающие к границе области грани не нужно, так как они все окажутся использованными после первых двух

этапов алгоритма. Таким образом, мы найдём все (замкнутые и незамкнутые) границы, разделяющие области со значениями ячеек меньше и больше или равно c .

Каждая из границ определяет одну изолинию. Грани, входящие в границу, принадлежат двум смежным ячейкам, значения в которых находятся по разные стороны на числовой оси относительно c . Так как мы предполагаем, что значения в ячейках соответствуют значению в центрах ячеек, то проведя линейную интерполяцию на отрезке, соединяющем центры эти смежных ячеек, найдём точку, значение в которой равно c . Назовём эту точку опорной. Каждая грань найденной границы определяет одну опорную точку. Пройдя по всем граням одной границы, в том порядке, в котором они находились, получим набор опорных точек. Соединяя в том же порядке опорные точки между собой, получим линию уровня (рис. 5).

В приведённом выше алгоритме, каждая грань сетки рассматривается не более двух раз (при построении границы и при отказе использовать грань в качестве начальной для границы) и, таким образом, сложность алгоритма оценивается в $O(nm)$, где n, m – количество столбцов и строчек сетки. Заметим, что строить изолинии (точнее опорные точки) можно за один проход с построением границ, разделяющих сетку на области меньше и больше или равно c , что сократит время выполнения алгоритма.

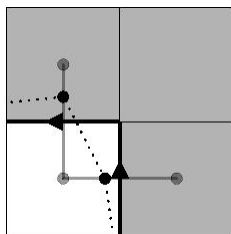


Рис. 5. Построение опорных точек (чёрные точки) и линии уровня (пунктирная линия), серые точки – центры ячеек

ВЛОЖЕННАЯ СЕТКА

Хотя метеорологические поля и некоторые модели, интегрированные в РОДОС, используют прямоугольные сетки (MATCH, ERMIN, IAMM), большинство моделей использует конфигурацию из вложенных сеток (рис. 6). Для того чтобы построить линии уровня, необходимо создать равномерную сетку, покрывающую всю расчетную область, а после строить изолинии по этой сетке.

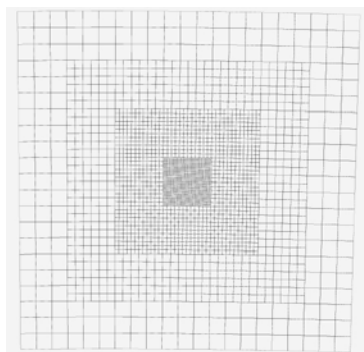


Рис. 6. Основная сетка расчетных моделей, интегрированных в СППР РОДОС

В общем случае для определения значений в ячейках равномерной сетки необходимо провести наиболее подходящую интерполяцию по центрам исходных ячеек (например, построить триангуляцию области и интерполировать значения методом конечных элементов). Однако, исходя из специфики расчетной области и той её особенности, что грани ячеек параллельны осям координат (в той проекции, в которой они

создавались), этот процесс можно упростить. Шагом равномерной сетки можно выбрать сторону ячейки самой внутренней области. Значения в ячейках равномерной сетки, центры которых не совпадают с центрами изначальной расчетной сетки, определяются путём билинейной интерполяции по четырём ближайшим центрам ячеек оригинальной сетки, образующим четырёхугольник, внутри которого находится центр искомой вершины равномерной сетки. Вершины этого четырёхугольника можно найти как ближайшие центры ячеек исходной сетки в каждом из четырех квадрантов.

РЕАЛИЗАЦИЯ

Окно пользовательского интерфейса позволяющее генерировать линии уровня с определённым шагом, добавлять или удалять произвольные значения при необходимости, а также сохранять набор уровней в файл для использования его в будущем, представлено на рис. 7. Пример построенных изолиний для расчетного поля концентрации также показан на рис. 7.

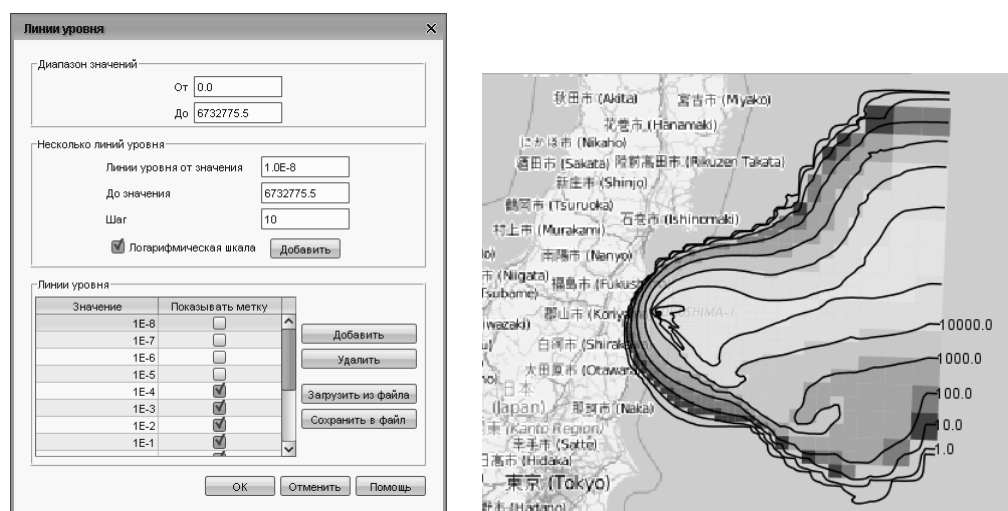


Рис. 7. Окно интерфейса пользователя (слева) и результат построения изолиний для поля концентрации Cs-137 (справа)

Функциональность ГИС модуля системы значительно облегчает анализ исходных данных и рассчитанных результатов и способствует процессу принятия решений. Реализация удовлетворяет всем требованиям, выдвинутым пользователями перед разработкой системы.

Литература

1. Raskob W. European approach to nuclear and radiological emergency management and rehabilitation strategies (EURANOS). *Kerntechnik* 72 (4) (2007) 172-175.
2. Ievdin Ie, Trybushnyi D., Zheleznyak M., Raskob W. RODOS re-engineering: aims and implementation details. *Radioprotection* 2010, Vol. 45, n° 5, p. 181–189.
3. Ievdin Ie, Treebushny D., Zheleznyak M.I. Distributed architecture of RODOS – Decision support system for nuclear emergency management in Europe, 3б. доповідей науково-практичної конференції з міжнародною участю “Системи підтримки прийняття рішень: теорія і практика. СППР-2009”, червень 2009 р., Київ. - С. 181-184.
4. Евдин Е., Трибушный Д., Железняк М. Разработка ГИС модуля Европейской системы поддержки принятия решений при радиационных авариях РОДОС. Ученые записки Таврического национального университета им. В.И. Вернадского. Серия: География. Том 23 (62). 2010г. №2. - С. 66-71.