

ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ ИССЛЕДОВАНИЯ ТРЕХМЕРНОГО ВЛАГОПЕРЕНОСА-ФИЛЬТРАЦИИ ЖИДКОСТЕЙ В МНОГОКОМПОНЕНТНЫХ СРЕДАХ НА PENTIUM-СКИТ

И.В. Сергиенко, В.С. Дейнека, В.В. Вещунов, М.В. Белоус

Институт кибернетики им. В.М. Глушкова НАН Украины
03680, Киев-187, проспект Академика Глушкова, 40,
тел.: 526 0637, e-mail: dejneka@public.icyb.kiev.ua

В работе описана информационная технология, предназначенная для исследования процессов диффузии и фильтрации жидкостей в трехмерных многокомпонентных грунтовых средах сложной внутренней структуры. В основу технологии положены математические модели и методы исследования процессов многокомпонентных тел с тонкими включениями и разработанные программно-алгоритмические составляющие комплекса Nadra3D, разработанные в Институте кибернетики им. В.М. Глушкова НАН Украины. Созданная автоматизированная система функционирует на вычислительном комплексе Pentium-SKIT. Рассмотрены ключевые решения, примененные при проектировании комплекса, и описаны основные используемые алгоритмы.

The information technology for research of diffusion and liquids filtration processes in 3D multicomponent ground medium that have difficult interior structure is described. The basis of technology are mathematical models, methods of research of processes in multicomponent bodies with thin inclusions and developed program-algorithmic constituents of the complex Nadra3D, developed in V.M. Glushkov Institute of Cybernetics of NASU. The created automated system functions on the computing complex Pentium-SKIT. The main decisions applied at planning of complex are considered, and the basic used algorithms are described

Введение

Процессы, протекающие в приповерхностных слоях земной коры, во многом определяют экологическую ситуацию региона. Изучение и прогнозирование возможных изменений состояний грунтовых сред и их последствий с помощью численного эксперимента является важной задачей. Для анализа процессов установившегося движения жидкостей в трехмерных многокомпонентных средах создана специализированная информационная технология. В основу технологии положены математические модели и методы исследования процессов многокомпонентных тел с тонкими включениями и разработанные программно-алгоритмические составляющие комплекса Nadra3D.

Автоматизированный программно-алгоритмический комплекс Nadra3D функционирует на настольных персональных компьютерах в операционных системах Windows 9x, NT, Me, 2000, XP и выполняет в интерактивном режиме подготовку геометрической области рассматриваемой задачи, прием и обработку физических данных составляющих среды, выполняет конечно-элементное разбиение сложных трехмерных объектов [1, 2], а также формирует систему уравнений метода конечных элементов (МКЭ) и находит ее решение [3, 4] для задач с относительно небольшим числом $p = n \cdot m$, где n – порядок системы уравнений, m – полуширина ленты ее ненулевых элементов. При больших p подсистема формирования и решения системы линейных алгебраических уравнений метода конечных элементов функционирует на многопроцессорных кластерах SKIT-2 и SKIT-3, созданных в Институте кибернетики им. В.М. Глушкова НАН Украины, и использует преимущества многопоточной обработки больших массивов данных [5, 6].

Программный комплекс Nadra 3D

Возможности комплекса. Программный комплекс Nadra 3D позволяет автоматизировать большинство этапов моделирования, подготовки данных, их корректировки и решения задач трехмерной фильтрации жидкостей. В число основных функций комплекса входят:

- создание и редактирование модели, описывающей геометрию рассматриваемой области;
- задание и корректировка физических характеристик среды;
- интерактивное визуальное и табличное представление геометрических и физических данных в составе модели;
- выполнение конечно-элементного разбиения геометрической модели с возможностью управляемых корректировок;
- формирование и решение систем алгебраических уравнений метода конечных элементов;
- визуальное и табличное представление решения в виде, облегчающем анализ рассматриваемых процессов.

В основу взаимодействия комплекса с пользователем положен интерактивный режим, когда изменения, производимые пользователем, непосредственно отображаются на экране монитора. Это позволяет детально контролировать все этапы проведения численного эксперимента.

Рабочее пространство комплекса представлено на рис. 1. Главное окно разбито на несколько зон,
© К. Georgiev, E. Donev, 2006

отличающихся функциональным назначением:

- 1) панели инструментов общего назначения, как, например, открытие или сохранение проекта;
- 2) контекстные панели инструментов, т.е. доступные в зависимости от вида или режима выполняемой операции, как, например, удаление зоны в режиме работы с отдельными зонами или группами зон;
- 3) окна просмотра геометрической модели в виде одной из 5 проекций;
- 4) контекстные информационные панели, которые содержат вспомогательную информацию о положении курсора и о выделенных частях модели.

Состав контекстных панелей инструментов гибко изменяется в соответствии с текущим режимом работы программы, таким образом, чтобы обеспечить пользователю наиболее легкое и прозрачное обращение к доступным функциям системы. Кроме этого, для облегчения работы начинающего пользователя, доступные функции визуализированы выделены на фоне недоступных в текущем контексте операций.

Окна просмотра могут представлять модель в виде плоских проекций на плоскости XY, XZ, YZ, трехмерной параллельной или трехмерной перспективной проекции. Геометрическая модель отображается в виде каркаса ребер, закрашенного объема или закрашенного объема с каркасом (рис. 2). Наличие в программе нескольких видов проекций и нескольких режимов отображения определено необходимостью облегчить работу пользователя при создании и редактировании сложных моделей.

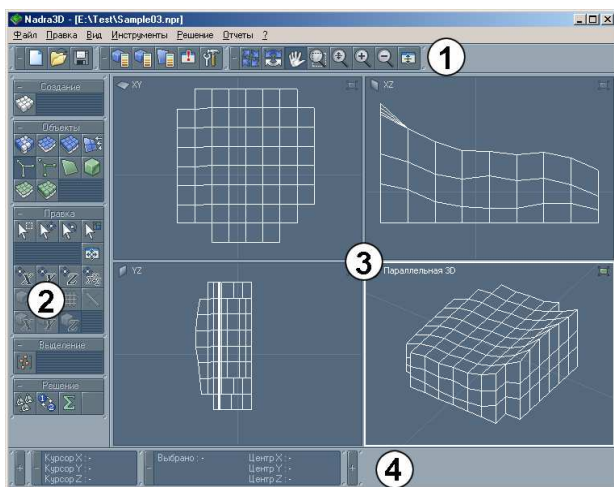


Рис. 1

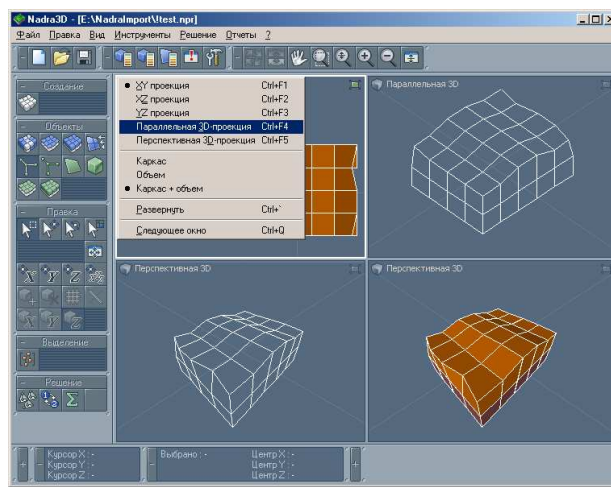


Рис. 2

При разработке комплекса Nadra 3D особое внимание уделено эффективности работы со встроенным инструментарием. Навигация, выбор и редактирование объектов в трехмерном случае является сложной задачей не только для начинающего, но и для искушенного пользователя. По этой причине в составе комплекса созданы специальные режимы, позволяющие оставить на экране монитора только часть большой модели, и работать только с ней, не отвлекаясь на поиск нужного элемента в большом потоке визуальной информации.

Технология описания сложных трехмерных объектов. В подавляющем большинстве случаев геометрические области реальных задач имеют сложную многослойную структуру со значительными различиями в свойствах отдельных участков среды, с наличием включений и других неоднородностей. С целью описания подобных областей разработано специализированное программно-алгоритмическое обеспечение. Оно позволяет представлять область в виде информационной модели и манипулировать данными, входящими в состав этой модели.

Выбор реализации информационной модели обусловлен применением для компьютерного анализа сложных процессов МКЭ. Геометрическая область аппроксимируется как дискретная сеточная модель [1], которая базируется на множестве кусочно-непрерывных функций, определенных в любом из участков модели. Физические параметры в одном геометрическом участке (зоне) считаются постоянными и могут изменяться на каждом шаге итерационного процесса или временного слоя. Применительно к задачам диффузии, массо- и теплопереноса, в их число входят коэффициенты фильтрации или теплопроводности в объеме элементарной геометрической формы, краевые условия I, II или III рода на его границах, условия сопряжения для тонких включений.

Основу технологии описания геометрических областей в программном комплексе составляет объектная модель, показанная на рис. 3. Составными элементами этой подсистемы являются объекты классов, предназначенные для решения узкоспециализированных задач манипулирования геометрическими и физическими данными. Они описывают элементы геометрической структуры: узлы, грани, ребра, прямоугольные зоны, горизонтальные и вертикальные слои. Каждый объект включает в себя помимо самих данных о геометрическом положении, размерах и формах, физических свойствах, также методы и процедуры работы с этими данными.

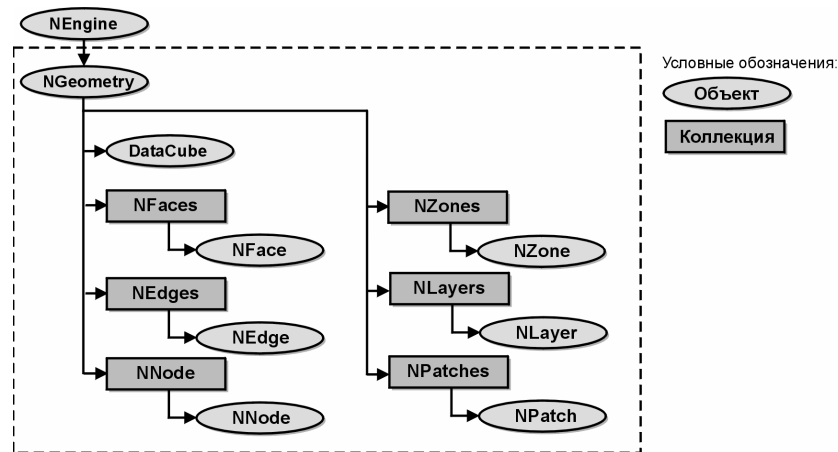


Рис. 3

Рассмотрим назначения и функции ее основных элементов. Объект *NEngine* – базовый объект ядра системы. Объект *NGeometry* описывает геометрию области в целом. Он имеет несколько подчиненных наборов элементов геометрии, таких как слои *NLayers*, границы разделения слоев *NPatches*, зоны *NZones*, стороны *NFaces*, ребра *NEdges* и узлы *NNodes*. Каждый из этих наборов содержит упорядоченную коллекцию однотипных элементов в виде двусвязных списков. К примеру, коллекция *NLayers* содержит все слои *NLayer*, которые присутствуют в геометрии области. Каждый список имеет параметры: *int Count* – количество элементов в списке, *NItem* First* – первый элемент в списке. Кроме того, списки имеют метод для произвольного доступа к элементу списка по его номеру *NItem* Item(int)*. Объектный класс *NItem* описывает поведение элемента двусвязного списка и является родителем, от которого наследуют свойства элементы вышеприведенных списков: свойство *NItem* Next* – следующий элемент списка и *NItem* Prev* – предыдущий элемент списка.

В составе подсистемы также присутствуют объекты, описывающие физические свойства модели, материалы *NMaterial* и типы граничных условий *NBorder*. Используя вышеописанный подход, эти объекты также выделены в отдельные списки *NMaterials* и *NBorders*. В зонах и гранях сохраняется только информация о принадлежности к определенному материалу. На рис. 4 показана объектная модель, описывающая физические свойства среды. Каждый материал или тип граничных условий имеет коллекцию свойств *NProperties*, в которых содержатся типизированные именованные параметры *NProperty*. Значения параметров *NProperty* могут соответствовать таким характеристикам среды, как коэффициентам фильтрации, проницаемости, теплопроводности, типам граничных условий и пр.

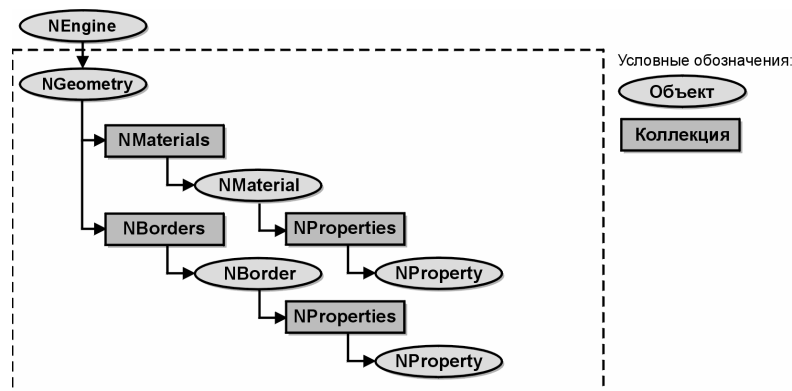


Рис. 4

Поддержание целостности данных. Важная роль в организации целостной геометрической и физической модели отводится структуре связей. В связанной среде помимо самих объектов геометрии, которые организованы в списки однотипных объектов, очевидно должна существовать структура описания связей одноранговых объектов, как, например, связи двух соседних зон, так и связей разноранговых объектов вида «примитивный элемент» – «обобщенный элемент» (узел-грань или ребро-прямоугольная зона). Эту роль выполняет трехмерный массив связей *DataCube* (рис. 3). В узлах массива содержатся указатели на объекты различных типов (рис. 5) в совокупности представляющих геометрию области. Определение связей в этом массиве осуществляется путем определения места положения значимых элементов связи. На рис. 5 элементам массива соответствуют узловые символы, обозначение символа соответствует типу объекта, указатель на который должен храниться в этом элементе массива: *n* – узел; *e* – ребро; *f* – грань; *z* – зона.

Полученная модель связей в геометрических областях позволила провести многоуровневую

оптимизацию. Во-первых, минимизирован объем занимаемой памяти, поскольку исключено значительное количество повторений связей по сравнению с другими методами представления. Во-вторых, ускорен доступ к произвольным данным по сравнению с одно- или двусвязными списками.

На основе массива связей объектная модель системы самостоятельно программно поддерживает целостность содержащихся в ней данных. Так удаление зоны приводит к удалению входящих в ее состав узлов, ребер и граней, но только в том случае если они не входят в состав других существующих зон.

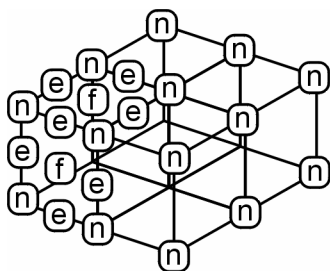


Рис. 5

Такая система описания трехмерных объектов является весьма эффективной в значительном количестве случаев применения. Она обладает хорошей скоростью и гибкостью доступа к данным с одной стороны, с другой – занимает минимальный необходимый объем памяти, с третьей – автоматически контролирует целостность модели.

Автодиагностика модели. Диагностика модели, которую реализует программа, позволяет контролировать корректность созданной геометрии области, а также обращает внимание пользователя системы на пропущенные участки при вводе физических характеристик среды. Операция проверки геометрии на корректность вызывается автоматически перед выполнением конечно-элементного разбиения. В составе этой операции выполняются проверки, являются ли грани выпуклыми многоугольниками, а также проверки на непересечение между гранями в составе зоны и отсутствие общего занимаемого объема для соседних зон. Участки, в которых были выявлены ошибки, выделяются мигающим красным цветом, а в окне сообщений выводится поясняющий появление ошибок текст. Геометрические коллизии должны быть исправлены пользователем системы, поскольку только в этом случае станут доступны операции конечно-элементного разбиения, формирования и решения системы уравнений МКЭ.

Как упоминалось выше, характеристики среды, определяющие физические свойства (материалы и типы граничных условий), заданы по умолчанию всем элементам области. Коэффициенты в составе материалов и типов граничных условий по умолчанию равны нулю и считаются не указанными. Наличие в составе модели подобных материалов не позволяет переходить к дальнейшим этапам решения задачи. Подобная проверка проводится перед выполнением конечно-элементного разбиения. В том случае, если найдены зоны или границы с не указанными коэффициентами, они выделяются мигающим красным цветом, а в окне сообщений выводится поясняющий текст. Всем указанным зонам необходимо назначить материалы с корректными коэффициентами фильтрации или теплопроводности, или, как частный случай, заполнить значения коэффициентов в материале по умолчанию. Подобные действия необходимо осуществить и по отношению к границам области и их свойствам (типам граничных условий).

Алгоритмы конечно-элементного разбиения. Поскольку в качестве математического аппарата для нахождения решения используется МКЭ, необходимо выполнить разбиение геометрической модели на конечные элементы, в качестве которых используются тетраэдры [2]. Для увеличения точности находимого решения в отдельных частях области возможно разбиение каждой из макрозон исходной области на $m \times n \times k$ элементов-микрозон, где m, n, k – количество разбиений по каждой из осей координат x, y, z , соответственно.

В случае наличия в области “инородных” тел – тонких включений используется модифицированный вариант алгоритма разбиения. Алгоритм разбиения области с учетом тонких включений позволяет учитывать разрыв сплошной среды в местах соединения произвольного количества включений. Для получения информации о совокупности разрывов среды в окрестности тонких включений используется алгоритм локализации включений.

В результате работы алгоритмов разбиения создается конечно-элементное разбиение, состоящее из конечных элементов МКЭ. На основании набора конечных элементов выполняется формирование матриц жесткости и правых частей метода конечных элементов. Однако, матрица жесткости при этом имеет вид разреженной матрицы размерностью $n \times n$ с большим количеством нулевых элементов. Решение системы уравнений в этом случае затруднено. Для уменьшения полуширины ленты ненулевых элементов матрицы МКЭ выполняется сортировка узлов конечно-элементного разбиения, таким образом, чтобы максимальное расстояние между номерами соседних узлов было как можно меньше. Сортировка основана на методе обратного упорядочивания графов Катхилла–Макки [7].

Прямой и обратный графический конвейеры. Основным инструментом визуализации геометрической модели выступают прямой и обратный графические конвейеры. Прямой конвейер выполняет проецирование и преобразование координат элементов геометрической модели из абсолютной системы координат модели в систему координат окна просмотра. При этом также выполняется отсечение элементов, вышедших за пределы окна просмотра. Обратный конвейер выполняет преобразование координат в окне просмотра и изменений координат, происходящих, как правило, в результате действий пользователя, в систему абсолютных координат модели.

Методы реалистичной визуализации геометрической модели. Получение реалистичного изображения геометрической модели подразумевает закрашивание внешней оболочки и учет влияния освещения. При расчете освещения в комплексе Nadra 3D учтены два типа отражений падающего света от поверхности: диффузное и зеркальное. Диффузное отражение описывает закон Ламберта [8], который определяет интенсивность отраженного света в зависимости от ориентации грани по отношению к источнику света. Для приближения к зеркальному отражению использована модель Блинна [8]. Перечисленные типы отражения скомбинированы с минимальным фоновым уровнем отраженного света с целью получить суммарное количество света, попадающего в глаз наблюдателя.

С проблемой визуализации покрашенных граней неразрывно связана проблема удаления невидимых поверхностей и невидимых ребер. Большинство объектов на сцене являются непрозрачными, вследствие чего не нужно рисовать объекты лежащие позади от видимых объектов. Для удаления невидимых поверхностей в комплексе Nadra 3D применен алгоритм с использованием деревьев двоичного разбиения пространства, как хорошо подходящий для работы с векторной графикой. Он основан на представлении совокупности внешних граней геометрической области в виде бинарного BSP-дерева [9].

Примененные алгоритмы в совокупности с графическим конвейером, выполняющим прямые и обратные преобразования из системы координат физического пространства модели на экран или принтер персонального компьютера, составляют основу визуализации геометрической модели в программном комплексе Nadra 3D.

Кластерная составляющая комплекса

После обработки программной составляющей ПЭВМ автоматизированной системы, сформированный файл данных пересылается на кластер. Управление процедурами обмена данными ПЭВМ-СКИТ осуществляется пользователем. Кроме того пользователь формирует и помещает в директорию с исполняемым файлом информационный файл `task.inf`, содержащий всю необходимую информацию для запуска системы, и запускает программу на выполнение. Информационный файл состоит из комментариев (строки, начинающиеся с символа #), содержащих справку о поддерживаемых режимах работы и некоторые заметки, полезные пользователю, и информационной части. Размер области комментариев не фиксирован и пользователь может при желании добавить в них интересующую его информацию. В информационной области необходимо заполнить следующие поля:

`mode` – целое значение, указывающее режим работы программы. На данном этапе поддерживается только режим счета;

`compare_with_etalon` – (значения по либо `yes`) – используется при тестировании системы на тестовых примерах с известным решением;

`block_size` – целое значение, указывающее размер рабочего блока для функции параллельного LDL^T – разложения матрицы МКЭ;

`task_type` – целое значение, указывающее тип рассчитываемой задачи. На данном этапе это может быть двумерная задача теории упругости, трехмерная задача фильтрации или трехмерная задача теории упругости;

`iteration` – указатель номера итерации в решении нелинейной задачи;

`file_data` – путь к файлу данных, сформированному интерфейсной частью, относительно исполняемого файла;

`file_out` – путь к файлу вывода относительно исполняемого файла;

`file_previous_solution` – путь к файлу, содержащему решение, полученное на предыдущей итерации. Это поле нужно заполнять при значении поля `iteration` > 1. В противном случае занесенное в него значение игнорируется;

`file_etalon` – путь к файлу, содержащему точное решение – используется при тестировании системы. Поле нужно заполнять при значении поля `compare_with_etalon` = `yes`. В противном случае занесенное в него значение игнорируется. Если поле `compare_with_etalon` имеет значение `yes`, а поле `file_etalon` не заполнено, проверка решения автоматически отключается.

После запуска программы на кластере управление передается системной компоненте, которая осуществляет разбор информационного файла, проверяет доступность указанных в нем файлов данных, их соответствие указанным параметрам и в случае успеха формирует соответствующий объект подсистемы формирования СЛАУ и передает ему управление.

Подсистема формирования СЛАУ имеет следующую архитектуру (рис. 6).

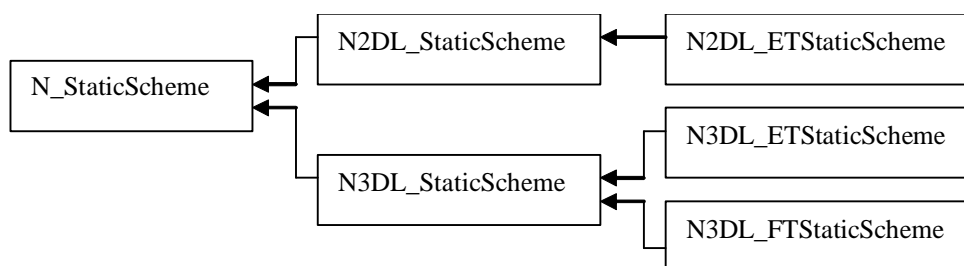


Рис. 6

Как видно из рис. 6 основу иерархии классов составляет абстрактный класс `N_StaticScheme`, реализующий некоторую обобщенную вычислительную схему для стационарной задачи. В этом классе реализуются базовые функции работы с файлами данных, а также общая для всех вычислительных схем последовательность вычислительного процесса, записанная в основном с помощью чисто виртуальных функций. Потомки базового класса являются конкретизациями этой схемы для двумерных и трехмерных стационарных задач. В них перегружается ряд функций, связанных с заполнением СЛАУ МКЭ и обработкой граничных условий. Фактически на этом уровне иерархии реализована основная работа системы. Классы последнего уровня представляют собой реализации конкретных вычислительных схем. В них определяются функции загрузки данных для конкретного типа схемы, функции вычисления элементарных матриц МКЭ и т.п.

Последовательность функционирования базового объекта подсистемы формирования СЛАУ, которую он проходит при получении управления, реализует следующие этапы.

1. Связывание с базовым объектом файла данных. Файл данных, обрабатываемый системой, имеет последовательно-кусочную структуру. Он представляет собой набор последовательно расположенных сегментов данных, каждый из которых начинается с заголовка, содержащего константу-идентификатор и размер этого сегмента. При разборе файла система считывает заголовок и в случае удачного распознавания идентификатора происходит соответствующая обработка содержащихся в сегменте данных, в случае неудачного – сегмент пропускается и система считывает следующий заголовок. Кроме того в начале файла записаны константы-метки, по которым система идентифицирует версию файла и принадлежность записанных в нем данных конкретному типу задачи. При несоответствии данных и запущенной вычислительной схемы генерируется ошибка и происходит завершение работы. При связывании файла с базовым объектом формируется его оглавление, позволяющее в дальнейшем осуществлять быстрый переход к сегментам данных, после чего происходит загрузка в память данных, необходимых для работы вычислительной схемы: параметров триангуляции, библиотек физических параметров, граничных условий, функций для вычисления правой части. Поскольку для различных вычислительных схем эти данные отличаются, функция привязки файла данных перегружается в объектах-потомках, реализующих конкретные схемы.

2. Связывание с базовым объектом файла вывода. Файл вывода имеет одинаковую структуру для всех задач. Он содержит набор констант-меток идентифицирующих задачу, некоторые дополнительные параметры и вектор решения. На этом этапе базовый объект записывает в файл вывода необходимые заголовки. Вектор решения записывается в файл вывода на этапе 4.

3. Связывание с базовым объектом файла, содержащего решение предыдущей итерации.

4. Запуск вычислительной схемы на счет. Получение решения базовым объектом состоит условно из следующих шести этапов:

4.1. Размещение СЛАУ в памяти. Имея размерность системы, полуширину ленты матрицы, размер рабочего блока и количество процессоров, выделенных для решения задачи, система определяет размер объема памяти, необходимого в каждом процессоре для хранения блоков матрицы и правой части, и пытается выделить эту память для использования. В случае успеха – переход к следующему этапу, иначе – генерация сообщения об ошибке и завершение работы.

4.2. Заполнение матрицы и вектора правой части. Матрица и вектор правой части распределены по процессорам по блочной столбцово-циклической схеме. В процессе заполнения СЛАУ каждый процессор считывает из файла данных блоки тетраэдров разбиения, выбирает из них те элементы, которые необходимы для формирования хранящейся в нем части системы и вычисляет соответствующие элементы матрицы и интегралы, необходимые для формирования правой части. Функция заполнения системы реализована таким образом, что работа с файлом, выборка необходимых элементов и взаимодействие с распределенной по процессорам системой осуществляется на уровне родительских классов, общих для ряда вычислительных схем, а вычисление интегралов, необходимых для формирования системы, осуществляется локально в классах-потомках, представляющих собой конечную реализацию конкретных схем.

4.3. Учет граничных условий второго и третьего рода. Идеология работы схожа с этапом 4.2.

Каждый процессор считывает блоки треугольников разбиения поверхности исследуемой области и на основании загруженной библиотеки граничных условий вычисляет необходимые поправки к матрице и вектору правой части.

4.4. Учет граничных условий первого рода. Из файла данных каждый процессор считывает набор точек, которые лежат на границе и в которых заданы условия первого рода, и на основании загруженной библиотеки граничных условий осуществляется пересчет элементов системы с целью удаления соответствующих загруженным узлам строк и столбцов.

4.5. Запуск программы Slae_bsb_sp, реализующей параллельный алгоритм решения СЛАУ с ленточной симметричной матрицей [5, 6].

4.6. Запись вектора решения в файл вывода.

После отработки вычислительной схемы управление возвращается системной компоненте, которая при необходимости осуществляет сверку полученного решения с точным решением из указанного пользователем файла, после чего уничтожает объект вычислительной схемы, закрывает используемые файлы и завершает работу.

После завершения работы программы пользователь может переслать файлы результатов на ПЭВМ для дальнейшего анализа полученного решения. При необходимости проведения еще одной итерации пользователю нужно подкорректировать информационный файл и снова запустить программу на выполнение.

Анализ результатов. На данном этапе подсистема анализа результатов, выполняемая на ПЭВМ, позволяет проводить сечение исследуемой области плоскостью в произвольном месте по произвольному направлению и отображать на срезе расцветочные изолинии интересующих пользователя параметров. В частности для задач фильтрации это пьезометрические напоры, высоты давления и значения коэффициентов фильтрации. Количество, значения и цвет изолиний выбираются пользователем произвольно.

Характеристики вычислительного процесса модельной задачи. Расчетная область с площадью ее верхней части – территории, составляющей примерно 40000 км.кв. и глубиной до 1 км. разбита на 2009664 пирамидальных элементов. Порядок расчетной системы нелинейных алгебраических уравнений составляет 359803 неизвестных. Полуширина ленты ее ненулевых элементов – 1726. Подготовка данных проводится на ПЭВМ PENTIUM. Время их передачи по сети internet на кластер составляет 65 мин. Для решения расчетной системы линейных алгебраических уравнений использовано 16 процессоров. Размеры рабочих блоков параллельного метода LDL^T разложения матрицы МКЭ – 10×10 .

Для решения указанной системы нелинейных алгебраических уравнений на каждом итерационном шаге метода последовательных приближений затрачено 49 мин. За четыре шага итерационного процесса решения системы нелинейных алгебраических уравнений МКЭ $A(X)X = B$, n – порядок этой системы, средняя

$$\text{погрешность} \frac{\sum_{i=1}^n |x_i^{j+1} - x_i^j|}{n} < \delta, \delta = 0.0097.$$

Выводы

В составе информационной технологии реализован инструментарий, позволяющий выполнить основные этапы подготовки и решения задач диффузии или фильтрации жидкостей в трехмерных средах. Программный комплекс Nadra 3D выполняет операции создания и редактирования геометрической и физической моделей среды, конечно-элементного разбиения с возможностью управления точностью разбиения, а, следовательно, и точностью решения, выполняет поиск решения и вывод полученных данных на печать. Подсистемы комплекса базируются на специально разработанных или широкоизвестных, но адаптированных для задач комплекса, алгоритмах. Подсистемы комплекса были подвергнуты многоуровневой оптимизации, которая позволила ускорить работу алгоритмов, уменьшить занимаемые машинные ресурсы и тем самым повысить порядок решаемых задач.

Для решения задач с большим порядком систем уравнений, которые не могут быть решены на ПК комплексом Nadra 3D, в силу физически ограниченных ресурсов персональных компьютеров, формирование и решение систем уравнений МКЭ выполняется специализированным математическим программным обеспечением, функционирующими на многопроцессорных комплексах СКИТ-1 и СКИТ-2, созданных в Институте кибернетики им. В.М. Глушкова НАН Украины. В этом случае с помощью комплекса Nadra 3D на ПК выполняется подготовка модели, конечно-элементное разбиение и вывод данных в файлы обмена, на основании которых в режиме параллельной многопроцессорной обработки данных формируются матрицы МКЭ и выполняется поиск решения системы уравнений. Данные о найденном решении могут быть импортированы в комплекс Nadra 3D для визуализации решения и формирования отчетов.

В настоящее время информационная технология применяется для решения практических задач. В то же время ее программно-алгоритмическая база продолжает совершенствоваться и развиваться в нескольких направлениях. Во-первых, это расширение классов решаемых задач на многопроцессорных кластерах СКИТ-1 и

СКИТ-2, а также на персональных компьютерах. Во-вторых, расширение функционального инструментария, целью которого является повышение удобства для пользователя при моделировании процессов в сложных областях, а также при анализе результатов расчетов.

1. *Молчанов И.Н., Николенко Л.Д.* Основы метода конечных элементов. – Киев: Наук. думка, 1989 – 270 с.
2. *Сегерлинд Л.* Применение метода конечных элементов. – М.: Мир, 1979. – 392 с.
3. *Дейнека В.С., Сергиенко И.В., Скопецкий В.В.* Математические модели и методы расчета задач с разрывными решениями. – Киев: Наук. думка, 1995. – 262 с.
4. *Дейнека В.С., Сергиенко И.В.* Модели и методы решения задач в неоднородных средах. – Киев: Наук. думка, 2001. – 606 с.
5. *Зенкевич О.* Метод конечных элементов в технике. – М.: Мир, 1975. – 540 с.
6. *Хилл Ф., Ореп GL.* Программирование компьютерной графики. – С.-П.: Питер, 2002. – 1082 с.
7. *Кнут Д.* Искусство программирования. т 1. Основные алгоритмы, 3-е изд. – М.: Издательский дом “Вильямс”, 2002. – 720 с.
8. *Исследование с помощью многопроцессорного вычислительного комплекса распределенных систем с большими объемами связанных данных / И.В. Сергиенко, В.С. Дейнека, А.Н. Химич, А.В. Попов, М.Ф. Яковлев, Т.В. Чистякова, М.В. Белоус* – Киев, 2005. – 33 с. – (Препр. /НАН Украины Институт кибернетики им. В.М. Глушкова; 2005 – 1).
9. *Попов А.В., Химич А.Н.* Параллельный алгоритм решения системы линейных алгебраических уравнений с ленточной симметричной матрицей // Компьютерная математика. – 2005. – № 2. – С. 52–59.