

ПРИМЕНЕНИЕ ПРОГРАММНОЙ ПЛАТФОРМЫ INMOST В РАСЧЕТНОМ КОДЕ GERA ДЛЯ РАБОТЫ С РАСПРЕДЕЛЕННЫМИ СЕТКАМИ ОБЩЕГО ВИДА

Ф.В. ГРИГОРЬЕВ^{*}, И.В. КАПЫРИН^{†‡}, И.Н. КОНЬШИН^{†‡}

^{*} Московский физико-технический институт, Москва, Россия

[†] Институт вычислительной математики РАН, Москва, Россия

[‡] Институт проблем безопасного развития атомной энергетики РАН, Москва, Россия
Email: igor.konshin@gmail.com

Ключевые слова: Математическое моделирование, программная платформа, геомиграция радионуклидов, плотностная и тепловая конвекция

Аннотация. Программная платформа INMOST предоставляет пользователям средства хранения и выполнения различных операций для сеток и сеточных данных, сборки и решения линейных систем, и многие другие, как в последовательном, так и в параллельном режимах. Возможности программной платформы демонстрируются на примере разработки расчетного кода GeRa, предназначенного для моделирования процессов подземной фильтрации и переноса радионуклидов в геологических средах. В качестве примера реализации сложных гидрогеологических моделей в GeRa рассмотрено моделирование процессов плотностной и тепловой конвекции.

SOFTWARE PLATFORM INMOST IN THE GERA CODE TO OPERATE WITH THE DISTRIBUTED MESH DATA

F.V. GRIGORIEV^{*}, I.V. KAPYRIN^{†‡} AND I.N. KONSHIN^{†‡}

^{*} Moscow Institute of Physics and Technology, Moscow, Russia

[†] Institute of Numerical Mathematics of the Russian Academy of Sciences, Moscow, Russia

[‡] Nuclear Safety Institute of the Russian Academy of Sciences, Moscow, Russia
Email: igor.konshin@gmail.com

Summary. The INMOST software platform offers means for storage and various operations with meshes and mesh data, linear systems assembly, linear solvers and many others both in parallel and sequential modes. The advantages of the software platform are demonstrated on the design of the computational code GeRa, which is aimed at groundwater flow and radionuclides transport problems. As an example of advanced hydrogeological models

2010 Mathematics Subject Classification: 76E06, 86-08, 97N80.

Key words and Phrases: Numerical Modeling, Software Platform, Geomigration of Radionuclides, Density and Temperature driven Convection.

implementation the variable density and temperature driven convection problem is considered.

1 ВВЕДЕНИЕ

В работе описывается процесс проведения моделирования физических процессов, который распределен по этапам, в зависимости от того насколько общим или частным является его содержание. Выделяются наиболее общие, но повторяющиеся этапы, которые можно было бы объединить в некоторую программную платформу. Мы постараемся объединить в программную платформу как можно больше повторяющихся функций для облегчения создания новых расчетных кодов.

При проведении моделирования на компьютерах с распределенной памятью объем машинно-зависимого кода, отражающего взаимодействие разрабатываемого кода с компьютером, а также взаимодействие параллельных процессов между собой, существенно возрастает. Это означает, что в программную платформу при расчетах на распределенных сетках можно включить дополнительные функции. В работе рассматривается пример такой программной платформы INMOST, а также расчетного кода GeRa, созданного на ее основе. Обсуждаются взаимодействия различных модулей в рамках единого расчетного кода. Демонстрируются новые возможности расчетного кода GeRa в моделировании плотностной и тепловой конвекции.

В разделе 2 описаны этапы моделирования и показаны наиболее общие действия, которые могли бы быть объединены в программную платформу. В разделе 3 описаны возможности и особенности программной платформы INMOST. Раздел 4 посвящен описанию расчетного кода GeRa и месту в нем программной платформы INMOST. В разделе 5 рассматривается моделирование процессов плотностной и тепловой конвекции в пористых средах расчетным кодом GeRa.

2 ЭТАПЫ МОДЕЛИРОВАНИЯ

Рассмотрим этапы моделирования некоторой задачи математической физики, характеризующейся определенными физическими процессами и взаимодействием программных модулей, которые реализуют процесс моделирования. Нашей задачей будет выделение общих действий, которые можно было бы заранее в общем виде реализовать в некоторой программной платформе, что облегчило бы разработчикам работу по созданию нового программного кода.

Можно выделить следующие этапы моделирования:

- 1) Выбор физической модели;
- 2) Дискретизация области;
- 3) Задание начальных и граничных условий;
- 4) Дискретизация уравнений;
- 5) Решение линейных систем;
- 6) Анализ решения.

Рассмотрим каждый этап более подробно, отмечая общие и специфические черты моделирования, а также особенности реализации программного кода для них.

- 1) *Выбор физической модели* является наиболее специфическим этапом моделирования и полностью зависит от физической и математической постановки решаемой задачи. На этом этапе выбираются моделируемые

физические процессы и выписываются уравнения, описывающие их, которые предстоит решать на последующих этапах моделирования.

- 2) *Дискретизация расчетной области* состоит в ее задании и построении расчетной сетки в этой области. Только в самых простейших модельных случаях можно обойтись заданием ее вручную из программы, несколько более сложным и общим является задание области из графического интерфейса с помощью операций (пересечение/объединение) над аналитически заданными примитивами (куб, шар, тор и т.д.). В последнем случае можно было бы выделить некоторые общие принципы работы такого модуля, но предполагается, что мы ориентируемся на решение серьезных промышленных задач, в которых расчетные области могут быть очень сложными (например, расчет обтекания самолета) и задаваться в других программных средствах. При этом расчетная область, например, может импортироваться в расчетный код из программы AutoCAD с помощью программного средства OpenCascade, и программная платформа должна обеспечить такую возможность. Здесь проявляется специфика самой решаемой задачи. Однако после задания расчетной области необходимо провести построение расчетной сетки, которую можно, например, в готовом виде считать из внешнего файла, построить ее с помощью внешнего генератора сеток, или построить ее самим, исходя из специфики будущей дискретизации уравнений. Общие черты всех этих действий (считывание сетки из файла, обеспечение генератора сеток всеми необходимыми функциями построения сеток) могут быть реализованы в программной платформе.
- 3) *Задание начальных и граничных условий* – это этап, который сильно зависит от решаемой задачи, и может быть обобщен только для наиболее простых модельных задач. Для реально решаемой задачи может существовать очень много различных возможностей. Например, в полноценной задаче, решаемой с помощью расчетного кода GeRa, может быть задано 20 геологических слоев, имеющих по 20 параметров среды в каждом слое, 100 скважин со своими режимами работы, 100 поверхностных водоемов, 10 переносимых и взаимодействующих компонент (при моделировании задачи переноса и учета химии и радиоактивного распада). Программная платформа здесь должна обеспечить удобный механизм сопоставления сеточных элементов и заданных на них начальных и граничных условий. На этом этапе полностью проявляется специфика задачи пользователя, хотя, в принципе, в некоторых случаях можно было бы разработать модуль, который скрыл бы в себе специфику решаемой задачи и унифицировал бы интерфейс взаимодействия с модулем дискретизации. Для программного кода GeRa такой модуль «UserData» был создан, он позволил разрабатывать модуль дискретизации в более общем виде, пригодном для применения и в других программных продуктах.
- 4) Этап *дискретизации уравнений* вступает в действие, когда сами уравнения заданы (специфика задачи), выбран также способ дискретизации уравнений (зависимость не только от самих уравнений и специфики решаемой задачи, но и от предпочтений самого разработчика), и остается только выполнить саму дискретизацию. Здесь очень помог бы модуль, аналогичный описанному модулю «UserData», который обрабатывал бы заданные пользователем начальные и граничные данные и в унифицированном виде поставлял бы в дискретизацию

запрашиваемую информацию. Таким образом, программная платформа будет выдавать для дискретизации насчитываемые в процессе расчета данные и привязанную к расчетной сетке информацию. Помимо этого должны быть обеспечены удобные возможности для формирования матрицы линейной системы и соответствующей правой части. Кроме перечисленных функций этого этапа, можно отметить недавно появившуюся в программной платформе INMOST возможность проведения «автоматического» дифференцирования, обобщающего способ нахождения первых и вторых производных, и согласованного заполнения матрицы и правой части.

- 5) Этап *решения линейных систем* обычно является наиболее затратным по времени расчета. В первую очередь, именно ради уменьшения времени на этом этапе стали разрабатываться параллельные методы решения линейных систем. В настоящее время существуют общепризнанные библиотеки линейных решателей (например, PETSc^[7] и Trilinos^[8]), с которыми привыкли работать разработчики программных кодов. Кроме возможности воспользоваться этими решателями, программная платформа должна обеспечить пользователя возможностью удобного подключения своего собственного решателя, который, например, был разработан специально для конкретного приложения и учитывает специфику решаемой задачи. Все эти возможности программной платформой INMOST пользователю предоставляются, кроме того, можно воспользоваться и внутренними решателями самой программной платформы, которые в некоторых случаях превосходят общепризнанные параллельные решатели^[4].
- 6) На этапе *анализа полученного решения*, который без сомнения также является очень сильно зависимым от специфики решаемой задачи, мы предлагаем воспользоваться библиотекой VTK^[22] (при самостоятельной реализации модуля графического анализа решения) или готовыми внешними программами, например, ParaView^[23] и VisIt^[24]. INMOST предоставляет средства сохранения сеток и данных на них как в собственном формате, так и в формате VTK, удобном для визуализации.

Стоит отметить, что при разработке коммерческого расчетного кода также потребуется создание графического интерфейса пользователя для интерактивного задания параметров задачи, ввода начальных и граничных условий, управления последовательностью действий и этапов. Эта задача может быть решена, к примеру, с использованием библиотек Qt и интерфейса работы с ними Qt Creator^[25].

3 INMOST – ПРОГРАММНАЯ ПЛАТФОРМА

INMOST представляет собой программную платформу для разработки параллельных численных моделей на сетках общего вида^{[1]-[4]}. Главной задачей описываемой программной платформы является обеспечение пользователя всеми необходимыми средствами для создания и исследования различных численных моделей. Сюда входит не только работа с распределенными по процессорам сеточными данными для сеток общего вида, но и удобный интерфейс для формирования систем линейных уравнений, а также их решения, с помощью встроенных либо подключаемых пакетов линейной алгебры. Программная платформа INMOST разрабатывалась с учетом следующих критериев:

- богатый функционал возможностей;
- эффективность;
- надежность;
- универсальность;
- простота использования;
- открытость исходного кода.

В настоящее время очень трудно найти комплексы программ, удовлетворяющие сразу всем вышеперечисленным требованиям. Существующие решения, такие как, библиотека FMDB (Flexible distributed Mesh DataBase), библиотека MOAB (A Mesh-Oriented datABase), библиотека MSTK (MeSh ToolKit), библиотека STK (Sierra ToolKit), пакет Salome, пакет OpenFOAM (Open Source Field Operation And Manipulation CFD ToolBox) и другие обладают рядом недостатков и в полной мере не соответствуют заявленным критериям. У существующих решений не всегда есть легкая переносимость между различными платформами (Windows, Linux), существующие реализации не всегда надежны, в некоторых пакетах невозможно, либо проблематично внедрить свои схемы дискретизации. Также стоит отметить, что имеются общедоступные пакеты, в некоторый степени, удовлетворяющие выбранным критериям и имеющие богатый функционал, к таким пакетам можно отнести ParMETIS^[5] и Zoltan^[6], которые производят распределение и перераспределение данных по процессорам, а также пакеты PETSc^[7] и Trilinos^[8], применяемые для решения систем линейных уравнений.

На основании этого в ИВМ РАН было решено самостоятельно разработать набор программных средств, обеспечивающий необходимый функционал возможностей и допускающий подключение упомянутых пакетов ParMETIS и Zoltan, а также PETSc и Trilinos. В результате этого решения была создана программная платформа INMOST (Integrated Numerical Modelling and Object-oriented Supercomputing Technologies)^[1].

3.1 Модули платформы INMOST

Основные модули, входящие в состав программной платформы:

- модуль сеточных операций;
- модуль балансировки сетки;
- модуль автоматического дифференцирования;
- модуль составления и решения линейных систем.

Основная задача программной платформы заключается в поддержке сеточных операций на распределенных сетках, которые могут потребоваться различным генераторам расчетных сеток, а также различным способам дискретизации уравнений, описывающих выбранную физическую модель. Рассматриваются конформные расчетные сетки, ячейки которых состоят из произвольного количества граней, каждая из которых состоит из произвольного количества ребер. Программная платформа INMOST поддерживает ячейки произвольной конфигурации, при этом в нее не входят сами генераторы сеток из-за того, что тип сетки напрямую зависит от типа используемой дискретизации, которая, в свою очередь, определяется выбранной физической моделью и типом решаемой задачи.

Модуль балансировки сеток отвечает за автоматическое распределение расчетной сетки по процессорам и за эффективность и сбалансированность такого распределения. Когда расчетная сетка распределена по процессорам, то следующим шагом

производится определение фиктивных ячеек по границам подобластей, в которых обновление данных с соседних процессоров производится вызовом всего лишь одной функции INMOST из модуля сеточных операций. Стоит заметить, что количеством слоев фиктивных ячеек можно управлять, вплоть до ручного определения фиктивных ячеек в случае использования сложных и неоднородных шаблонов дискретизации. Заметим также, что даже само использование библиотеки обменов MPI скрыто от пользователя, для того чтобы, при необходимости, например, для отладки программ, легко можно было собрать последовательную однопроцессорную версию программной платформы INMOST.

Модуль автоматического дифференцирования призван помочь в записи конкретной дискретизации уравнений, заполнения матрицы якобиана и формировании правой части линейной системы. Этот модуль был разработан только в 2016 году и еще не получил широкого использования в других приложениях. Однако есть все основания ожидать, что он мог бы существенно упростить дискретизацию при введении в расчетный код GeRa новых моделей физических процессов, а также помочь в верификации уже внедренных дискретизации для работающих и опробованных моделей.

Модуль составления и решения систем линейных уравнений вместе с модулем сеточных операций является одним из самых важных модулей программной платформы INMOST. Отметим основные особенности этого модуля:

- удобное формирование матрицы линейной системы через адресацию по глобальным номерам строк и столбцом, аналогичную обращению к элементам плотной матрицы при сохранении всей эффективности работы с разреженной матрицей;
- возможность использования большого количества встроенных линейных решателей, а также линейных решателей из внешних пакетов, таких как PETSc и Trilinos;
- удобную возможность переключения линейных решателей без изменения процедуры формирования линейных систем.

3.2 Установка INMOST

Как уже было отмечено, INMOST разрабатывался с условием полной кроссплатформенности и переносимости между различными операционными системами (Windows, Linux, MacOSX). Чаще всего пользователи работают на персональных компьютерах в операционной системе Windows, а параллельные расчеты для наиболее сложных задач необходимо проводить на высокопроизводительных вычислительных кластерах, которые, как правило, находятся под управлением операционной системы Linux. Пользователю предоставляется возможность собирать INMOST непосредственно из исходных кодов, подробная документация и инструкции доступны на официальном сайте программной платформы INMOST^[2].

4 РАСЧЕТНЫЙ КОД GERA

4.1 Назначение и функциональные возможности

Расчетный код GeRa^[9] разрабатывается в рамках подпроекта «Коды нового

поколения» проекта «Прорыв» Госкорпорации «Росатом» и предназначен для решения задач оценки безопасности захоронения радиоактивных отходов с точки зрения потенциальной миграции радионуклидов в подземных водах. Код является интегральным, то есть, позволяет решить задачу оценки безопасности объекта полностью^[10]: создать геологическую модель площадки с помощью встроенных средств интерполяции и геостатистики; построить расчетную сетку; провести расчет фильтрационной и миграционной задач; визуализировать результаты и получить дозы для населения при водопользовании, что и является необходимым результатом оценки безопасности. Отличительными особенностями кода являются использование трехмерных неструктурированных конформных сеток, состоящих из многогранников^[12], методов конечных объемов для пространственной дискретизации задач, ориентация на использование как в последовательном режиме на персональных компьютерах, так и в параллельном режиме на удаленных кластерах. Параллелизация кода реализована с помощью MPI.

К настоящему моменту разработана первая версия кода GeRa (GeRa V.1), которая подана на аттестацию в Ростехнадзор. В ней реализованы модели следующих процессов:

- течение грунтовых вод (фильтрация) в напорной и безнапорной постановках;
- насыщенно-ненасыщенная фильтрация;
- адвективно-дисперсионно-диффузионный перенос с возможностью учета сорбции, описываемой изотермой;
- радиоактивный распад переносимых радионуклидов с расчетом цепочек превращений;
- химические взаимодействия в системе вода-порода путем включения в GeRa геохимического модуля iPHREEQC^{[13],[11]}.
- плотностная конвекция.

Для организации хранения сеток и данных на сетках, а также сборки и решения систем линейных алгебраических уравнений в параллельном и последовательном режимах в GeRa V.1 использована программная платформа MSPP собственной разработки. Платформа MSPP была создана в 2012 году, и фактически является предыдущим поколением по отношению к INMOST. Вопрос целесообразности перехода с MSPP на INMOST будет рассмотрен далее путем сравнения их производительности на характерных примерах использования платформы в GeRa (см. раздел 4.4).

4.2 Перспективы развития кода

На основе опыта использования кода GeRa на практике и анализа текущих разработок других гидрогеологических расчетных кодов^{[14],[15]} определено два основных направления дальнейшей разработки (второй версии кода, GeRa V.2). Во-первых, это усовершенствование программной платформы и внедрение новых численных методов для повышения эффективности работы кода. В частности:

- Переход на платформу INMOST для ускорения операций с сеточными элементами (ячейками, гранями, вершинами) и данными на сетках.
- Повышение эффективности решателей разреженных линейных систем.
- Разработка эффективных методов решения нелинейных задач, возникающих, к

примеру, при дискретизации задач ненасыщенной и двухфазной фильтрации.

– Параллелизация всех реализованных моделей.

Во-вторых, это расширение спектра моделируемых процессов, а именно:

- Двухфазные течения вода-газ.
- Перенос в трещиноватых средах.
- Теплоперенос, тепловая конвекция и тепловыделение вследствие радиоактивного распада.
- Фильтрация растворов переменной вязкости.
- Коллоидный перенос.
- Геомеханические и биологические процессы.

Помимо этого, предполагается интеграция кода с модулями моделирования ближней зоны захоронения, расчета дозовых нагрузок, оценки неопределенностей. В настоящей работе демонстрируется новый функционал GeRa по совместному моделированию процессов тепловой и плотностной конвекции.

4.3 Модульная структура кода

Расчетный код GeRa имеет модульную структуру, и его структура полностью опирается на программную платформу (MSPP или INMOST):

- 1) Модуль построения геологической модели на основе исходных данных.
- 2) Модуль построения расчетных сеток.
- 3) Модуль дискретизации математических моделей.
- 4) Модуль расчета геохимических взаимодействий в системе вода-порода.
- 5) Параллельная программная MPI-платформа для работы с сетками и данными на сетках.
- 6) Модуль решения больших линейных систем.
- 7) Модуль программных средств хранения и обработки исходных данных, данных мониторинга, результатов расчетов и постпроцессинга по моделям.
- 8) Модуль калибровки моделей и оценки достоверности результатов.
- 9) Модуль расчета дозовых нагрузок для населения.
- 10) Графический интерфейс расчетного комплекса, интегрирующий все остальные модули и обеспечивающий средства визуализации.

Здесь модули 5 и 6, собственно, и составляют программную платформу, модули 2 и 3 существенно на нее опираются, модули 4, 8 и 10 активно ее используют. Таким образом, большая часть программных модулей используют программную платформу, она служит не только для сохранения сеточных данных, но и является средством взаимодействия программных модулей друг с другом.

4.4 Сравнительный анализ производительности платформ MSPP и INMOST

Для сравнения эффективности этих двух платформ между собой и со стандартными средствами C++ было проведено несколько тестов на прямоугольной сетке $100 \times 100 \times 50$ с общим числом граней равным 1520000. Тесты направлены на анализ производительности платформы на характерных для GeRa видах использования программной платформы: циклах по элементам сетки; доступ к данным на сетке; определение типа сеточного элемента с точки зрения принадлежности к процессору

(может быть Owned, Ghost, Shared).

Тест 1: сравнение скоростей работы итераторов

Сравнивались скорости работы Mesh::iteratorFace в платформах MSPP и INMOST, а также std::vector<Face*>::iterator – итератор по стандартному вектору STL (референтный). В циклах выполнялись простые действия, чтобы оптимизирующий компилятор их не проигнорировал. Ниже представлены участки программы, для которых проводились измерения.

Сеточный итератор (элемент платформы):

```
for (Mesh::iteratorFace iface = m->BeginFace();
     iface != m->EndFace(); iface++){
    a++;
    if (a%2) a += 2;
}
```

Итератор вектора:

```
for (vector<Face*>::iterator the_iterator = facePtr.begin(); the_iterator != facePtr.end();
     the_iterator++){
    b++;
    if (b%2) b += 2;
}
```

Результаты тестирования скорости работы итераторов приведены в таблице 1.

	MSPP	INMOST
Сеточный итератор	0.063042	0.024117
Итератор вектора	0.002215	0.002151
Соотношение	28.4608	11.207

Таблица 1 : Результаты тестирования скорости работы итераторов в MSPP и INMOST.

Тест 2: сравнение времени доступа к тегу и к значению вектора

Сравнивались время доступа к значениям тегов постоянного размера и время доступа к значениям элементов стандартного вектора. Тег в программных платформах MSPP и INMOST используется для привязки данных к элементам сетки. В качестве итератора в обоих случаях выступал vector<Face*>::iterator. Ниже представлены участки программы, для которых проводились измерения.

Тег:

```
for (vector<Face*>::iterator the_iterator = facePtr.begin(); the_iterator != facePtr.end();
     the_iterator++){
    uface = *the_iterator;
    s += uface -> Integer(probe); //Суммируем значения тега на гранях
}
```

Элемент вектора:

```
for (vector<int>::iterator the_iterator = faceInt.begin(); the_iterator != faceInt.end();
     the_iterator++){
```

```
s += faceInt[i++];
}
```

Сравнение времени доступа к тегу и к значению вектора для программных платформ MSPP и INMOST приведены в таблице 2.

	MSPP	INMOST
Тег	0.124058	0.0217268
Элемент вектора	0.004865	0.0049009
Соотношение	25.4996	4.43321

Таблица 2 : Сравнение скорости доступа к тегам и элементам вектора.

Тест 3: сравнение времени выполнения функции-маркера и времени доступа к bool значениям вектора

Сравнивалось время выполнения функции `Element::GetStatus` и время доступа к значениям элементов вектора типа `bool`. В качестве итератора в обоих случаях выступает `vector<Face*>::iterator`. Ниже представлены участки программы, для которых проводились измерения.

```
Функция-маркер (Element::GetStatus):
for (vector<Face*>::iterator the_iterator = facePtr.begin(); the_iterator != facePtr.end();
the_iterator++){
    uface = *the_iterator;
    boolTemp |= uface->GetStatus() != Element::Ghost;
}
Элемент вектора:
for (vector<bool>::iterator the_iterator = faceBool.begin(); the_iterator != faceBool.end();
the_iterator++){
    boolTemp |= faceBool[i++];
}
```

Сравнение времени выполнения функции-маркера и времени доступа к `bool` значениям вектора приведено в таблице 3.

	MSPP	INMOST
Функция-маркер	0.019155	0.00926089
Элемент вектора	0.002925	0.00297785
Соотношение	6.5486	3.10993

Таблица 3 : Сравнение времени выполнения функции-маркера и времени доступа к bool значениям вектора в MSPP и INMOST.

Во всех трех проведенных тестах программная платформа INMOST показала большую эффективность, чем используемая в настоящее время программная платформа MSPP, хотя и не такую высокую, как при использовании стандартных контейнеров STL. Однако в сравнении с STL использование платформы значительно упрощает процесс создания параллельных приложений и обеспечивает простые логические связи элементов и данных, а кроме этого не требуется выделять

дополнительную память на хранение контейнеров STL.

5 МОДЕЛИРОВАНИЕ СОПРЯЖЕННЫХ ПРОЦЕССОВ ПЛОТНОСТНОЙ И ТЕПЛОВОЙ КОНВЕКЦИИ В ПОРИСТЫХ СРЕДАХ КОДОМ GERA

Наличие градиентов температуры и концентраций растворенных веществ может оказывать существенное влияние на движение подземных вод. Для описания данного процесса необходимо совместно решать систему уравнений массо-, теплопереноса и фильтрации с учетом переменной плотности растворов. Ниже будет сформулирована математическая модель, описана численная модель, реализованная в коде GeRa, а также приведен пример решения одной из тестовых задач.

5.1 Математическая модель

Уравнения, описывающие тепло-, массоперенос и фильтрацию жидкости в пористой среде выводятся из законов сохранения массы и энергии (см., например, Bear и Cheng^[16], Diersch и Kolditz^[17]). При этом обычно предполагается линейная зависимость плотности от температуры и концентрации примеси:

$$\rho = \rho_0(1 - \beta(T - T_0)) + \kappa_{vol}C. \quad (1)$$

Здесь ρ_0 – референтная плотность (плотность жидкости без примесей при референтной температуре T_0) [ML⁻³], β – коэффициент теплового расширения жидкости [T⁻¹], κ_{vol} – коэффициент объемного расширения за счет примеси [–], T – температура жидкости [K], C – концентрация примеси [ML⁻³].

Течение жидкости описывается уравнением фильтрации (в насыщенных условиях):

$$\rho S \frac{\partial h}{\partial t} - \varphi \rho_0 \beta \frac{\partial T}{\partial t} + \varphi \kappa_{vol} \frac{\partial C}{\partial t} + \nabla(\rho \vec{u}) = \rho_s q_s. \quad (2)$$

Здесь S – коэффициент упругой емкости [L⁻¹], h – гидравлический напор [L], φ – пористость [–], ρ_s – плотность жидкости в источнике [ML⁻³], q_s – интенсивность источника [T⁻¹], \vec{u} – скорость фильтрации [LT⁻¹].

Перенос примеси описывается уравнением конвекции-диффузии:

$$\varphi \frac{\partial C}{\partial t} + \nabla(\vec{u}C) - \nabla(D\nabla C) = C_s q_s. \quad (3)$$

Здесь C_s – концентрация примеси в источнике, D – тензор диффузии-дисперсии:

$$D = D_C + D_m, \quad (4)$$

где D_m – эффективный коэффициент молекулярной диффузии [L²T⁻¹] в пористой среде, D_C – тензор дисперсии, определяемый формулой

$$D_C = \|\vec{u}\|((\alpha_l - \alpha_t)E(\vec{u}) + \alpha_t I), \quad (5)$$

α_l и α_t – продольная и поперечная дисперсивности соответственно [L], I – единичная

матрица размером 3×3 , $E(\vec{u})$ – тензор с компонентами $E_{ij}(u) = u_i u_j / \|\vec{u}\|^2$, $i, j \in \{1, 2, 3\}$.

Уравнение теплопереноса записывается следующим образом:

$$\left[\varphi \rho_0 c^f + (1 - \varphi) \rho^{\text{rock}} c^s \right] \frac{\partial T}{\partial t} + \rho_0 c^f \nabla(\vec{u}T) - \nabla \left[(\lambda + \varphi \rho_0 c^f D_C) \nabla T \right] = q_s \rho_s c^f T_s. \quad (6)$$

Здесь c^f – удельная теплоемкость жидкости [$L^2 T^{-2} K^{-1}$], c^s – удельная теплоемкость твердой матрицы [$L^2 T^{-2} K^{-1}$], ρ^{rock} – плотность твердой матрицы без учета пор [ML^{-3}], T_s – температура источника [K], λ – тензор теплопроводности, определенный следующим образом:

$$\lambda = \left[\varphi \lambda^f + (1 - \varphi) \lambda^s \right] I. \quad (7)$$

Здесь λ^f и λ^s – коэффициенты теплопроводности раствора и твердой матрицы, соответственно, [$ML^3 T^{-2} K^{-1}$].

Скорость фильтрации подчиняется обобщенному закону Дарси:

$$\vec{u} = -K \left(\nabla h + \frac{\rho - \rho_0}{\rho_0} \nabla z \right), \quad (8)$$

где z – вертикальная координата [L], K – тензор фильтрации [LT^{-1}], который, вообще говоря, зависит от концентрации примеси и температуры.

5.2 Численная модель

Система уравнений (1)–(8) решается итерационно методом расщепления по физическим процессам. Пусть верхний индекс n означает значение переменной на n -ом шаге по времени, а индекс k соответствует k -ой итерации. Запишем алгоритм решения задачи:

- 1) С использованием температуры $T^{n+1,k}$ и концентрации $C^{n+1,k}$ с предыдущей итерации вычисляются плотность, вязкость, а также производные $\frac{\partial C}{\partial t}$ и $\frac{\partial T}{\partial t}$. Они подставляются в уравнение фильтрации.

- 2) Уравнение фильтрации решается неявной по времени схемой:

$$\begin{aligned} & \rho^{n+1,k} S \frac{h^{n+1,k+1} - h^n}{\Delta t} - \nabla \left(\rho^{n+1,k} K^{n+1,k} \left(\nabla h^{n+1,k+1} + \frac{\rho^{n+1,k} - \rho_0}{\rho_0} \nabla z \right) \right) = \\ & = \rho_s^{n+1} q_s^{n+1} + \varphi \rho_0 \beta \frac{\partial T^{n+1,k}}{\partial t} - \varphi \kappa_{vol} \frac{\partial C^{n+1,k}}{\partial t} \end{aligned} \quad (9)$$

- 3) По закону Дарси (8) вычисляются фильтрационные потоки $\vec{u}^{n+1,k+1}$.
- 4) Решается уравнение переноса примеси:

$$\varphi \frac{C^{n+1,k+1} - C^n}{\Delta t} + \nabla(\vec{u}^{n+1,k+1} C^n) - \nabla(D^{n+1,k+1} \nabla C^{n+1,k+1}) = C_s^{n+1} q_s^{n+1} \quad (10)$$

5) Решается уравнение теплопереноса:

$$\begin{aligned} & \left[\varphi \rho_0 c^f + (1-\varphi) \rho^{\text{rock}} c^s \right] \frac{T^{n+1,k+1} - T^n}{\Delta t} + \rho_0 c^f \nabla \left(\bar{u}^{n+1,k+1} T^n \right) - \\ & - \nabla \left[\left(\lambda + \varphi \rho_0 c^f D_C^{n+1,k+1} \right) \nabla T^{n+1,k+1} \right] = q_s^{n+1} \rho_s^{n+1} c^f T_s^{n+1} \end{aligned} \quad (11)$$

где Δt – шаг по времени, n – номер шага по времени.

Шаги повторяются несколько итераций до тех пор, пока разница между значениями напоров, концентраций, температур на данной и предыдущей итерациях не станет меньше наперед заданного числа. Величина шага по времени определяется пользователем.

Уравнения тепло- и массопереноса (10) и (11) решаются явно-неявной схемой. Дискретизация по времени выглядит следующим образом (рассмотрим на примере задачи массопереноса, для задачи теплопереноса схема строится аналогично):

$$\frac{C^{n+1} - C^n}{\Delta t} = \frac{C^{n+1} - C^{ad,n+1} + C^{ad,n+1} - C^n}{\Delta t} = \frac{C^{n+1} - C^{ad,n+1}}{\Delta t} + \frac{C^{ad,n+1} - C^n}{\Delta t}. \quad (12)$$

Задача разбивается на две подзадачи:
адвективную

$$\varphi \frac{C^{ad,n+1} - C^n}{\Delta t} + \nabla \left(\bar{u}^{n+1} C^n \right) = C_s^n q_s^n; \quad (13)$$

диффузионную

$$\varphi \frac{C^{n+1} - C^{ad,n}}{\Delta t} - \nabla \left(D^{n+1,k+1} \nabla C^{n+1} \right) = 0. \quad (14)$$

Для решения адвективной задачи с помощью метода конечных объемов (МКО) проинтегрируем уравнение (13) по ячейке E расчетной сетки:

$$\int_E \varphi \frac{C^{ad,n+1} - C^n}{\Delta t} dV + \int_E \nabla \left(\bar{u}^{n+1} C^n \right) dV = \int_E C_s^n q_s^n dV. \quad (15)$$

Заменим функции под интегралами их кусочно-постоянными на ячейках сетки аппроксимациями. Интеграл по объему заменим интегралом по поверхности ячейки по формуле Гаусса–Остроградского:

$$\varphi \frac{C_E^{ad,n+1} - C_E^n}{\Delta t} V_E + \sum_{e \in \partial E} \int_e C_e^n \bar{u}^{n+1} \cdot \bar{n}_{e,E} dS = \int_E C_s^n q_s^n dV. \quad (16)$$

Здесь V_E – объем ячейки E , C_e^n – концентрация на грани e , $\bar{n}_{e,E}$ – внешняя для ячейки E нормаль к грани e , суммирование ведется по граням ячейки E . Фильтрационные потоки через грань $u_{e,E}^{n+1} = \int_e \bar{u}^{n+1} \cdot \bar{n}_{e,E} dS$ рассчитываются в рамках фильтрационной задачи. Тогда итоговое уравнение для решения задачи адвекции:

$$\varphi \frac{C_E^{ad,n+1} - C_E^n}{\Delta t} V_E + \sum_{e \in \partial E} C_e^n u_{e,E}^{n+1} = \int_E C_s^n q_s^n dV \quad (17)$$

Для обеспечения устойчивости схемы необходима противопотоковая аппроксимация концентраций на гранях. В коде GeRa реализованы два вида такой аппроксимации:

- 1) в схеме первого порядка в качестве C_e^n выбирается концентрация с той ячейки, которая лежит выше по потоку;
- 2) в схеме высокого разрешения осуществляется кусочно-линейное восполнение концентрации на ячейках сетки: по некоторым правилам на каждой ячейке определяется линейная функция, среднее значение которой совпадает с концентрацией в этой ячейке. В качестве C_e^n берется среднее по грани значение функции-восполнения с той ячейки, что лежит выше по потоку.

Для второго вида аппроксимации реализованы два варианта кусочно-линейного восполнения:

- метод, основанный на точном решении задачи минимизации квадратичного функционала с ограничением в виде неравенств^[18];
- метод, основанный на приближенном решении задачи минимизации квадратичного функционала с ограничениями в виде неравенств (см. Barth и Ohlberger^[19]).

Первый метод вносит меньшую численную диффузию, однако вычислительно более сложен по сравнению со вторым.

Для дискретизации фильтрационной и диффузионной задач в GeRa могут быть использованы неявные схемы линейного МКО: многоточечная O-схема^[20] или классическая двухточечная схема.

5.3 Результаты вычислительного эксперимента

Продемонстрируем возможности кода GeRa моделировать совместно плотностную и тепловую конвекцию на примере расширенной задачи Элдера, впервые предложенной Diersch и Kolditz^[17] и модифицированной Jamshidzadeh^[21].

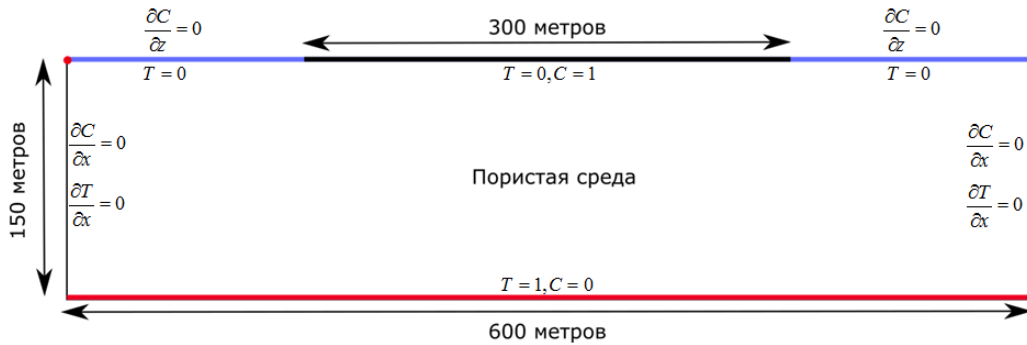


Рис. 1: Расширенная задача Элдера: область моделирования и граничные условия^[21].

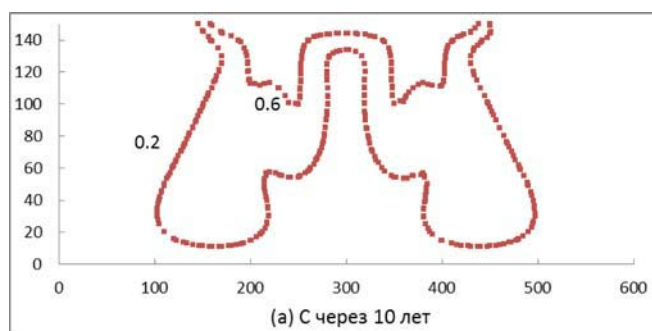
Область моделирования и граничные условия показаны на рисунке 1. Нижняя граница поддерживается горячей. На участке верхней границы поддерживается постоянная максимальная концентрация $C = 1$ (концентрация C здесь является

безразмерной величиной, приведенной к пределу растворимости). На всей границе задано условие непротекания, кроме левого и правого верхних углов, где задан напор $h = 150$ м. Числовые параметры модели приведены в таблице 4.

Задача решается кодом GeRa на равномерной гексаэдральной сетке размером $60 \times 1 \times 30$ ячеек. Используется шаг по времени 30 суток. Для решения фильтрационной задачи применяется двухточечная схема; для решения задач тепло-, массопереноса – явно-неявная схема с использованием схемы высокого разрешения для адвективной подзадачи и двухточечной схемы для диффузионной подзадачи. На рисунках 2 и 3 приведены изолинии концентрации и изотермы для моментов времени 10 и 20 лет. Решение, полученное кодом GeRa, соответствует результатам, изложенным в работе Jamshidzadeh^[21].

Параметр	Символ	Значение	Система единиц
Коэффициент фильтрации	K	0.41	м/сут
Коэффициент диффузии	D_m	0.31	м ² /сут
Упругая емкость	S	10^{-6}	м ⁻¹
Пористость	φ	0.1	-
Козф. продольной дисперсии	α_l	0	м
Козф. поперечной дисперсии	α_t	0	м
Референтная плотность	ρ_0	998.2	кг/м ³
Козф. объемного расширения	κ_{vol}	200	кг/м ³
Теплопроводность жидкости	λ^l	56160	Дж/(м сут °С)
Теплопроводность тв. матрицы	λ^s	137462.4	Дж/(м сут °С)
Козф. теплового расширения	β	5×10^{-4}	°С ⁻¹
Референтная температура	T_0	0	°С
Максимальная температура	T_{max}	1	°С
Плотность твердой матрицы	ρ^{rock}	2650	кг/м ³
Удельная теплоемкость тв. матрицы	c^s	0	Дж/(кг °С)
Удельная теплоемкость жидкости	c^l	4180	Дж/(кг °С)

Таблица 4 : Числовые параметры для расширенной задачи Элдера.



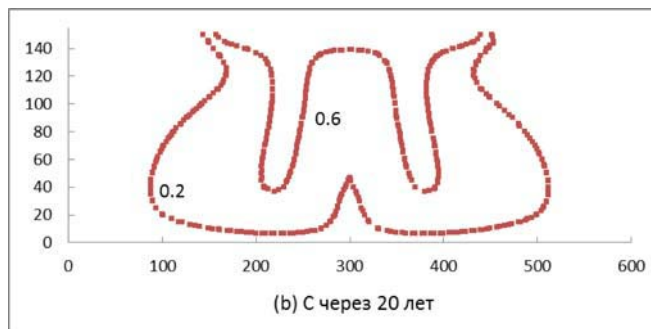


Рис. 2: Изолинии концентрации примеси (а) через 10 лет, (b) через 20 лет.

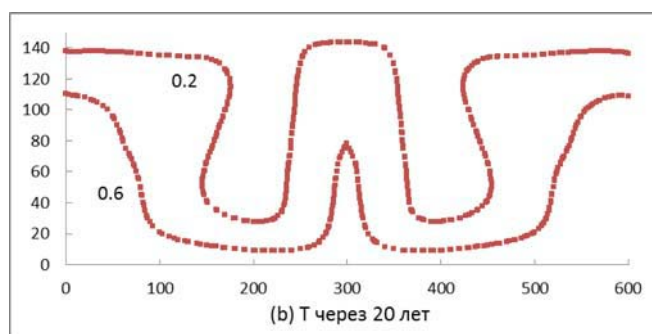
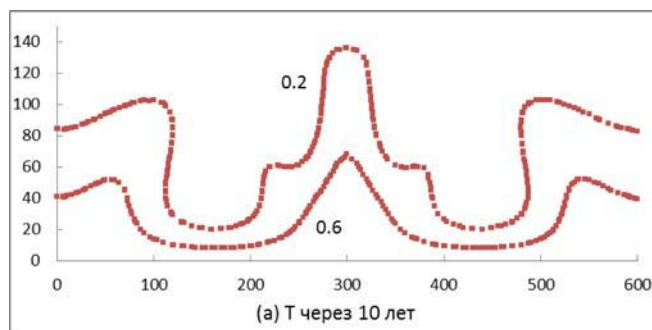


Рис. 3: Изотермы (а) через 10 лет, (b) через 20 лет.

ЗАКЛЮЧЕНИЕ

В работе описаны общие этапы моделирования и выделены наиболее важные функции, которые следует реализовать в сеточной программной платформе общего назначения. Приводится описание программной платформы INMOST и расчетного кода GeRa, созданного на ее основе. Продемонстрированы результаты моделирования совместной задачи плотностной и тепловой конвекции в пористых средах, полученные с помощью расчетного кода GeRa.

REFERENCES

- [1] Vassilevski Yu., Konshin I., Kopytov G., Terekhov K. "INMOST – a software platform and

- graphical environment for development of parallel numerical models on general meshes”, Lomonosov Moscow State Univ. Publ., Moscow, 144 pp (in Russian) (2013).
- [2] INMOST – a toolkit for distributed mathematical modeling. URL: <http://www.inmost.org>.
- [3] Danilov A., Terekhov K., Konshin I., Vassilevski Y. “The structure of INMOST program platform and its usage for numerical modeling problems”. *Proceedings of Russian Supercomputing Days 2015*, 104–109 (2015).
- [4] Konshin I., Kaporin I., Nikitin K., Vassilevski Y. “Parallel linear systems solution for multiphase flow problems in the INMOST framework”. *Proceedings of Russian Supercomputing Days 2015*, 96–103 (2015).
- [5] ParMETIS: Parallel Graph Partitioning and Fill-reducing Matrix Ordering. URL: <http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview> (accessed: 22.07.2016).
- [6] Zoltan: Parallel Partitioning, Load Balancing and Data-Management Services. URL: <http://www.cs.sandia.gov/zoltan> (accessed: 22.07.2016).
- [7] PETSc (Portable, Extensible Toolkit for Scientific Computation). URL: <https://www.mcs.anl.gov/petsc> (accessed: 22.07.2016).
- [8] The Trilinos Project. URL: <https://trilinos.org> (accessed: 22.07.2016).
- [9] Kapyrin I., Vassilevski Yu., Utkin S. “Concept of the design and application of the GeRa numerical code for radioactive waste disposal safety assessment”. *Voprosy Atomnoi Nauki i Tekhniki. Ser.: Mat. Model. Fiz. Prots.*, **4**, 44-54 (in Russian) (2014).
- [10] Kapyrin I., Ivanov V., Kopytov G., Utkin S. “Integral code GeRa for radioactive waste disposal safety validation”. *Gornyi Zhurnal*, **10**, 44-50 (in Russian) (2015).
- [11] Boldyrev K. A., Kapyrin I. V., Konstantinova L. I., Zakharova E. V. “Simulation of strontium sorption onto rocks at high concentrations of sodium nitrate in the solution”. *Radiochemistry*, **58** (3), 243-251 (2016).
- [12] Plenkin A., Chernyshenko A., Chugunov V., Kapyrin I. “Adaptive unstructured mesh generation methods for hydrogeological problems”. *Vychislitelnye metody i programmirovaniye*, **16**, 518-533 (2015).
- [13] Charlton S. R., Parkhurst D. L. “Modules based on the geochemical model PHREEQC for use in scripting and programming languages”. *Computers & Geosciences*, **37** (10), 1653-1663 (2011).
- [14] Freedman V. L. et al. A high-performance workflow system for subsurface simulation. *Environmental Modelling & Software*, **55**, 176-189 (2014).
- [15] Hammond G. E., Lichtner P. C., Mills R. T. “Evaluating the performance of parallel subsurface simulators: An illustrative example with PFLOTRAN”. *Water Resources Research*, **50** (1), 208-228 (2014).
- [16] Bear J., Cheng A.H.-D., *Modeling groundwater flow and contaminant transport*, Springer, Heidelberg, Germany, 834 pp (2010).
- [17] Diersch H. J. G., Kolditz O., “Coupled groundwater flow and transport: 2. Thermohaline and 3D convection systems”, *Adv. Water Res.*, **21**, 401–425 (1998).
- [18] Nikitin K., Vassilevski Yu., “A monotone nonlinear finite volume method for advection–diffusion equations on unstructured polyhedral meshes in 3D”, *Russ. J. Numer. Anal. Math. Modeling*, **25**(4), 335-358 (2010).
- [19] Barth T., Ohlberger M., “Finite volume methods: foundation and analysis”, *Encyclopedia of Computational Mechanics*, 439–473 (2004).
- [20] Aavatsmark I., Barkve T., Boe O., Mannseth T. “Discretization on unstructured grids for inhomogeneous, anisotropic media. Part I: Derivation of the methods”, *SIAM. J. Sci. Comput.*, **19**(5), 1700–1716 (1998).
- [21] Jamshidzadeh Z., Tsai F.T.-C., Mirbagheri S.A., Ghasemzadeh H., “Fluid dispersion effects on density-driven thermohaline flow and transport in porous media”, *Adv. Water Res.*, **61**, 12–28 (2013).
- [22] The Visualization Toolkit (VTK): an open-source, freely available software system for 3D

computer graphics, image processing, and visualization. URL: <http://www.vtk.org> (accessed: 22.07.2016).

[23] ParaView: an open-source, multi-platform data analysis and visualization application. URL: <http://www.paraview.org> (accessed: 22.07.2016).

[24] VisIt: an open source, scalable, visualization, animation and analysis tool. URL: <http://visit.llnl.gov> (accessed: 22.07.2016).

[25] Qt Creator. URL: https://ru.wikipedia.org/wiki/Qt_Creator (accessed: 22.07.2016).

The results were presented at the 14-th International seminar "Mathematical models & modeling in laser-plasma processes & advanced science technologies" (July 4 - 9, 2016, Moscow, Russia).

Received May, 25 2016.