

Министерство образования и науки РФ

Федеральное государственное автономное образовательное учреждение
высшего профессионального образования

«Московский физико-технический институт
(государственный университет)»

Факультет управления и прикладной математики
Кафедра математического моделирования

Люпа Анастасия Александровна

Магистерская диссертация по направлению 010900
«Прикладные математика и физика»:

**Решение задач трехфазной неизотермической фильтрации
на высокопроизводительных вычислительных системах**

Научный руководитель:
с.н.с., к.ф.-м.н. Чурбанова Н.Г.
Рецензент:
с.н.с., к.ф.-м.н. Шильников Е.В.

г. Москва – 2014

Содержание

1 Введение	3
2 Физическая модель	4
3 Математическая модель	7
3.1 Капиллярные давления на границах фаз	7
3.2 Относительные фазовые проницаемости	8
3.3 Теплопроводность, теплоемкость, вязкость	10
3.4 Уравнения состояния	11
3.5 Модифицированное уравнение неразрывности	11
3.6 Полная система уравнений	12
3.7 Границные и начальные условия	12
4 Алгоритм расчета модели	13
4.1 Последовательность расчетов	13
4.2 Численные методы	13
4.3 Решение системы методом Ньютона	16
5 Тестовые задачи	17
5.1 Задача о закачке газа под давлением	17
5.2 Задача о перераспределении фаз под действием силы тяжести в резервуаре	20
5.3 Двумерная задача просачивания с источником на границе	20
5.4 Трехмерная задача просачивания в резервуаре с источником на границе	26
6 Использование высокопроизводительных вычислительных систем	32
6.1 Вычислительные системы	32
6.2 О технологии CUDA	33
6.3 Параллелизм	34
6.4 Практические аспекты реализации программного комплекса	35
6.4.1 Программный комплекс	35
6.4.2 Индексация массива данных	35
6.4.3 Сетка из процессоров	35
6.4.4 Алгоритм первоначального распределения области по процессорам	36
6.4.5 Локальная и глобальная системы координат	37
6.4.6 Организация межпроцессорных обменов	39
6.4.7 Понятие активной точки	39
6.4.8 Ввод/Вывод данных	40

6.5 Результаты расчетов на кластере	42
6.5.1 Результаты применения библиотеки MPI	43
6.5.2 Результаты использования технологии CUDA	43
6.5.3 Результаты совместного применения библиотек CUDA и MPI . .	45
7 Заключение	47
Список литературы	48
Список публикаций	49
Благодарности	50

1 Введение

Теория фильтрации, изучающая законы движения жидкостей, газов и их смесей в пористой среде, имеет обширное практическое применение. На протяжении многих лет фильтрационные расчеты занимают очень важное место при разработке технологий добычи нефти и газа, при проектировании, постройке и эксплуатации гидротехнических и мелиоративных сооружений, в горном деле, в решении экологических проблем. Модели неизотермической фильтрации используются для описания процессов извлечения нефти с применением различных теплоносителей (горячей воды, пара, очага внутрипластового горения). Эти технологии используются, в основном, для добычи высоковязких нефтей и битумов [1]. В связи с растущим потреблением нефти и нефтепродуктов, постепенным истощением ранее разведанных нефтяных месторождений, сверхвязкие тяжелые нефти становятся особенно востребованными в экономике. Такие нефти активно применяются в строительстве, а после очистки их можно использовать в химической промышленности – для производства kleев и пластиков. Поэтому задачи неизотермической фильтрации становятся все более и более актуальными.

Настоящая работа посвящена моделированию проблем, связанных с трехфазной фильтрацией в однородной пористой среде с учетом тепловых процессов, слабой сжимаемости жидкостей и капиллярных сил. Три рассматриваемые фазы: вода, газ и NAPL(Non-Aqueous Phase Liquids). К NAPL, например, относятся нефти, битумы, бензин, тетрахлороэтилен, минеральное топливо, растворители, очищающие средства. Эти жидкости не смешиваются с водой и с газом, поэтому при моделировании их течения в почве говорят о многофазной фильтрации. Результатом моделирования в конечном счете являются распределения насыщенностей, давлений трех фаз и температуры в пласте среды в зависимости от времени, начальных и граничных условий, свойств пористой среды. Численные решения таких задач, также как и других задач газо- и гидродинамики, требуют больших вычислительных затрат. В то же время существует необходимость в наиболее быстром получении результатов. Использование высокопроизводительных вычислительных систем с распределенной памятью позволяет проводить вычисления в течение разумного времени. Поэтому созданные вычислительные алгоритмы были адаптированы к расчетам на многопроцессорных вычислительных комплексах. Для параллельных вычислений используются стандартная библиотека MPI и технология Nvidia CUDA. При расчетах применяются явные разностные схемы, допускающие эффективное распараллеливание. Вычисления проводятся на гибридном вычислительном кластере К-100 ИПМ им. М.В.Келдыша РАН.

2 Физическая модель

Законы движения флюидов в пористых средах базируются на сохранении массы, энергии, импульса. Основные физические свойства пористой среды связаны с элементарным объемом среды, который должен быть достаточно большим по сравнению с размером пор. Подземные пространства характеризуются сложной системой пор, каналов, трещин, размеры которых малы по сравнению с характерными размерами среды, и по которым может происходить течение жидкости или газа. Качественной характеристикой пористости среды служит отношение объема пор к общему объему:

$$m = V_n/V,$$

где m – коэффициент пористости, V_n – объем пор, V – общий объем данного элемента среды. Течение через такие пористые тела, при котором сила трения флюида (жидкости, газа) о скелет играет определяющую роль, называется фильтрацией. Часть системы, все компоненты которой имеют одинаковые физические и химические свойства, называется фазой.

Главными характеристиками движения многофазной системы являются насыщенности и скорости фильтрации каждой из фаз. Насыщенностью S_i порового пространства i -ой фазы называется доля объема пор, занятая этой фазой в элементарном объеме:

$$S_i = \frac{\Delta V_i}{\Delta V}, i = 1, 2 \dots n, \quad \sum_{i=1}^n S_i = 1. \quad (2.1)$$

Для каждой фазы существует предельная насыщенность, такая, что при меньших значениях насыщенности эта фаза неподвижна. Эти значения носят название остаточных насыщенностей. Обозначим их с помощью нижнего индекса r у насыщенностей. Таким образом, совместное течение двух фаз имеет место лишь в интервале насыщенностей. Для работы в данном интервале насыщенностей вводится понятие эффективных насыщенностей фаз, их обозначаем с помощью символа верхней черты. Эффективная насыщенность i -ой фазы может быть представлена в виде:

$$\overline{S}_i = \frac{S_i - S_{ir}}{1 - \sum_{k=1}^n S_{kr}}.$$

Проекция скорости фильтрации i -ой фазы в некоторой точке \vec{u}_i на некоторое направление равна отношению объемного расхода данной фазы к площадке, перпендикулярной к указанному направлению. Основное соотношение теории фильтрации – закон фильтрации(закон Дарси) – устанавливает связь между вектором скорости фильтрации и тем полем давления, которое вызывает фильтрационное течение. Закон Дарси в теории фильтрации заменяет собой уравнение движения. Для i -ой фазы при учете силы тяжести его можно записать в виде:

$$\vec{u}_i = -K \frac{k_i(S_i)}{\mu_i} (\text{grad}P_i - \rho_i \vec{g}), \quad (2.2)$$

где K – характеристика пористой среды, называемая абсолютной проницаемостью, определяемая по данным о фильтрации однородной жидкости и не зависящая от свойств жидкости; μ_i – коэффициент динамической вязкости i -ой фазы ($\mu_i = \mu_i(T)$); $k_i(S_i)$ – относительная фазовая проницаемость i -ой фазы (может зависеть не только от насыщенности i -ой фазы), определяемая экспериментально; P_i – давление в i -ой фазе; ρ_i – плотность i -ой фазы; \vec{g} – вектор ускорения свободного падения. Можно выделить верхнюю и нижнюю границы применимости закона Дарси [2]. Верхняя граница связана с проявлением инерционных сил при достаточно высоких скоростях фильтрации. Нижняя – с взаимодействием жидкости с твердым скелетом пористой среды при достаточно малых скоростях фильтрации.

Еще одно фундаментальное соотношение в теории фильтрации - уравнение неразрывности. Для i -ой фазы оно принимает вид:

$$\frac{\partial(m\rho_i S_i)}{\partial t} + \operatorname{div}(\rho_i \vec{u}_i) = \rho_i q_i. \quad (2.3)$$

В отсутствие объемных источников i -ой фазы соответствующее слагаемое в правой части уравнения ($\rho_i q_i$) обнуляется.

При малых размерах области фильтрации и малых скоростях капиллярные силы могут превзойти внешний перепад давления, и их необходимо учитывать. Капиллярные эффекты обусловлены межмолекулярными взаимодействиями двух различных фаз. Эти силы приводят к появлению угла смачивания на границе раздела двух фаз и к разрыву давления на этой границе. Разность фазовых давлений есть так называемое *капиллярное давление*. Соотношения между капиллярными давлениями и насыщенностями обычно получают по опытным данным как функции насыщенностей.

При неизотермической фильтрации свойства и состав флюидов зависят еще от одной переменной – температуры. Поэтому для замыкания системы уравнений фильтрации необходимо дополнительное уравнение – закон сохранения энергии. Для многофазной системы без учета химических реакций закон сохранения энергии может быть записан в виде

$$\frac{\partial \left(m \sum_i \rho_i S_i E_i + (1-m) \rho_r E_r \right)}{\partial t} + \operatorname{div} \left(\sum_i \rho_i H_i \vec{u}_i \right) = \operatorname{div} (\lambda_{eff} \operatorname{grad} T), \quad (2.4)$$

где суммирование производится по всем активным фазам, индекс r обозначает твердую породу, λ_{eff} – эффективный коэффициент теплопроводности:

$$\lambda_{eff} = m \sum_i S_i \lambda_i + (1-m) \lambda_r, \quad (2.5)$$

связь между внутренней энергией E_i и энталпией H_i :

$$E_i = H_i - \frac{P_i}{\rho_i}, \quad (2.6)$$

для нахождения энталпии необходимо определить зависимость теплоемкости вещества при постоянном давлении от температуры:

$$H_i = H_{i0} + \int_{T_0}^T C_P(T) dT. \quad (2.7)$$

Зависимости коэффициентов теплопроводности и теплоемкостей находятся эмпирически.

Для замыкания системы дополнительно вводятся уравнения состояния рассматриваемого флюида и пористой среды.

3 Математическая модель

В работе проводится моделирование трехфазной неизотермической фильтрации несмешивающихся жидкостей и газа с учетом их сжимаемости. Скелет породы считаем неподвижным, пористость и плотность породы – постоянными во всем рассматриваемом объеме, среду – изотропной, жидкие фазы – слабосжимаемыми, газ – идеальным, температуру – единой для всех фаз.

Рассматриваются одномерные, двумерные и трехмерные области, заданные в декартовых координатах.

Рассмотрим более детально описанные в главе, посвященной физической модели, законы и принципы с учетом введенных ограничений и опишем используемые зависимости физических свойств фаз.

Пусть трехфазная система представляет собой две жидкие фазы – вода и легкий нефтяной продукт (LNAPL – от английского Light Non-Aqueous Phase Liquids, плотность которого меньше плотности воды), и одну газообразную. Для удобства введем индексные обозначения для разных фаз: w – вода, n – легкая нефть, g – газ.

В качестве базовых переменных выбираем насыщенности, давления фаз и температуру. Причем для насыщенностей в силу определения справедливо $S_w + S_n + S_g = 1$. А давления P_n и P_g отличаются от P_w на величины капиллярных давлений, являющихся функциями от насыщенностей.

Вводятся эффективные насыщенности каждой из фаз: $\overline{S_i} = \frac{S_i - S_{ir}}{1 - S_{wr} - S_{nr} - S_{gr}}$, где $i = w, n, g$, S_{wr} , S_{nr} , S_{gr} – остаточные насыщенности фаз.

3.1 Капиллярные давления на границах фаз

Для описания капиллярных давлений выбрана приближенная модель Паркера [3]:

$$P_{cnw}(\overline{S_w}) = P_n - P_w = \frac{1}{\gamma \delta_{nw}} \left(\overline{S_w}^{\frac{n}{1-n}} - 1 \right)^{\frac{1}{n}}, \quad 0 < \overline{S_w} < 1,$$

$$P_{cgn}(\overline{S_g}) = P_g - P_n = \frac{1}{\gamma \delta_{gn}} \left((1 - \overline{S_g})^{\frac{n}{1-n}} - 1 \right)^{\frac{1}{n}}, \quad 0 < \overline{S_g} < 1,$$

где $P_{cnw}(\overline{S_w})$ и $P_{cgn}(\overline{S_g})$ – капиллярные давления на границах вода-нефть и нефть-газ, соответственно, n и γ – параметры породы Ван Генухтена из соответствующего приближения Ван Генухтена [4], δ_{nw} , δ_{gn} – коэффициенты, определяемые поверхностным натяжением жидкостей. В работе используются следующие значения параметров: $n = 3.25$, $\gamma = 0.00048 \text{ Па}^{-1}$, $\delta_{nw} = 0.67$, $\delta_{gn} = 2$. Зависимости капиллярных давлений от насыщенностей фаз изображены на Рис. 3.1, Рис. 3.2.

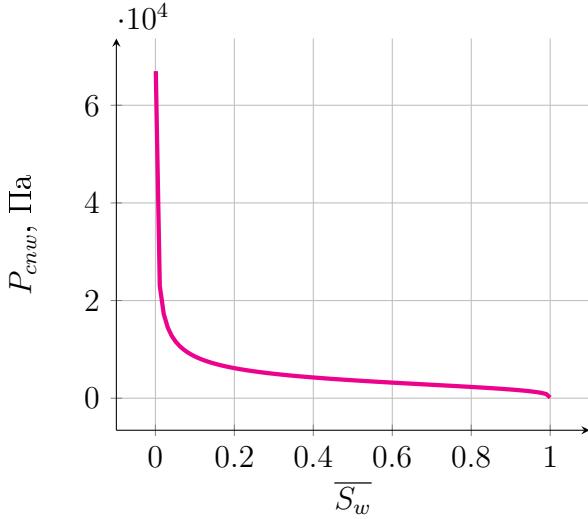


Рис. 3.1. Зависимость капиллярного давления на границе нефть-вода P_{cnw} от эффективной насыщенности водной фазы \overline{S}_w

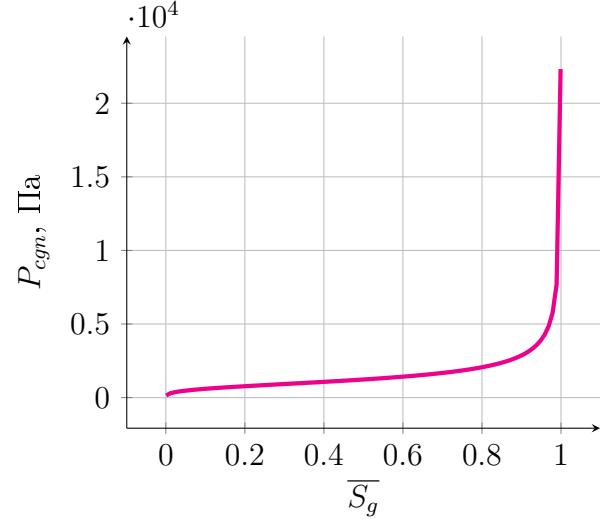


Рис. 3.2. Зависимость капиллярного давления на границе газ-нефть P_{cgn} от эффективной насыщенности газовой фазы \overline{S}_g

3.2 Относительные фазовые проницаемости

Относительные фазовые проницаемости определяются в работе в соответствии с приближением Стоуна [2]:

$$k_w(\overline{S}_w) = \begin{cases} \overline{S}_w^{-\frac{1}{2}} \left(1 - \left(1 - \overline{S}_w^{\frac{n}{n-1}} \right)^{\frac{n-1}{n}} \right)^2, & 0 < \overline{S}_w < 1 \\ 1, & \overline{S}_w \geq 1 \\ 0, & \overline{S}_w \leq 0 \end{cases}$$

$$k_g(\overline{S}_g) = \begin{cases} \overline{S}_g^{-\frac{1}{2}} \left(1 - \left(1 - \overline{S}_g \right)^{\frac{n}{n-1}} \right)^{\frac{2(n-1)}{n}}, & 0 < \overline{S}_g < 1 \\ 1, & \overline{S}_g \geq 1 \\ 0, & \overline{S}_g \leq 0 \end{cases}$$

$$k_n(\overline{S}_w, \overline{S}_n) = \begin{cases} \frac{\overline{S}_n k_{nw}(\overline{S}_w) k_{ng}(\overline{S}_n)}{(1 - \overline{S}_w)(\overline{S}_w + \overline{S}_n)}, & \overline{S}_w < 1, \quad \overline{S}_w + \overline{S}_n > 0 \\ 0, & \text{иначе} \end{cases}, \text{ где}$$

$$k_{nw}(\overline{S}_w) = \begin{cases} (1 - \overline{S}_w)^{\frac{1}{2}} \left(1 - \overline{S}_w^{\frac{n}{n-1}} \right)^{\frac{2(n-1)}{n}}, & 0 < \overline{S}_w < 1 \\ 1, & \overline{S}_w \leq 0 \\ 0, & \overline{S}_w \geq 1 \end{cases},$$

$$k_{ng}(\overline{S_n}) = \begin{cases} \overline{S_n}^{\frac{1}{2}} \left(1 - \left(1 - \overline{S_n}^{\frac{n}{n-1}} \right)^{\frac{n-1}{n}} \right)^2, & 0 < \overline{S_n} < 1 \\ 1, & \overline{S_n} \geq 1 \\ 0, & \overline{S_n} \leq 0 \end{cases}$$

Таким образом, относительные проницаемости воды и газа являются функциями одной переменной (насыщенности одной из фаз), а нефти – двух. Данные зависимости изображены на Рис. 3.3, Рис. 3.4, Рис. 3.5, соответственно.

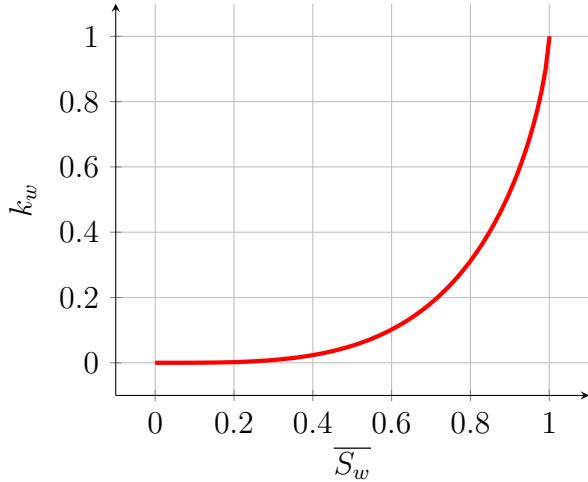


Рис. 3.3. Зависимость относительной фазовой проницаемости воды k_w от эффективной насыщенности водной фазы $\overline{S_w}$

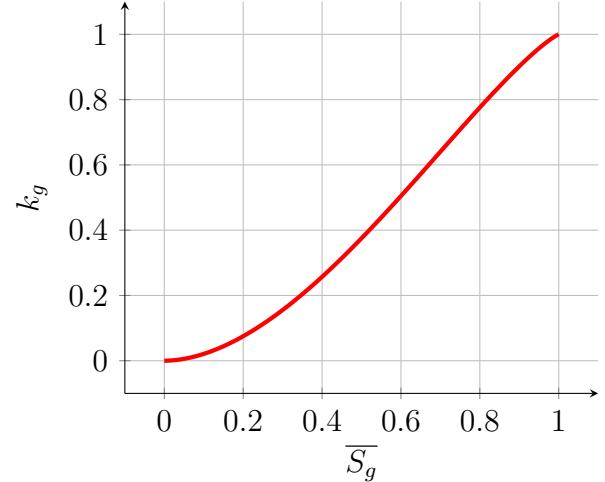


Рис. 3.4. Зависимость относительной фазовой проницаемости газа k_g от эффективной насыщенности газовой фазы $\overline{S_g}$

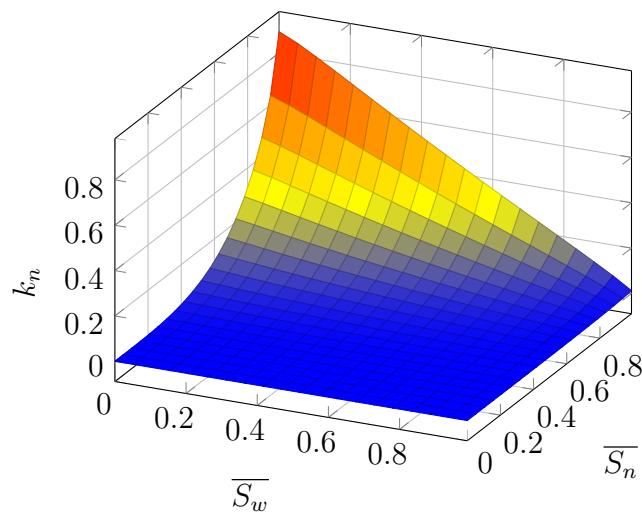


Рис. 3.5. Зависимость относительной фазовой проницаемости нефти k_n от эффективных насыщенностей водной и нефтяной фаз $\overline{S_w}$ и $\overline{S_n}$

3.3 Теплопроводность, теплоемкость, вязкость

Коэффициенты теплопроводности фаз представлены зависимостями:

$$\begin{aligned}\lambda_w(T) &= 0.553 \times (1 - 0.003 \times (T - T_0)), \\ \lambda_n(T) &= 0.14 \times (1 - 0.001 \times (T - T_0)), \\ \lambda_g(T) &= 0.237 \times \left(\frac{T}{T_0}\right)^{0.82}, \\ \lambda_r(T) &= 1.0.\end{aligned}\tag{3.1}$$

Теплоемкости фаз при постоянном давлении:

$$\begin{aligned}C_{Pw}(T) &= 4194 - 1.15 \times (T - T_0) + 0.015 \times (T - T_0)^2, \\ C_{Pn}(T) &= 1700 - 3.4 \times (T - T_0), \\ C_{Pg}(T) &= 1000 - 0.119 \times (T - T_0), \\ C_{Pr}(T) &= 800 - 0.75 \times (T - T_0).\end{aligned}\tag{3.2}$$

Динамические вязкости фаз (см. Рис. 3.6, Рис. 3.7, Рис. 3.8):

$$\begin{aligned}\mu_w(T) &= \frac{1}{29.21 \times T - 7506.64}, \\ \mu_n(T) &= 7.256 \times 10^{-10} \times e^{\frac{4141.9}{T}}, \\ \mu_g(T) &= 1.717 \times 10^{-5} \times \left(\frac{T}{T_0}\right)^{0.683}.\end{aligned}\tag{3.3}$$

Здесь $T_0 = 273K$.

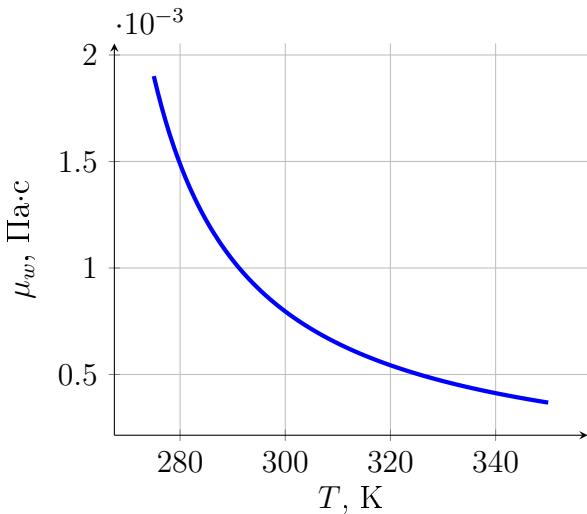


Рис. 3.6. Зависимость вязкости воды μ_w от температуры

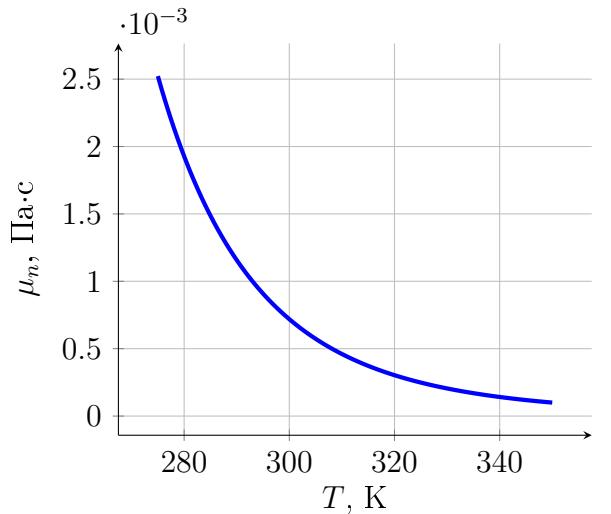


Рис. 3.7. Зависимость вязкости нефти μ_n от температуры

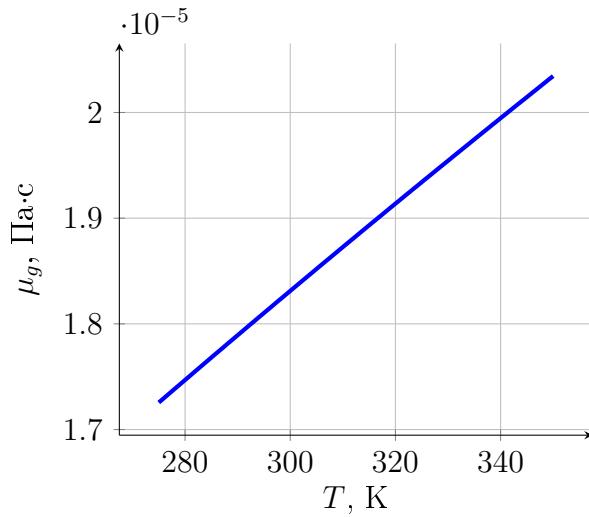


Рис. 3.8. Зависимость вязкости воздуха μ_g от температуры

3.4 Уравнения состояния

Перейдем к уравнениям состояний фаз. Водную и нефтяную фазы считаем слабосжимаемыми и линейно зависящими от перепада температуры:

$$\rho_i = \rho_{i0}(1 + \beta_i(P_i - P_0) - \alpha_i(T - T_0)), \quad 0 < \beta_i \ll 1, \quad 0 < \alpha_i \ll 1, \quad i = w, n,$$

где ρ_{i0} – известное значение плотности i -ой фазы, соответствующее значению давления P_0 и температуры T_0 .

Для газа предполагаем справедливым уравнение состояния идеального газа:

$$\rho_g = \rho_{g0} \frac{P_g}{P_0} \frac{T_0}{T}.$$

В работе используются следующие значения:

$$\begin{aligned} \rho_r &= 2000 \text{ кг/м}^3, \quad \rho_{w0} = 1000 \text{ кг/м}^3, \quad \rho_{n0} = 850 \text{ кг/м}^3, \quad \rho_{g0} = 1.4 \text{ кг/м}^3, \\ \beta_w &= 4.4 \cdot 10^{-7} \text{ 1/Па}, \quad \beta_n = 10^{-6} \text{ 1/Па}, \quad \alpha_w = 1.32 \cdot 10^{-7} \text{ 1/K}, \quad \alpha_n = 9.2 \cdot 10^{-7} \text{ 1/K}. \end{aligned}$$

3.5 Модифицированное уравнение неразрывности

При построении модели наряду с классическим уравнением неразрывности 2.3 рассматривается модифицированное уравнение (строится по аналогии с квазигазодинамической системой уравнений [5]):

$$\frac{\partial(m\rho_i S_i)}{\partial t} + \operatorname{div}(\rho_i \vec{u}_i) = \rho_i q_i + l c_i \cdot \operatorname{div}(\operatorname{grad}(\rho_i S_i)), \quad (3.4)$$

где l – характерный масштаб (расстояние порядка сотни размеров зерен породы [6]), c_i – скорость распространения звука в i -ой среде. В работе используются следующие значения: $l = 10^{-7}$ м, $c_w = 1500$ м/с, $c_n = 1000$ м/с, $c_g = 330$ м/с.

В [7], [8] были предложены модели одно- и двухфазной фильтрации, основанные на кинетическом подходе. В данной работе предложена аналогичная модель трехфазной фильтрации.

3.6 Полная система уравнений

Таким образом, полная система уравнений для описания трехфазной фильтрации имеет вид:

$$\left\{ \begin{array}{l} \frac{\partial \left(m \sum_i \rho_i S_i E_i(P_i, T) + (1-m) \rho_r E_r(P_w, T) \right)}{\partial t} + \\ + \operatorname{div} \left(\sum_i \rho_i H_i(T) \vec{u}_i \right) = \operatorname{div} (\lambda_{eff} \operatorname{grad} T), \\ \frac{\partial (m \rho_i S_i)}{\partial t} + \operatorname{div} (\rho_i \vec{u}_i) = \rho_i q_i, \\ \vec{u}_i = -K \frac{k_i}{\mu_i(T)} (\operatorname{grad} P_i - \rho_i \vec{g}), \\ i = w, n, g, \\ P_n = P_w + P_{cnw}(\overline{S_w}), \\ P_g = P_w + P_{cnw}(\overline{S_w}) + P_{cgn}(\overline{S_g}), \\ S_w + S_n + S_g = 1, \\ k_w = k_w(\overline{S_w}), \quad k_g = k_g(\overline{S_g}), \quad k_n = k_n(\overline{S_w}, \overline{S_n}), \\ \rho_i = \rho_i(P_i, T). \end{array} \right. \quad (3.5)$$

Где $\rho_i q_i$ – источниковые члены, индекс r обозначает твердую породу. Используемые параметры породы: $K = 6.64 \cdot 10^{-11}$ м², $m=0.4$. Все значения констант приводятся в системе единиц СИ.

3.7 Границные и начальные условия

В зависимости от конкретной постановки задачи в работе используются различные граничные условия. На границе поддерживается постоянное давление, или же оно определяется из условия непротекания фазы (нормальная компонента скорости фазы на границе равна 0, $(\vec{u}_i \cdot \vec{n}) = 0$). Для насыщенности ставится условие равенства нулю потока через граничную поверхность или же задается поток насыщенности в виде известной функции через $\frac{\partial S_i}{\partial \vec{n}}$, $i = w, n$, \vec{n} – нормаль к границе. Также для насыщенности может быть задано некоторое постоянное значение на границе. Для температуры тоже ставятся граничные условия первого или второго рода.

В начальный момент времени считаем известными распределения давления водной фазы, насыщенностей фаз и температуры по всей области.

4 Алгоритм расчета модели

4.1 Последовательность расчетов

Численное решение полученной системы уравнений разбито на этапы. После применения начальных условий на каждом шаге по времени выполняется следующая последовательность действий:

1. Применение граничных условий.
2. Вычисление давлений P_n , P_g через P_w и капиллярные давления.
3. Вычисление плотностей фаз.
4. Нахождение относительных фазовых проницаемостей фаз и вязкости.
5. Определение коэффициентов в законе Дарси для фаз.
6. Нахождение $\rho_i S_i$ на следующем шаге по времени из уравнения неразрывности (2.3) явным численным методом.
7. Нахождение внутренней энергии системы на следующем шаге по времени из уравнения сохранения энергии (2.4) явным численным методом.
8. Решение нелинейной системы из пяти уравнений методом Ньютона, в результате чего находим P_w , S_w , S_n , S_g , T на следующем шаге по времени.
9. Сохранение полученных значений переменных в текстовый файл в формате, подходящем для визуализации.
10. Обмены данными при многопроцессорных вычислениях.

4.2 Численные методы

Остановимся подробнее на численных методах, используемых на шаге 6 описанного выше алгоритма. Выбран класс явных двухслойных схем на равномерных декартовых сетках, допускающих эффективное распараллеливание решения. Рассмотрены две различные схемы:

- с направленными разностями для уравнения неразрывности 2.3;
- с центральными разностями для модифицированного уравнения неразрывности 3.4.

Для описания схем используем обозначения из [9]:

Пусть n, m, k – текущие индексы по осям x, y, z , соответственно,
 $\widehat{}$ – обозначает значение величины на следующем слое по времени,
 τ – шаг по времени,
 h_x, h_y, h_z – шаги по пространству по вдоль направлений осей x, y, z ;

Схема с направленными разностями

Представим уравнение (2.3) в виде:

$$m \frac{\partial(\rho_i S_i)}{\partial t} + \operatorname{div}(\rho_i \chi_i (\operatorname{grad} P_i - \rho_i \vec{g})) = \rho_i q_i, \quad (4.1)$$

$$\chi_i = -K \frac{k_i}{\mu_i}.$$

Разностная схема для этого уравнения может быть записана следующим образом:

$$x_{li} = \frac{P_{i_{n,m,k}} - P_{i_{n-1,m,k}}}{h_x}, \quad x_{ri} = \frac{P_{i_{n+1,m,k}} - P_{i_{n,m,k}}}{h_x};$$

$$y_{li} = \frac{P_{i_{n,m,k}} - P_{i_{n,m-1,k}}}{h_y} - \rho_{i_{n,m,k}} g, \quad y_{ri} = \frac{P_{i_{n,m+1,k}} - P_{i_{n,m,k}}}{h_y} - \rho_{i_{n,m,k}} g;$$

$$z_{li} = \frac{P_{i_{n,m,k}} - P_{i_{n,m,k-1}}}{h_z}, \quad z_{ri} = \frac{P_{i_{n,m,k+1}} - P_{i_{n,m,k}}}{h_z};$$

$$f_{xi} = \frac{1}{2h_x} ((x_{ri} - |x_{ri}| - x_{li} - |x_{li}|) \chi_{i_{n,m,k}} \rho_{i_{n,m,k}} \\ - (x_{li} - |x_{li}|) \chi_{i_{n-1,m,k}} \rho_{i_{n-1,m,k}} \\ + (x_{ri} - |x_{ri}|) \chi_{i_{n+1,m,k}} \rho_{i_{n+1,m,k}});$$

$$f_{yi} = \frac{1}{2h_y} ((y_{ri} - |y_{ri}| - y_{li} - |y_{li}|) \chi_{i_{n,m,k}} \rho_{i_{n,m,k}} \\ - (y_{li} - |y_{li}|) \chi_{i_{n,m-1,k}} \rho_{i_{n,m-1,k}} \\ + (y_{ri} - |y_{ri}|) \chi_{i_{n,m+1,k}} \rho_{i_{n,m+1,k}});$$

$$f_{zi} = \frac{1}{2h_z} ((z_{ri} - |z_{ri}| - z_{li} - |z_{li}|) \chi_{i_{n,m,k}} \rho_{i_{n,m,k}} \\ - (z_{li} - |z_{li}|) \chi_{i_{n,m,k-1}} \rho_{i_{n,m,k-1}} \\ + (z_{ri} - |z_{ri}|) \chi_{i_{n,m,k+1}} \rho_{i_{n,m,k+1}});$$

$$\widehat{(\rho_i S_i)}_{k,l,m} = (\rho_i S_i)_{k,l,m} + \frac{\tau}{m} (\rho_i q_i - f_{xi} - f_{yi} - f_{zi}).$$

Если рассматривается меньшее количество измерений, соответствующее слагаемое f_i обнуляется.

Данный метод обладает первым порядком аппроксимации по времени и пространству и устойчив при $\tau < 0.25h_{min}^2$.

Аналогично строится схема для нахождения \widehat{E} .

Схема с центральными разностями

Представим уравнение (3.4) в виде:

$$m \frac{\partial(\rho_i S_i)}{\partial t} + \operatorname{div}(\rho_i \chi_i (\operatorname{grad} P_i - \rho_i \vec{g})) = \rho_i q_i + l c_i \cdot \operatorname{div}(\operatorname{grad}(\rho_i S_i)), \quad (4.2)$$

$$\chi_i = -K \frac{k_i}{\mu_i}.$$

Разностная схема для этого уравнения может быть записана следующим образом:

$$\begin{aligned} f_{xi} = & \chi_{i_{n,m,k}} \rho_{i_{n,m,k}} \frac{P_{i_{n+1,m,k}} - 2P_{i_{n,m,k}} + P_{i_{n-1,m,k}}}{h_x^2} \\ & + \frac{P_{i_{n+1,m,k}} - P_{i_{n-1,m,k}}}{2h_x} \cdot \frac{\chi_{i_{n+1,m,k}} \rho_{i_{n+1,m,k}} - \chi_{i_{n-1,m,k}} \rho_{i_{n-1,m,k}}}{2h_x} \\ & - l c_i \frac{\rho_{i_{n+1,m,k}} S_{i_{n+1,m,k}} - 2\rho_{i_{n,m,k}} S_{i_{n,m,k}} + \rho_{i_{n-1,m,k}} S_{i_{n-1,m,k}}}{h_x^2}; \end{aligned}$$

$$\begin{aligned} f_{yi} = & \chi_{i_{n,m,k}} \rho_{i_{n,m,k}} \frac{P_{i_{n,m+1,k}} - 2P_{i_{n,m,k}} + P_{i_{n,m-1,k}}}{h_y^2} \\ & + \frac{P_{i_{n,m+1,k}} - P_{i_{n,m-1,k}}}{2h_y} \cdot \frac{\chi_{i_{n,m+1,k}} \rho_{i_{n,m,k}} - \chi_{i_{n,m-1,k}} \rho_{i_{n,m-1,k}}}{2h_y} \\ & - l c_i \frac{\rho_{i_{n,m+1,k}} S_{i_{n,m+1,k}} - 2\rho_{i_{n,m,k}} S_{i_{n,m,k}} + \rho_{i_{n,m-1,k}} S_{i_{n,m-1,k}}}{h_y^2}; \end{aligned}$$

$$\begin{aligned} f_{zi} = & \chi_{i_{n,m,k}} \rho_{i_{n,m,k}} \frac{P_{i_{n,m,k+1}} - 2P_{i_{n,m,k}} + P_{i_{n,m,k-1}}}{h_z^2} \\ & + \frac{P_{i_{n,m,k+1}} - P_{i_{n,m,k-1}}}{2h_z} \cdot \frac{\chi_{i_{n,m,k+1}} \rho_{i_{n,m,k+1}} - \chi_{i_{n,m,k-1}} \rho_{i_{n,m,k-1}}}{2h_z} \\ & - l c_i \frac{\rho_{i_{n,m,k+1}} S_{i_{n,m,k+1}} - 2\rho_{i_{n,m,k}} S_{i_{n,m,k}} + \rho_{i_{n,m,k-1}} S_{i_{n,m,k-1}}}{h_z^2}; \end{aligned}$$

$$(\widehat{\rho_i S_i})_{k,l,m} = (\rho_i S_i)_{k,l,m} + \frac{\tau}{m} (\rho_i q_i - f_{xi} - f_{yi} - f_{zi}).$$

Данный метод позволяет увеличить точность, так как имеет второй порядок аппроксимации по пространству. Благодаря наличию дополнительного члена в правой части уравнения он является условно устойчивым.

4.3 Решение системы методом Ньютона

На шаге 8 предложенного алгоритма в каждом узле расчетной сетки возникает нелинейная система уравнений. Ее решение проводится методом Ньютона [9], выполняется семь итераций метода. Каждая итерация состоит из следующей последовательности действий:

$$\begin{aligned}
F_1 &= \rho_w(P_w, T)S_w - (\widehat{\rho_w S_w}) \\
F_2 &= \rho_n(P_w + P_{cnw}(S_w))S_n - (\widehat{\rho_n S_n}) \\
F_3 &= \rho_g(P_w + P_{cnw}(S_w) + P_{cgn}(S_g))S_g - (\widehat{\rho_g S_g}) \\
F_4 &= m \left(S_w(\rho_w(P_w, T)H_w(T) - P_w) \right. \\
&\quad + S_n(\rho_n(P_w + P_{cnw}(S_w), T)H_n(T) - P_w - P_{cnw}(S_w)) \\
&\quad + S_g(\rho_g(P_w + P_{cnw}(S_w) + P_{cgn}(S_g), T)H_g(T) - P_w - P_{cnw}(S_w) - P_{cgn}(S_g)) \Big) \\
&\quad + (1-m)(\rho_r H_r - P_w) - \widehat{E} \\
F_5 &= S_w + S_n + S_g - 1 \\
A &= \begin{pmatrix} \frac{\partial F_1}{\partial P_w} & \frac{\partial F_1}{\partial S_w} & \frac{\partial F_1}{\partial S_n} & \frac{\partial F_1}{\partial S_g} & \frac{\partial F_1}{\partial T} \\ \frac{\partial F_2}{\partial P_w} & \frac{\partial F_2}{\partial S_w} & \frac{\partial F_2}{\partial S_n} & \frac{\partial F_2}{\partial S_g} & \frac{\partial F_2}{\partial T} \\ \frac{\partial F_3}{\partial P_w} & \frac{\partial F_3}{\partial S_w} & \frac{\partial F_3}{\partial S_n} & \frac{\partial F_3}{\partial S_g} & \frac{\partial F_3}{\partial T} \\ \frac{\partial F_4}{\partial P_w} & \frac{\partial F_4}{\partial S_w} & \frac{\partial F_4}{\partial S_n} & \frac{\partial F_4}{\partial S_g} & \frac{\partial F_4}{\partial T} \\ \frac{\partial F_5}{\partial P_w} & \frac{\partial F_5}{\partial S_w} & \frac{\partial F_5}{\partial S_n} & \frac{\partial F_5}{\partial S_g} & \frac{\partial F_5}{\partial T} \end{pmatrix} \tag{4.3}
\end{aligned}$$

Тогда

$$\begin{pmatrix} P_w \\ S_w \\ S_n \\ S_g \\ T \end{pmatrix}^{new} = \begin{pmatrix} P_w \\ S_w \\ S_n \\ S_g \\ T \end{pmatrix} - A^{-1} \begin{pmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \end{pmatrix} \tag{4.4}$$

Для обращения матрицы A используется метод Гаусса с выбором главного элемента [9].

5 Тестовые задачи

В соответствии с описанной математической моделью были проведены расчеты нескольких тестовых задач, результаты которых качественно верно описывают рассматриваемые явления. Используемые параметры модели описаны в разделе 3. Во всех тестовых задачах пористость и абсолютная проницаемость породы постоянны во всем объеме, остаточные насыщенности полагаются равными нулю, размер области $1 \text{ м} \times 1 \text{ м} \times 1 \text{ м}$.

5.1 Задача о закачке газа под давлением

Рассмотрим две одномерные модельные задачи неизотермической трехфазной фильтрации, где течение происходит из-за разности давлений на концах рассматриваемого отрезка в горизонтальном направлении, отличающиеся только граничными условиями на температуру. Таким образом, покажем влияние нагревания на процесс фильтрации в полубесконечной области. Под давлением идет закачка газа слева-направо.

Начальные условия:

$$S_w = 0.4, S_n = 0.3, S_g = 0.3, P_w = P_{\text{атм}}, T = 285K. \quad (5.1)$$

Граничные условия для насыщенностей и давления:

$$\begin{aligned} P_w|_{x=0} &= 1.1 \cdot P_{\text{атм}}, \quad P_w|_{x=1} = P_{\text{атм}}; \\ S_w|_{x=0} &= 0.15, \quad \left. \frac{\partial S_w}{\partial x} \right|_{x=1} = 0; \\ S_n|_{x=0} &= 0.15, \quad \left. \frac{\partial S_n}{\partial x} \right|_{x=1} = 0. \end{aligned} \quad (5.2)$$

Два варианта граничных условий для температуры:

$$T|_{x=0} = T|_{x=1} = 285K; \quad (5.3)$$

$$T|_{x=0} = 320K, \quad \left. \frac{\partial T}{\partial x} \right|_{x=1} = 0. \quad (5.4)$$

Результаты расчетов представлены в виде зависимостей профилей насыщенностей фаз, распределения давления и температуры от времени на Рис. 5.9 - Рис. 5.20. Сравним, как проходит процесс при двух различных граничных условиях для температуры. Можно видеть, что благодаря нагреванию изменяется форма графиков насыщенностей, просачивание происходит быстрее, меняется распределение давлений.

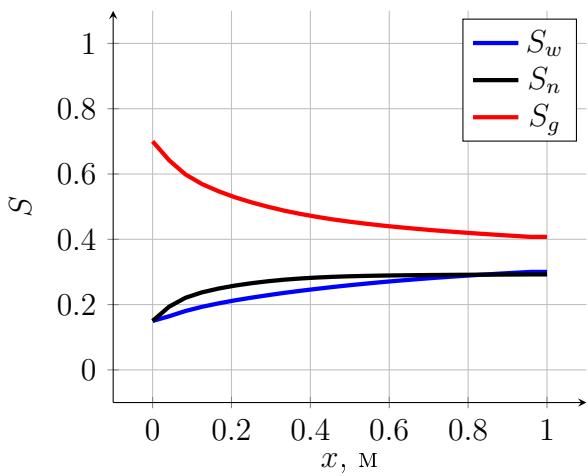


Рис. 5.9. Насыщенности в момент времени $t = 5000$ с, граничные условия 5.3

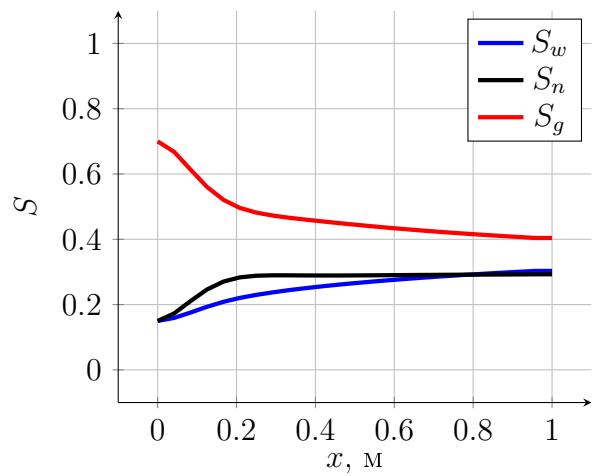


Рис. 5.10. Насыщенности в момент времени $t = 5000$ с, граничные условия 5.4

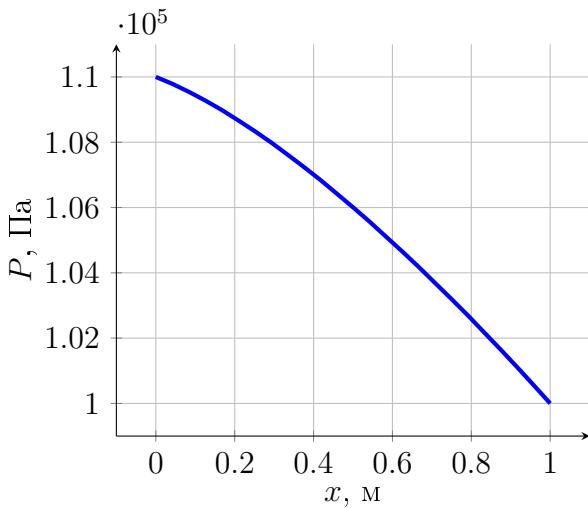


Рис. 5.11. Давление P_w в момент времени $t = 5000$ с, граничные условия 5.3

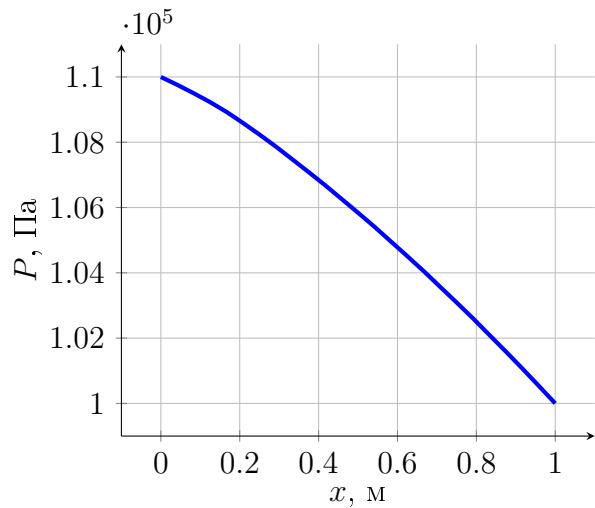


Рис. 5.12. Давление P_w в момент времени $t = 5000$ с, граничные условия 5.4

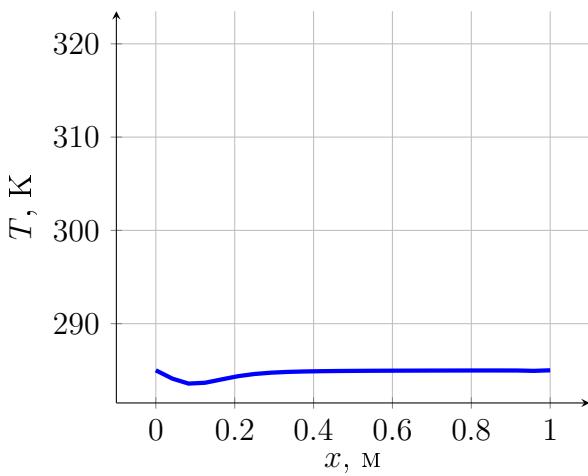


Рис. 5.13. Температура в момент времени $t = 5000$ с, граничные условия 5.3

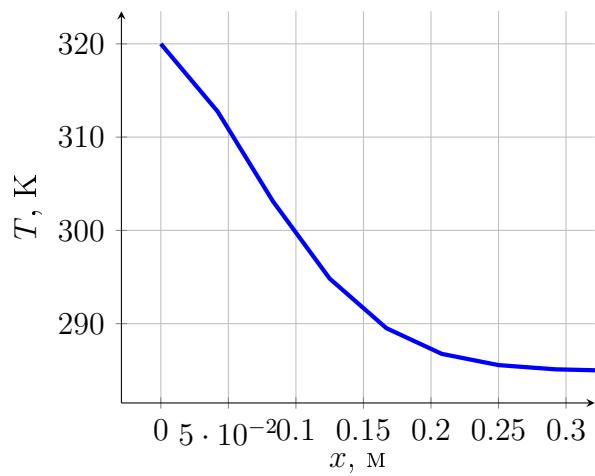


Рис. 5.14. Температура в момент времени $t = 5000$ с, граничные условия 5.4

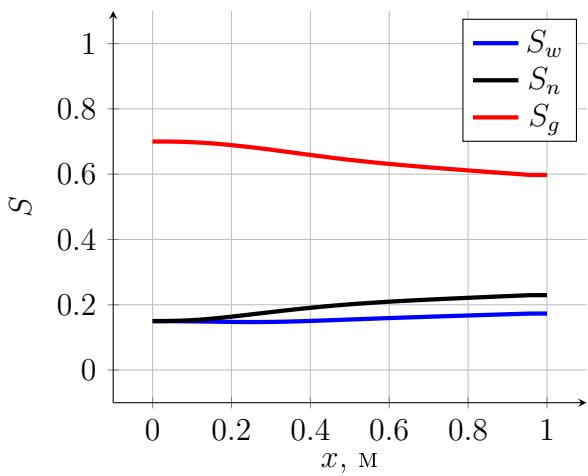


Рис. 5.15. Насыщенности в момент времени $t = 25000\text{с}$, граничные условия 5.3

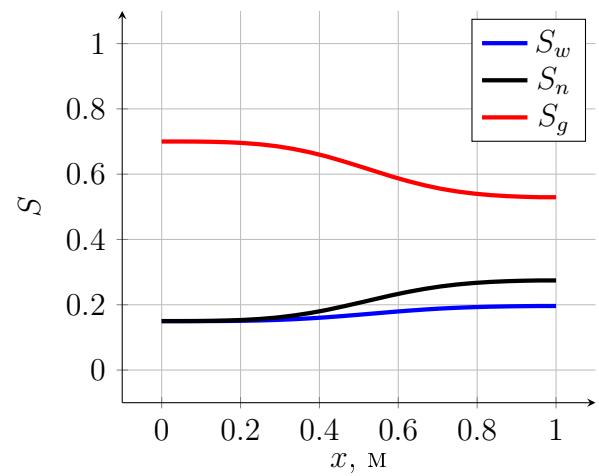


Рис. 5.16. Насыщенности в момент времени $t = 25000\text{с}$, граничные условия 5.4

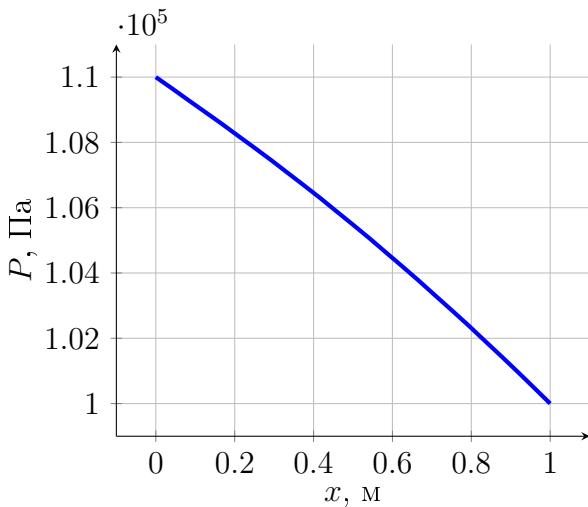


Рис. 5.17. Давление P_w в момент времени $t = 25000\text{с}$, граничные условия 5.3

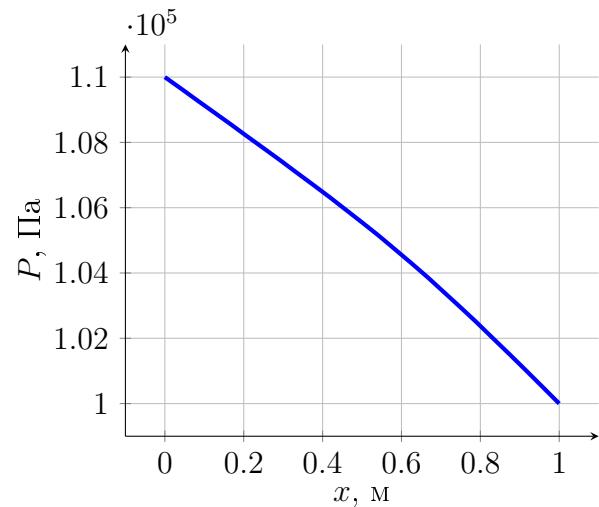


Рис. 5.18. Давление P_w в момент времени $t = 25000\text{с}$, граничные условия 5.4

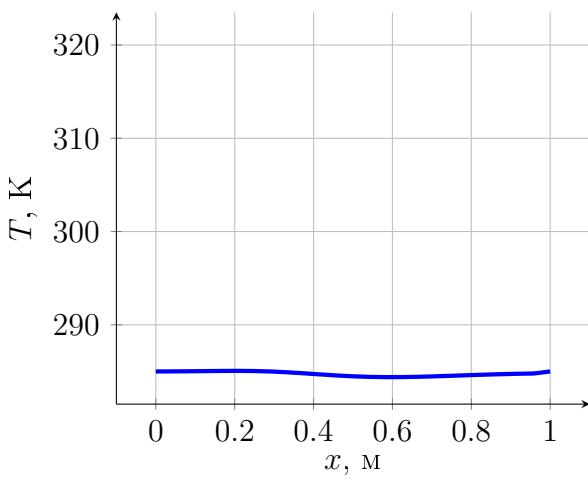


Рис. 5.19. Температура в момент времени $t = 25000\text{с}$, граничные условия 5.3

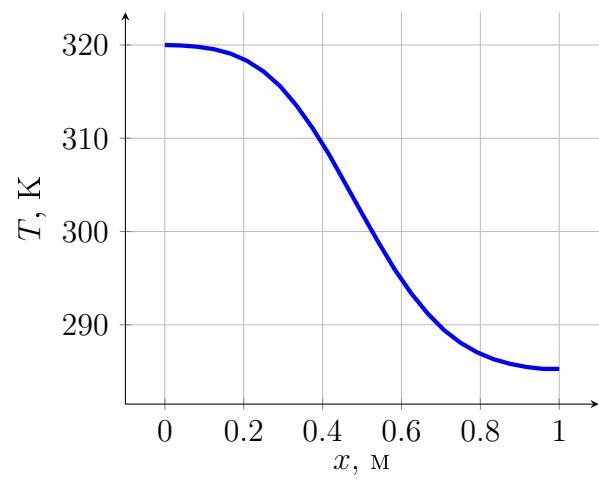


Рис. 5.20. Температура в момент времени $t = 25000\text{с}$, граничные условия 5.4

5.2 Задача о перераспределении фаз под действием силы тяжести в резервуаре

Рассмотрим одномерную модельную задачу неизотермической трехфазной фильтрации, где перераспределение фаз в резервуаре происходит под действием силы тяжести при нагревании верхней границы области. Резервуар сверху открыт, а его нижняя граница является непроницаемой для фаз. Считаем, что самый верхний слой почти полностью занят газом. Ось x направлена сверху-вниз.

Начальные условия:

$$S_w = 0.4, S_n = 0.3, S_g = 0.3, P_w = P_{\text{атм}}, T = 285K. \quad (5.5)$$

Границные условия:

$$\begin{aligned} T|_{x=0} &= 320K, \quad T|_{x=1} = 285K; \\ P_i|_{x=0} &= P_{\text{атм}}, \quad P_i|_{x=1} \text{ определяется из условия } \vec{u}_i|_{x=1} = 0; \\ S_w|_{x=0} &= 0.01, \quad \left. \frac{\partial S_w}{\partial x} \right|_{x=1} = 0; \\ S_n|_{x=0} &= 0.01, \quad \left. \frac{\partial S_n}{\partial x} \right|_{x=1} = 0. \end{aligned} \quad (5.6)$$

Результаты расчетов представлены в виде зависимостей профилей насыщенности фаз, распределения давления и температуры от времени на Рис. 5.21 - Рис. 5.26. На графиках можно видеть, как выравнивается давление, как распространяется тепловой фронт, а по изменению распределения насыщенностей – как происходит замещение одних фаз другими. Благодаря высокой подвижности и большой разнице в плотности по сравнению с другими фазами газ вытесняется в первую очередь, затем вода вытесняет нефть.

5.3 Двумерная задача просачивания с источником на границе

Рассмотрим двумерную область пористой среды, имеющую форму прямоугольника. В начальный момент времени водой занято лишь 10% области, а для нефти и газа задано периодическое распределение. Ось x направлена слева-направо, ось y – снизу-вверх. На левой границе в верхней половине области расположен источник воды. На правой границе задано постоянное значение давления. Через верхнюю, нижнюю границы, вторую половину левой границы может происходить просачивание. Учитывается влияние силы тяжести. Месторасположение источника нагревается. Пусть число расчетных узлов – $N_x \cdot N_y$.

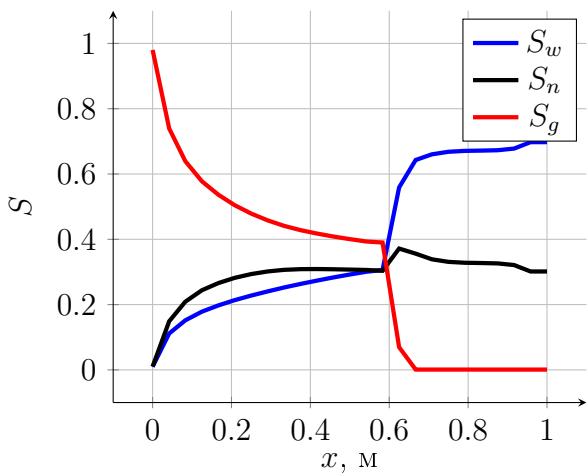


Рис. 5.21. Насыщенности в момент времени $t = 4000$ с

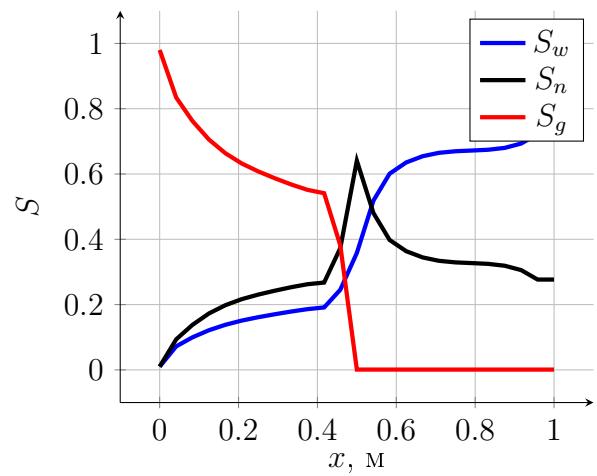


Рис. 5.22. Насыщенности в момент времени $t = 10000$ с

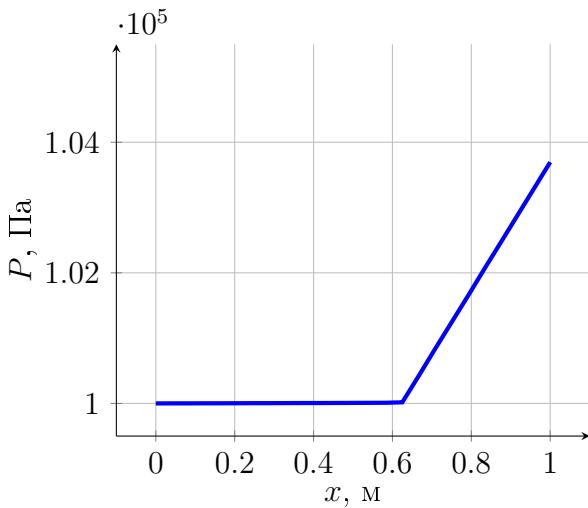


Рис. 5.23. Давление P_w в момент времени $t = 4000$ с

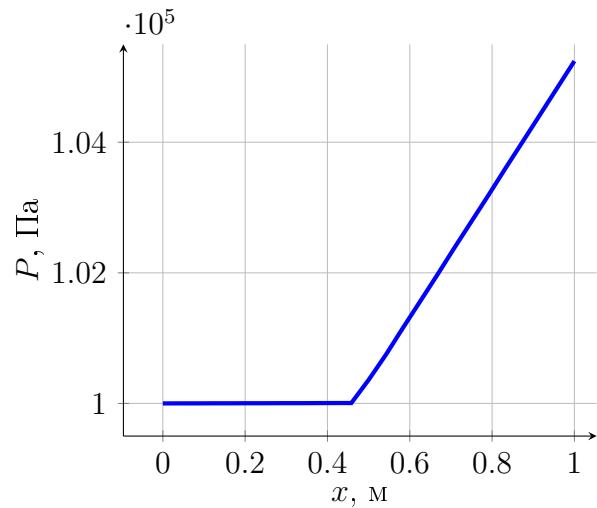


Рис. 5.24. Давление P_w в момент времени $t = 10000$ с

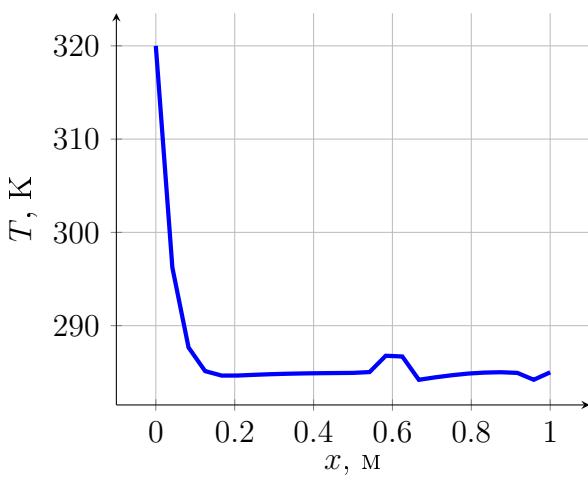


Рис. 5.25. Температура в момент времени $t = 4000$ с

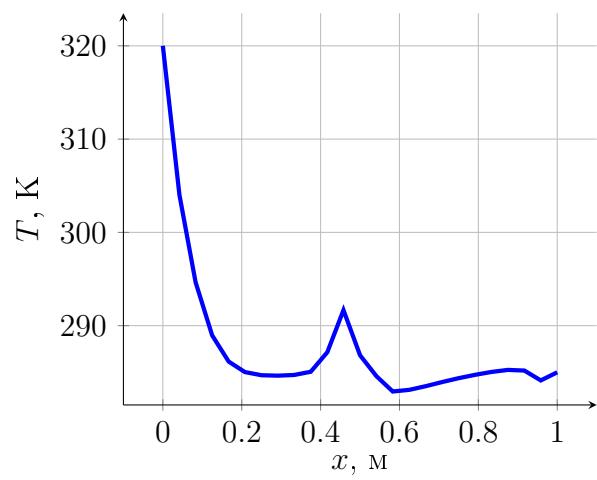


Рис. 5.26. Температура в момент времени $t = 10000$ с

Начальные условия:

$$\begin{aligned}
 S_w &= 0.1, \\
 S_n(x, y) &= 0.4 + 0.1 \cdot \sin^2(x \cdot N_x + y \cdot N_y), \\
 S_g(x, y) &= 0.4 + 0.1 \cdot \cos^2(x \cdot N_x + y \cdot N_y), \\
 P_w &= P_{\text{атм}}, \\
 T &= 285K.
 \end{aligned} \tag{5.7}$$

Границные условия:

$$\begin{aligned}
 T|_{x=0, 0.5 < y < 1} &= 320K, \quad \left. \frac{\partial T}{\partial x} \right|_{x=0, 0 < y < 0.5} = 0, \quad \left. \frac{\partial T}{\partial x} \right|_{x=1} = 0, \\
 \left. \frac{\partial T}{\partial y} \right|_{y=0} &= 0, \quad \left. \frac{\partial T}{\partial y} \right|_{y=1} = 0; \\
 P_w|_{x=0, 0.5 < y < 1} &= 1.1 \cdot P_{\text{атм}}, \quad \left. \frac{\partial P_w}{\partial x} \right|_{x=0, 0 < y < 0.5} = 0, \quad P_w|_{x=1} = P_{\text{атм}}, \\
 \left. \frac{\partial P_w}{\partial y} \right|_{y=0} &= 0, \quad \left. \frac{\partial P_w}{\partial y} \right|_{y=1} = 0; \\
 S_w|_{x=0, 0.5 < y < 1} &= 0.6, \quad \left. \frac{\partial S_w}{\partial x} \right|_{x=0, 0 < y < 0.5} = 0, \quad \left. \frac{\partial S_w}{\partial x} \right|_{x=1} = 0, \\
 \left. \frac{\partial S_w}{\partial y} \right|_{y=0} &= 0, \quad \left. \frac{\partial S_w}{\partial y} \right|_{y=1} = 0; \\
 S_n|_{x=0, 0.5 < y < 1} &= 0.15, \quad \left. \frac{\partial S_n}{\partial x} \right|_{x=0, 0 < y < 0.5} = 0, \quad \left. \frac{\partial S_n}{\partial x} \right|_{x=1} = 0, \\
 \left. \frac{\partial S_n}{\partial y} \right|_{y=0} &= 0, \quad \left. \frac{\partial S_n}{\partial y} \right|_{y=1} = 0.
 \end{aligned} \tag{5.8}$$

Результаты расчетов изображены на Рис. 5.27 - Рис. 5.41, приведены распределения давления, температуры и насыщенности трех фаз в три различные моменты времени. Можно видеть, что фронты источника распространяются по области и далее за ее границы, вода постепенно вытесняет нефть и газ.

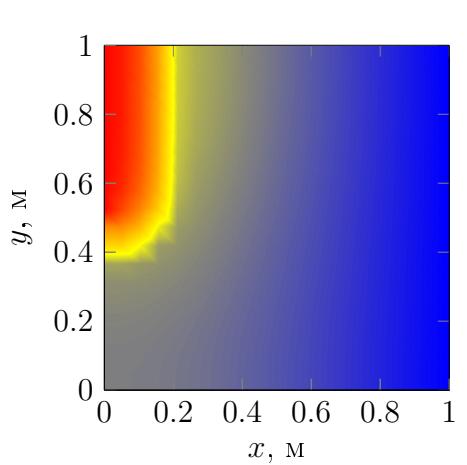


Рис. 5.27. Давление P_w в момент времени $t = 100\text{с}$

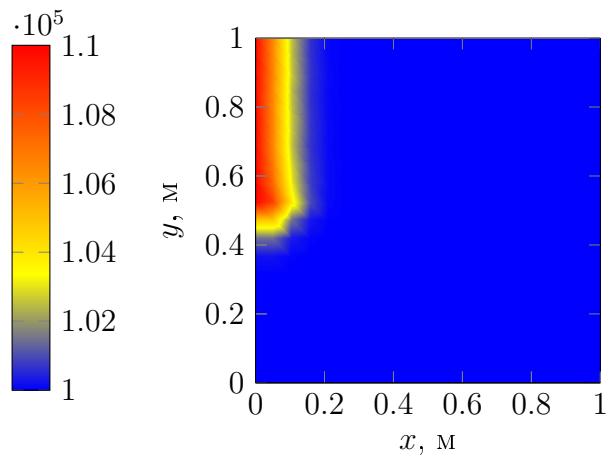


Рис. 5.28. Температура в момент времени $t = 100\text{с}$

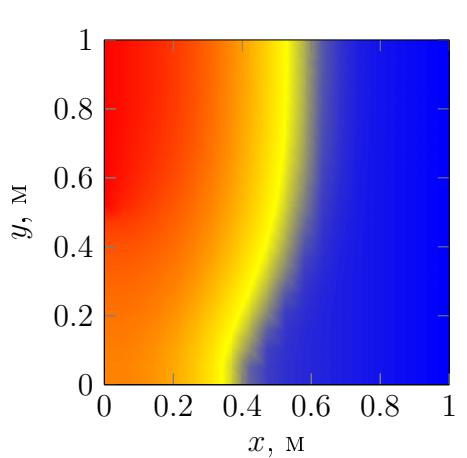


Рис. 5.29. Давление P_w в момент времени $t = 1000\text{с}$

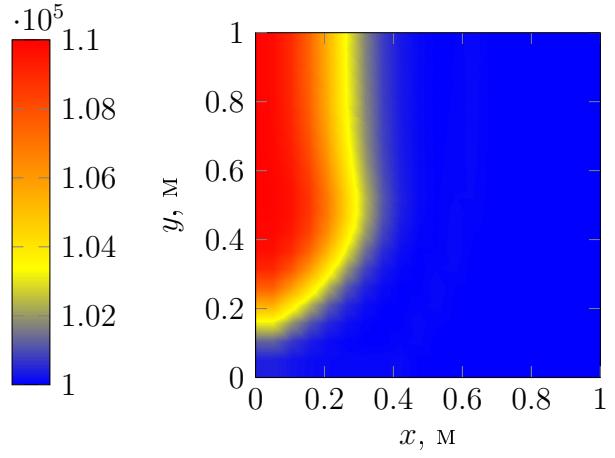


Рис. 5.30. Температура в момент времени $t = 1000\text{с}$

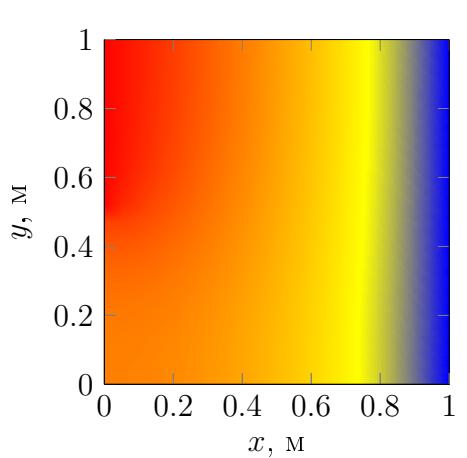


Рис. 5.31. Давление P_w в момент времени $t = 7000\text{с}$

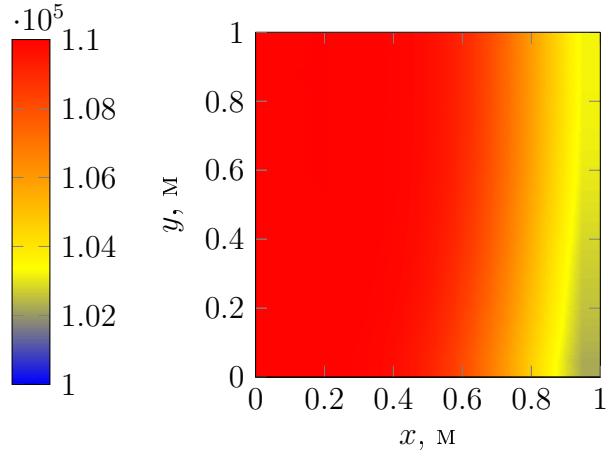


Рис. 5.32. Температура в момент времени $t = 7000\text{с}$

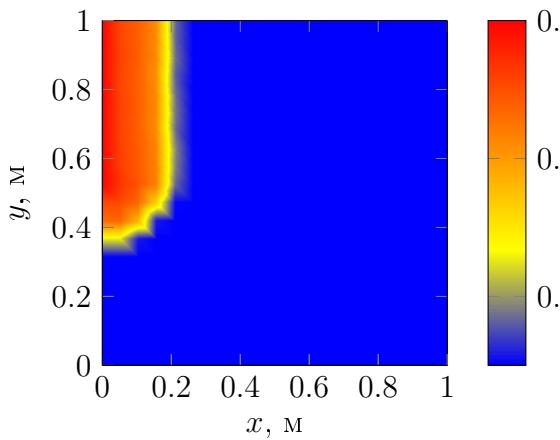


Рис. 5.33. Насыщенность S_w в момент времени $t = 100\text{c}$

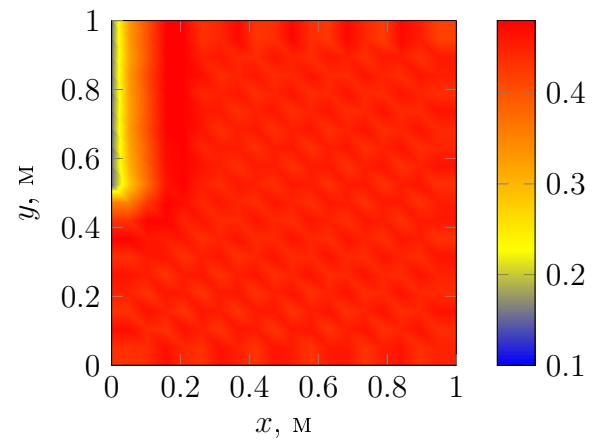


Рис. 5.34. Насыщенность S_n в момент времени $t = 100\text{c}$

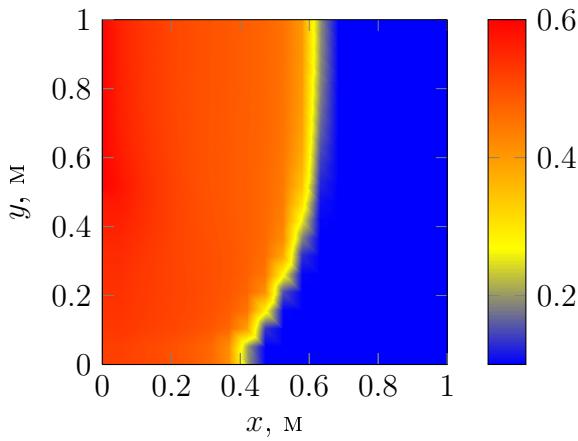


Рис. 5.35. Насыщенность S_w в момент времени $t = 1000\text{c}$

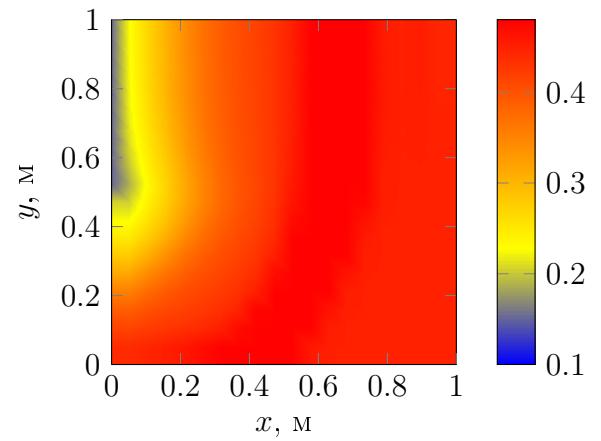


Рис. 5.36. Насыщенность S_n в момент времени $t = 1000\text{c}$

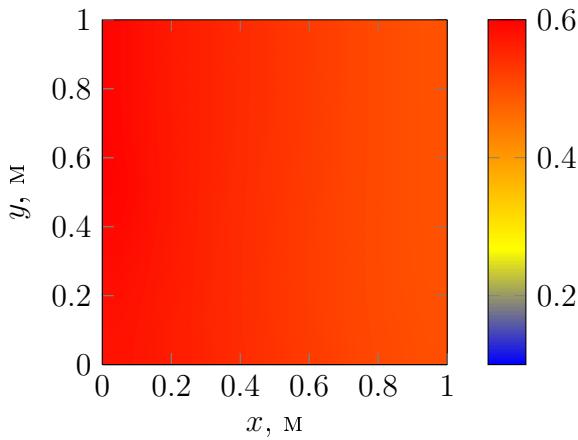


Рис. 5.37. Насыщенность S_w в момент времени $t = 7000\text{c}$

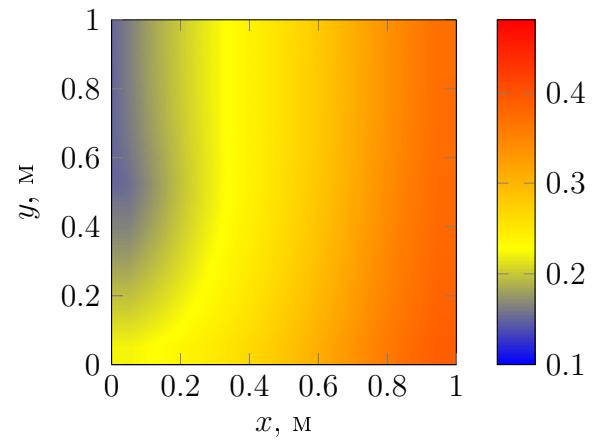


Рис. 5.38. Насыщенность S_n в момент времени $t = 7000\text{c}$

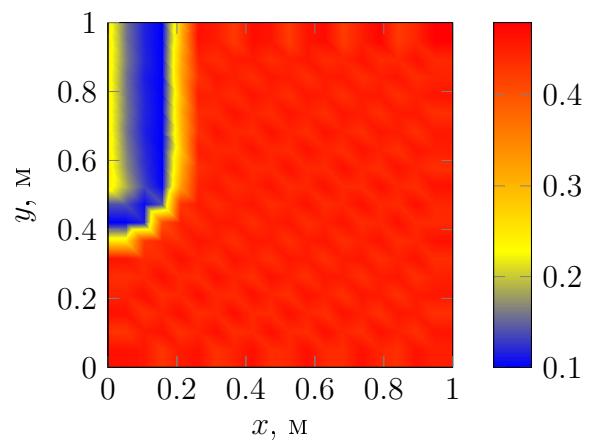


Рис. 5.39. Насыщенность S_g в момент времени $t = 100\text{с}$

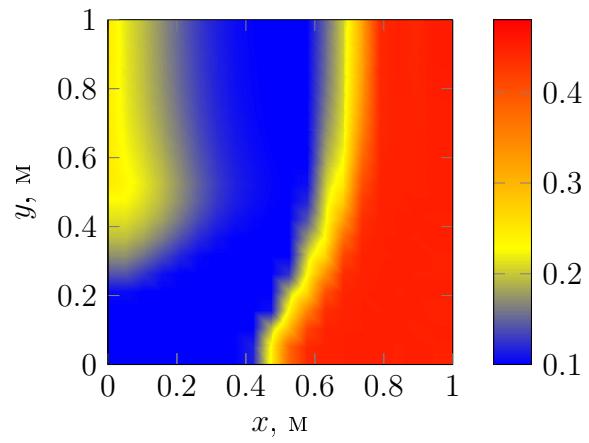


Рис. 5.40. Насыщенность S_g в момент времени $t = 1000\text{с}$

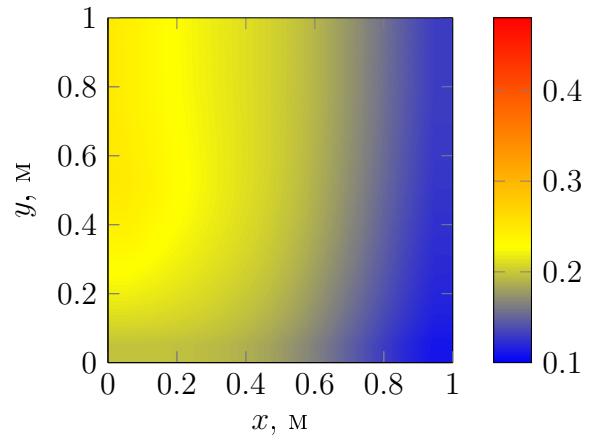


Рис. 5.41. Насыщенность S_g в момент времени $t = 7000\text{с}$

5.4 Трехмерная задача просачивания в резервуаре с источником на границе

Рассмотрим трехмерную область пористой среды, имеющую форму куба, верхняя поверхность открытая, нижняя грань ограничена резервуаром, через боковые грани может происходить просачивание. В начальный момент времени водой занято лишь 10% области, а для нефти и газа задано периодическое распределение. Ось y направлена снизу-вверх. На верхней грани в углу, расположена источник воды, он занимает девятую часть поверхности. Месторасположение источника нагревается. Просачивание происходит под действием силы тяжести. Для ускорения процесса просачивания значение ускорения свободного падения положено равным 98 м/с^2 . Пусть число расчетных узлов – $N_x \cdot N_y \cdot N_z$.

Начальные условия:

$$\begin{aligned} S_w &= 0.1, \\ S_n(x, y, z) &= 0.4 + 0.1 \cdot \sin^2(x \cdot N_x + y \cdot N_y + z \cdot N_z), \\ S_g(x, y, z) &= 0.4 + 0.1 \cdot \cos^2(x \cdot N_x + y \cdot N_y + z \cdot N_z), \\ P_w &= P_{\text{атм}}, \\ T &= 285K. \end{aligned} \quad (5.9)$$

Границные условия:

$$\begin{aligned} T|_{y=1, 0 < x < 0.33, 0 < z < 0.33} &= 320K, \\ P|_{y=1, 0 < x < 0.33, 0 < z < 0.33} &= P_{\text{атм}}, \\ P_i|_{y=0} \text{ определяется из условия } (\vec{u}_i \cdot \vec{n})|_{y=0} &= 0; \\ S_w|_{y=1, 0 < x < 0.33, 0 < z < 0.33} &= 0.6, \\ S_n|_{y=1, 0 < x < 0.33, 0 < z < 0.33} &= 0.05, \\ \text{на остальных границных поверхностях потоки} \\ T, S_w, S_n, P_w \text{ через нормаль равны нулю.} \end{aligned} \quad (5.10)$$

Результаты расчетов изображены на Рис. 5.42 - Рис. 5.61, приведены распределения давления, температуры и насыщенностей трех фаз в три различные моменты времени. Можно видеть, что фронты источника распространяются по области, около нижней границы происходит накопление воды, вода постепенно вытесняет нефть и газ через боковые и верхнюю грани.

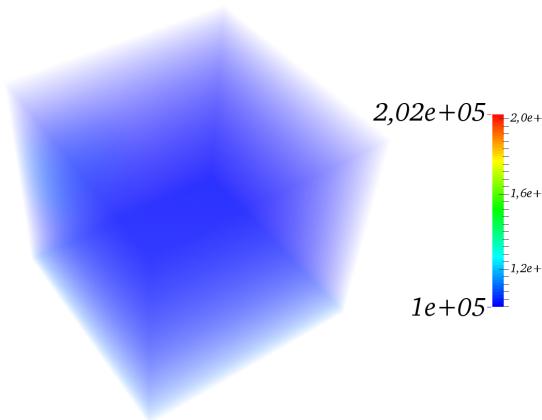


Рис. 5.42. Давление P_w в момент времени Рис. 5.43. Давление P_w в момент времени
 $t = 200\text{с}$ $t = 1000\text{с}$

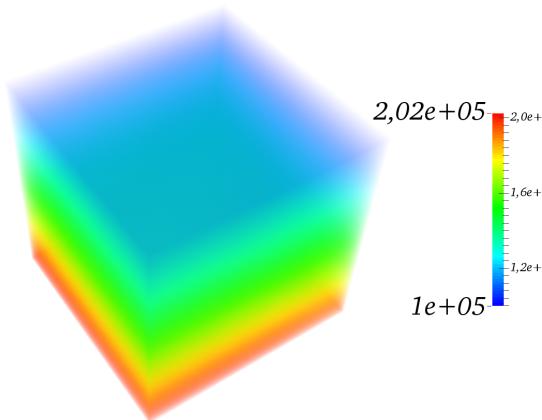
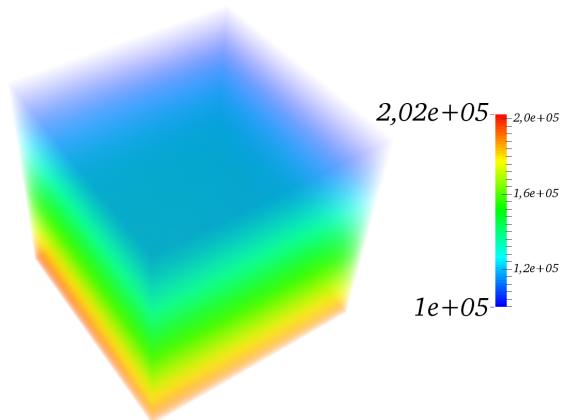
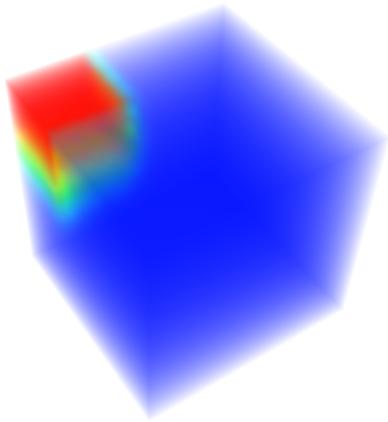
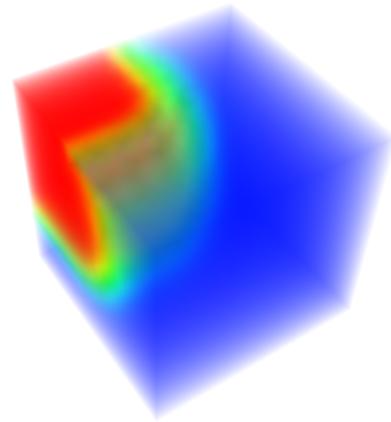


Рис. 5.44. Давление P_w в момент времени Рис. 5.45. Давление P_w в момент времени
 $t = 2000\text{с}$ $t = 6600\text{с}$

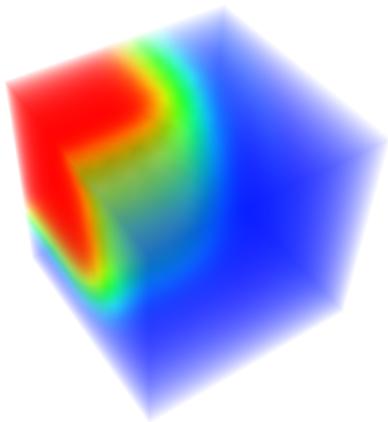


320
 $3,2e+02$
3,0e+02
284

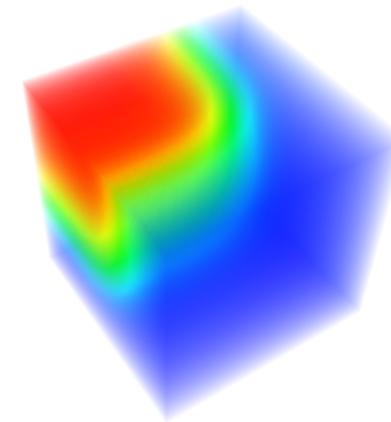


320
 $3,2e+02$
3,0e+02
284

Рис. 5.46. Температура в момент времени Рис. 5.47. Температура в момент времени
 $t = 200\text{c}$ $t = 1000\text{c}$



320
 $3,2e+02$
3,0e+02
284



320
 $3,2e+02$
3,0e+02
284

Рис. 5.48. Температура в момент времени Рис. 5.49. Температура в момент времени
 $t = 2000\text{c}$ $t = 6600\text{c}$

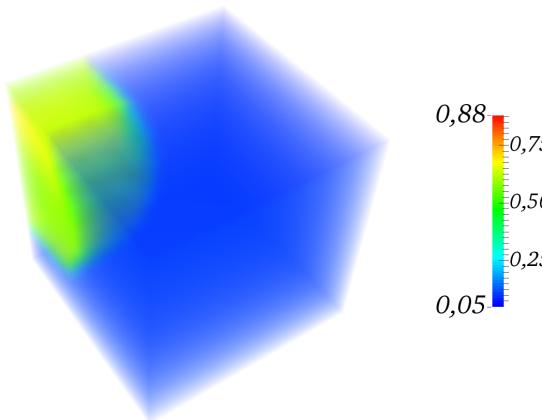


Рис. 5.50. Насыщенность S_w в момент време-
ни $t = 200\text{с}$

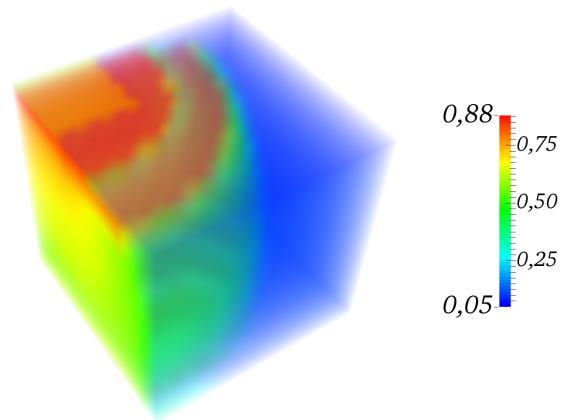


Рис. 5.51. Насыщенность S_w в момент време-
ни $t = 1000\text{с}$

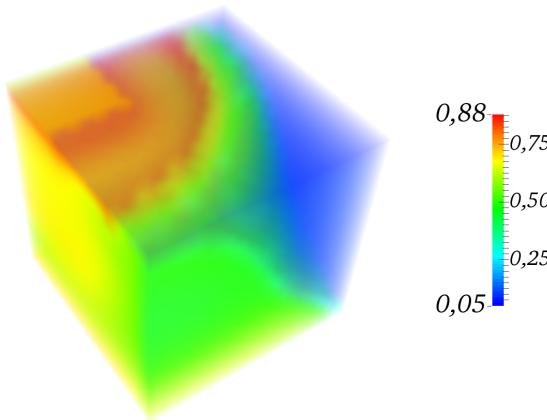


Рис. 5.52. Насыщенность S_w в момент време-
ни $t = 2000\text{с}$

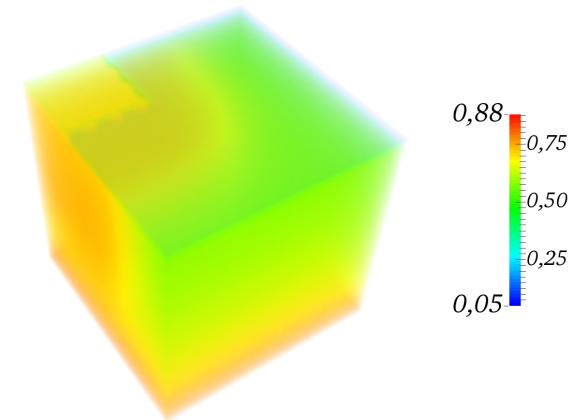


Рис. 5.53. Насыщенность S_w в момент време-
ни $t = 6600\text{с}$

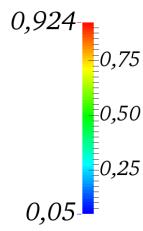
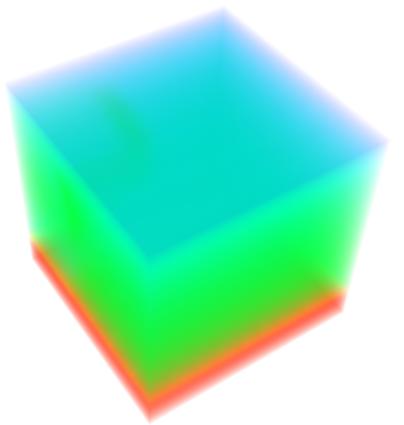


Рис. 5.54. Насыщенность S_n в момент вре-
мени $t = 200\text{с}$

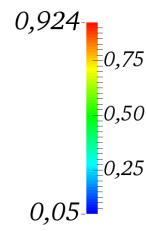
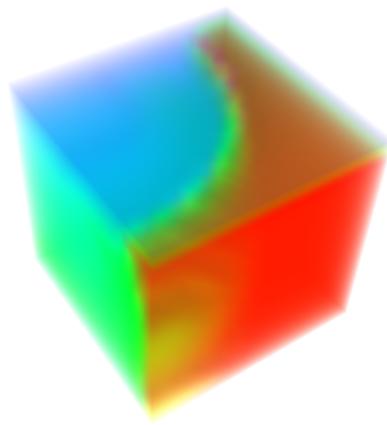


Рис. 5.55. Насыщенность S_n в момент вре-
мени $t = 1000\text{с}$

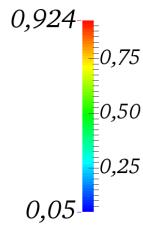
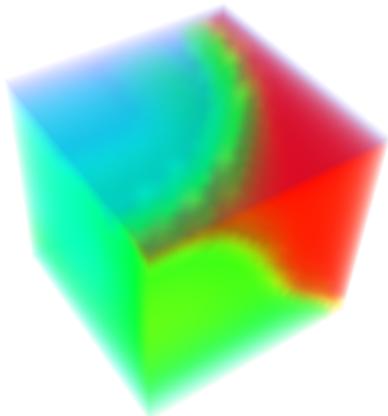


Рис. 5.56. Насыщенность S_n в момент вре-
мени $t = 2000\text{с}$

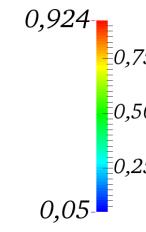
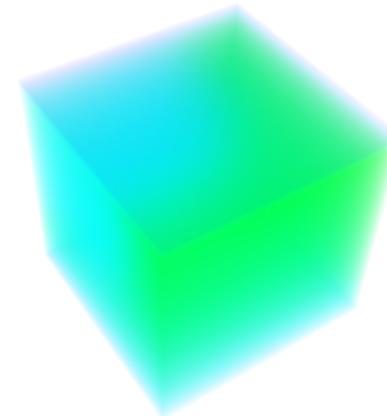


Рис. 5.57. Насыщенность S_n в момент вре-
мени $t = 6600\text{с}$

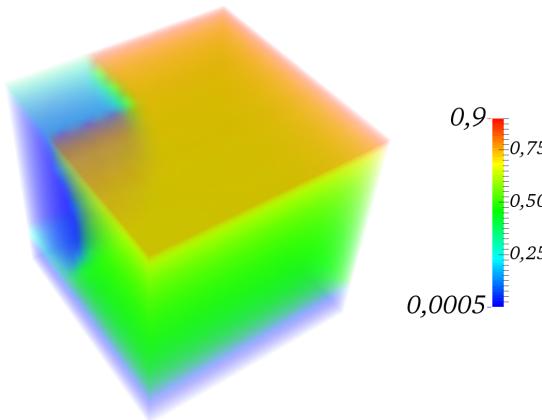


Рис. 5.58. Насыщенность S_g в момент време-
ни $t = 200\text{с}$

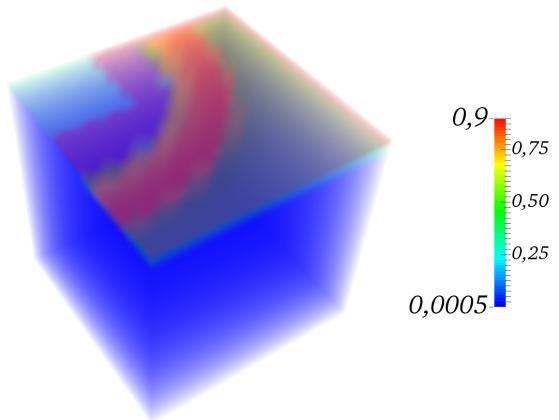


Рис. 5.59. Насыщенность S_g в момент време-
ни $t = 1000\text{с}$

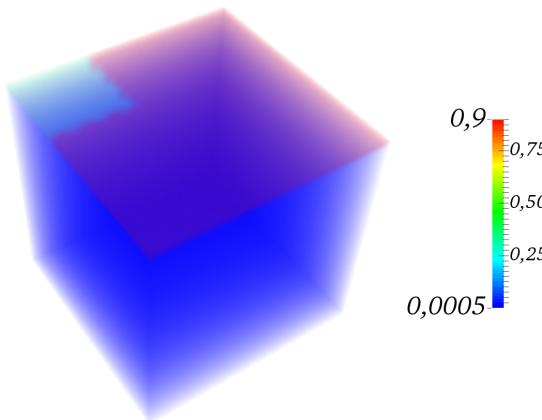


Рис. 5.60. Насыщенность S_g в момент време-
ни $t = 2000\text{с}$

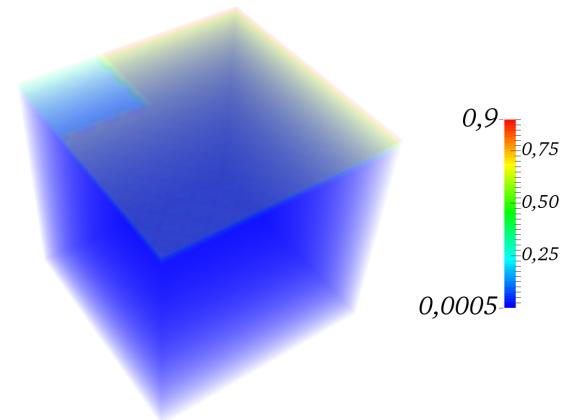


Рис. 5.61. Насыщенность S_g в момент време-
ни $t = 6600\text{с}$

6 Использование высокопроизводительных вычислительных систем

6.1 Вычислительные системы

Одним из наиболее интересных направлений развития современной вычислительной техники являются многопроцессорные системы. В настоящее время происходит фантастически быстрый рост мощности вычислительной техники, что дает реальную возможность приступить к моделированию тех задач, которые ранее были недоступны для численного решения. Но, с другой стороны, пришлось столкнуться с той ситуацией, когда высокопроизводительная многопроцессорная вычислительная техника используется лишь на небольшую часть своих потенциальных возможностей. Эта проблема в первую очередь связана с трудностями адаптации алгоритмов решения к архитектуре многопроцессорных ЭВМ с распределенной памятью.

Многопроцессорные системы можно разделить на 2 класса - системы с общей памятью и системы с распределенной памятью.

1. Системы с общей памятью. С точки зрения программиста привлекательно выглядят системы с общей памятью. Разбитая на взаимодействующие процессы (нити) программа в большинстве таких систем автоматически распределяется по доступным процессорам системы. Достаточно широкий круг последовательных алгоритмов может быть успешно адаптирован к функционированию на системах с общей, или, что тоже самое, разделяемой памятью. Однако у систем с общей памятью есть ряд существенных недостатков:

- относительно небольшое число процессоров;
- отсутствие возможности наращивания числа процессоров - масштабируемости;
- пиковая производительность систем с общей памятью ниже пиковой производительности систем с раздельной памятью;
- высокая, относительно аналогичных по производительности систем с раздельной памятью, стоимость.

Сказанное выше приводит к тому, что сконструированная система, как правило, не предусматривает возможности существенного наращивания числа процессорных узлов и обуславливает крайне высокую, относительно рассматриваемых систем с раздельной памятью, стоимость.

2. Системы с распределенной памятью. Масштабируемые системы массового параллелизма конструируют на основе объединения каналами передачи данных

процессорных узлов, обладающих своей локальной оперативной памятью, недоступной другим процессорам. Обмен данными между процессорами при таком подходе возможен лишь с помощью сообщений, передаваемых по каналам связи. Такая схема обладает рядом преимуществ по сравнению с системами, построеннымными на основе общей памяти. Подчеркнем основные преимущества систем с распределенной памятью:

- сравнительно низкая стоимость;
- масштабируемость (возможность построения систем требуемой производительности и наращивания их мощности за счет установки дополнительных процессоров).

Системы с раздельной памятью, по-видимому, всегда будут лидировать по показателю пиковой производительности, поскольку любые новые однопроцессорные (или многопроцессорные на основе общей памяти) системы могут быть легко объединены сетью и использованы в качестве многопроцессорных комплексов с раздельной памятью. Но, к сожалению, эффективное использование систем с распределенной памятью требует значительных усилий со стороны разработчиков прикладного обеспечения и возможно далеко не для всех типов задач. Для широкого круга хорошо зарекомендовавших себя последовательных алгоритмов не удается построить эффективные параллельные аналоги.

6.2 О технологии CUDA

В настоящее время графические процессоры (GPU – Graphics Processing Unit) являются оптимальной по соотношению цена-производительность параллельной архитектурой с общей памятью. Причина высокой производительности видеокарт в том, что они изначально были сконструированы для одновременного применения одной и той же шейдерной функции к большому числу пикселей, или, другими словами, для высокопроизводительных параллельных вычислений.

До недавнего времени использовать вычислительные мощности видеокарт было крайне неудобно, так как программисту необходимо было овладеть шейдерным языком программирования, соответствующим коду, выполняемому на GPU. Причем, в графических API полностью отсутствует возможность взаимодействия между параллельно обрабатываемыми пикселями, что в графике действительно не нужно, а для вычислительных задач оказывается довольно желательным. Ситуация изменилась, когда появились средства разработки GPGPU-приложений(General-Purpose computing on Graphics Processing Units). В качестве таких средств выступают CUDA, OpenCL, Dx11 Compute Shaders.

Предложенная компанией Nvidia технология CUDA (Compute Unified Device Architecture) – заметно облегчает написание GPGPU-приложений. Данная технология предназначена для разработки приложений для массивно-параллельных вычислений. Все программы пишутся на «расширенном» языке C, есть хорошая документация, инструменты, набор библиотек, поддерживается кроссплатформенность. CUDA является полностью бесплатной и уже приобрела широкую популярность в научных кругах [10].

CUDA строится на концепции, что GPU выступает в роли массивно-параллельного сопроцессора к CPU. При этом последовательный код выполняется на CPU, а для массивно-параллельных вычислений соответствующий код выполняется на GPU как набор одновременно выполняющихся нитей(потоков, threads).

Таким образом, GPU рассматривается как специализированное вычислительное устройство, которое:

- является сопроцессором к CPU;
- обладает собственной памятью;
- обладает возможностью параллельного выполнения огромного количества отдельных нитей.

В данной работе рассматривается применение технологии CUDA к задачам трехфазной фильтрации. Портирование кода происходило на базе примеров из [11].

6.3 Параллелизм

В данной работе в основу параллельной реализации положен принцип геометрического параллелизма. Геометрический параллелизм является одним из наиболее распространенных типов параллелизма, применяемых при решении задач, где имеется большое число однородных действий над однородными данными, которые можно так разбить на группы, что при обработке каждой группы не потребуется обращений к данным из других групп, за исключением их малого числа. Последнее свойство будем называть свойством локальности алгоритма. Локальный алгоритм допускает разбиение данных на части по числу процессоров, при котором обработку каждой части осуществляет соответствующий процессор. Этот подход весьма эффективен при условии, что действия, выполняемые одним процессором, зависят лишь от небольшого, ограниченного объема данных, расположенных на других процессорах. Желательно, чтобы эти «чужие» данные были локализованы на ограниченном, небольшом количестве процессоров. В настоящее время широкое распространение получил стандарт разработки параллельных программ MPI (Message Passing Interface - интерфейс передачи сообщений) для систем с разделенной памятью. В данной работе он используется для осуществления передачи сообщений между процессорами.

6.4 Практические аспекты реализации программного комплекса

6.4.1 Программный комплекс

Программный комплекс написан на языке программирования C/C++ с использованием библиотек MPI и CUDA в среде Eclipse CDT. Комплекс является кроссплатформенным и запускается в системах Windows и Linux, что позволяет использовать его на различных кластерах и пользовательских ПК.

6.4.2 Индексация массива данных

Для однородности способа обращения к данным независимо от размерности задачи используется одномерная индексация массивов. Например, обращение к элементу $T(i, j, k)$: $T[i + j * locN_x + k * locN_x * locN_y]$. $locN_x, locN_y, locN_z$ – размеры расчетной сетки, обрабатываемой данным процессором. Продемонстрируем универсальность данного подхода. Допустим, отсутствует измерение, соответствующее z , тогда $locN_z = 1$, k «пробегает» значения от 0 до $locN_z - 1$, т.е. в данном случае единственное возможное значение $k = 0$, при этом обращение к элементу становится идентичным $T[i + j * locN_x]$, т.е. двумерной индексации. Таким образом, учет размерности задачи происходит автоматически. Однако, в коде при вычислении, например, градиента функции следует все же проверять, что $locN_z > 2$ (если используется трехточечный шаблон), прежде чем проводить вычисления.

6.4.3 Сетка из процессоров

Расчетная область делится на подобласти в одном, двух или трех направлениях. Можно сказать, что процессоры образуют сетку, в которой местоположение каждого процессора может быть задано с помощью декартовых координат. Для наглядности приведем пример такого деления на Рис. 6.62. Дополнительно к стан-

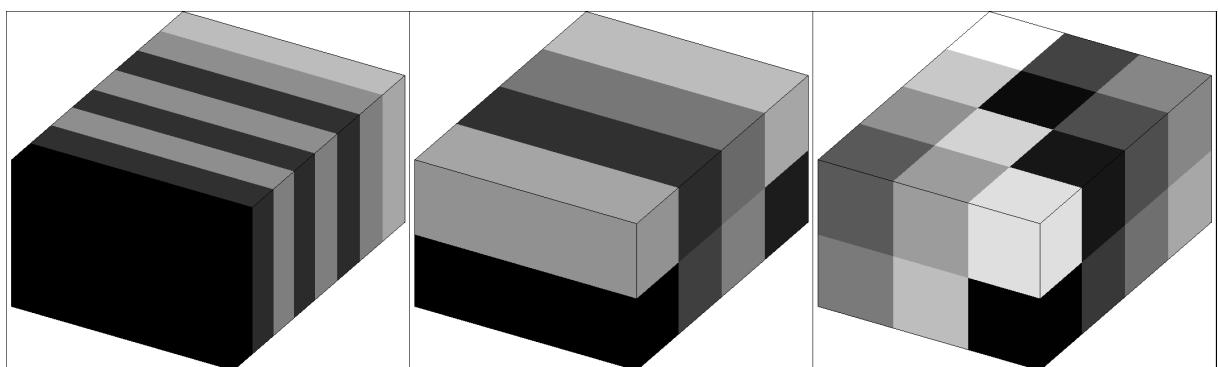


Рис. 6.62. Деление расчетной области на подобласти в одном, двух или трех направлениях

дартным для MPI понятиям $size$, $rank$ каждым из процессоров запоминаются значения $size_x$, $size_y$, $size_z$, $rank_x$, $rank_y$, $rank_z$ (аналоги $size$, $rank$ отдельно для каждого из направлений). $size_x$, $size_y$, $size_z$ определяются алгоритмом деления, причем $size_x \cdot size_y \cdot size_z = size$. Ранги определяются в Листинге 1, причем $rank_x + rank_y \cdot size_x + rank_z \cdot size_x \cdot size_y = rank$. Данные обозначения вводятся

```
rankx = rank % sizex;  
ranky = (rank / sizex) % sizey;  
rankz = (rank / sizex) / sizey;
```

Листинг 1. Определение координат процессора

для удобства реализации обменов данными и обращения к расчетной области.

6.4.4 Алгоритм первоначального распределения области по процессорам

Для того чтобы предварительно оценить время, необходимое для расчета, выяснить, насколько рационально деление области по различным измерениям, и эффективно распределить нагрузку между вычислительными узлами, был разработан и встроен в основной код программы алгоритм первоначального распределения области по процессорам.

В случае многопроцессорной реализации в общем времени вычислений можно выделить три основных составляющих: время непосредственно численного расчета, время межпроцессорных обменов и время загрузки/выгрузки данных на графический ускоритель (в случае гибридных вычислительных систем). Доля, которую вносит каждое из слагаемых, зависит как от вычислительной системы, так и от выбора конфигурации разбиения области. Чем больше направлений разбиения, тем меньше граничных узлов, но больше количество обменов на границах.

Предлагаемый алгоритм автоматического выбора оптимального разбиения состоит из трех основных шагов:

1. Найти эмпирически время, идущее на вычисления в одной расчетной точке на одном временном слое задачи. Следует заметить, что нахождение времени расчета аналитически в одной точке не является тривиальной задачей, а экспериментально засечь время расчета можно только для некоторого множества точек на нескольких слоях по времени. Для разного числа точек получаем набор времен и строим зависимость времени расчета некоторого числа точек от их количества. Для явных схем эту зависимость можно приближенно считать линейной. В этом случае имеет смысл говорить о времени расчета одной точки. Для неявных схем зависимость нелинейна, с увеличением числа точек растет время, условно приходящееся на расчет значения в одной точке.

2. Найти эмпирически время, идущее на коммуникации. Для нахождения времени пересылки между CPU, принадлежащими разным узлам кластера, измеряются времена пересылок для различного объема данных. Предполагается, что время пересылки состоит из времени инициализации соединения и времени непосредственно отправки-получения данных, которое линейно зависит от объема данных. Из полученных измерений можно найти данную линейную зависимость. Время копирования данных из памяти CPU в память GPU определяется аналогичным способом.
3. Решить задачу поиска минимума общего времени вычислений при ограниченном числе доступных вычислителей. Эту задачу можно решить, перебрав в цикле все возможные способы разбиения области при заданном количестве расчетных точек.

Рассмотрим пример применения данного алгоритма. Допустим, что нужно произвести расчеты на трехмерной сетке размером $500 \times 100 \times 50 = 2.5$ млн точек, при этом в распоряжении пользователя имеется 24 ядра CPU. Можно оценить, насколько сильно меняется общее время вычислений в зависимости от того, каким образом распределена расчетная сетка. На Рис. 6.63 представлены результаты измерений полученных времен расчета некоторой тестовой задачи трехфазного просачивания для различных конфигураций разбиения. Видно, что времена вычислений могут отличаться как на пять процентов, так и в полтора раза, что может быть очень существенно.

6.4.5 Локальная и глобальная системы координат

В работе представлен гибкий и хорошо масштабируемый механизм разделения расчетной сетки на локальные подсетки. Благодаря такому механизму нет необходимости хранить расчетную сетку целиком на каком-либо из процессоров, а преобразование локальных координат в глобальные позволяет воссоздать общую картину полученных в результате расчетов данных. Используем следующие обозначения: $locN_x, locN_y, locN_z$ – размеры расчетной (локальной) сетки, обрабатываемой данным процессором, N_x, N_y, N_z – размеры общей (глобальной) сетки. Разделение глобальной сетки на локальные происходит с помощью алгоритма, описанного в Листинге 2. Если сетка по выделенному измерению поровну не делится, то процессоры с $rank_x < N_x mod size_x$ получают нагрузку на один слой больше. Каждый процессор содержит дополнительный слой в массиве для обмена данными, при наличии «соседа» (если есть «соседи» с обеих сторон – два дополнительных слоя). Глобальные границы хранятся как обычные точки.

Переход от локальной системы координат к глобальной представлен в Листинге 3.

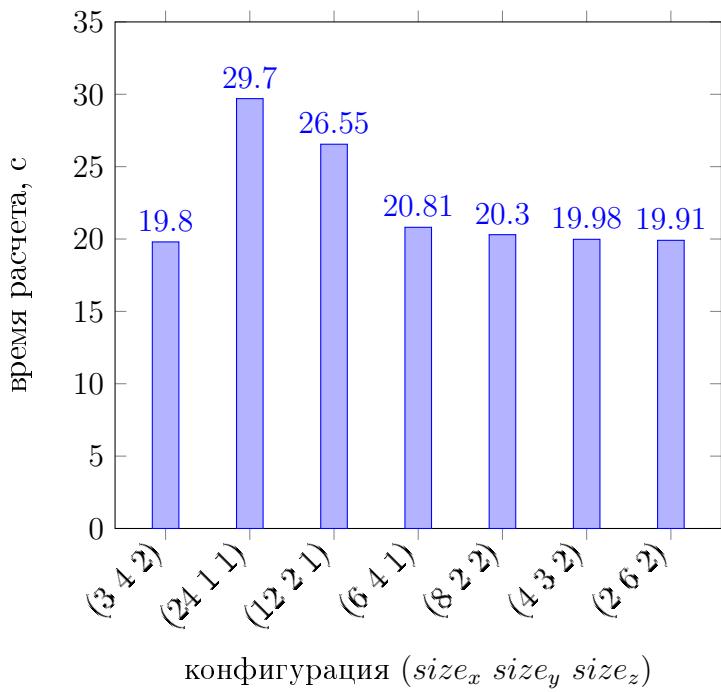


Рис. 6.63. Времена расчета тестовой задачи для различных конфигураций для 24 процессоров

```

locNx = Nx / sizex;
if (rankx < Nx % sizex) locNx++;
if (sizex > 1) {
    if ((rankx == 0) || (rankx == sizex - 1)) locNx++;
    else locNx += 2;
}
locNy = Ny / sizey;
if ((ranky < Ny % sizey)) locNy++;
if (sizey > 1) {
    if ((ranky == 0) || (ranky == sizey - 1)) locNy++;
    else locNy += 2;
}
locNz = Nz / sizez;
if (rankz < Nz % sizez) locNz++;
if (sizez > 1) {
    if ((rankz == 0) || (rankz == sizez - 1)) locNz++;
    else locNz += 2;
}

```

Листинг 2. Разделение расчетной сетки на подсетки

```

int global_index = local_index;
switch (axis)
{
case 'x':
    global_index += rankx * (Nx / sizex) + min(rankx, Nx % sizex) - min(rankx, 1);
    break;
case 'y':
    global_index += ranky * (Ny / sizey) + min(ranky, Ny % sizey) - min(ranky, 1);
    break;
case 'z':
    global_index += rankz * (Nz / sizez) + min(rankz, Nz % sizez) - min(rankz, 1);
    break;
default:
    print_error();
}
return global_index;

```

Листинг 3. Переход от локальной системы координат к глобальной

6.4.6 Организация межпроцессорных обменов

В первую очередь граничные данные для обмена помещаются в буферный массив. Затем посредством сообщений процессоры обмениваются содержимым буферов. Процессоры условно делятся на четные и нечетные отдельно по всем трем направлениям пересылки. По каждому направлению процессоры обмениваются в два этапа: от четных к нечетным и от нечетных к четным. Таким образом, полный обмен граничными данными со всеми «соседями» происходит за шесть итераций. Реализация обменов описанным способом приводится в Листингах 4, 5. Обмен завершается копированием полученных данных из буфера в локальную расчетную область процессора.

При проведении расчетов на GPU к обмену сообщениями и использованию буфера на хосте добавляется копирование данных для обмена из памяти видеокарты в память хоста.

6.4.7 Понятие активной точки

По способу обработки узлы расчетной сети условно можно разделить на:

- активные (значения в которых вычисляются текущим процессором);
- вспомогательные (служащие для хранения данных, полученных от других процессоров).

```

void send_recv(double* arr, char axis, double* buf, int size, int dst_rank, int id) {
    MPI_Status status;
    load_exchange_data_part(arr, axis);
    MPI_Sendrecv_replace(buf, size, MPI_DOUBLE, dst_rank, id, dst_rank, id + 1,
                         MPI_COMM_WORLD, &status);
    save_exchange_data_part(arr, axis);
}

void recv_send(double* arr, char axis, double* buf, int size, int dst_rank, int id) {
    MPI_Status status;
    load_exchange_data_part(arr, axis);
    MPI_Sendrecv_replace(buf, size, MPI_DOUBLE, dst_rank, id + 1, dst_rank, id,
                         MPI_COMM_WORLD, &status);
    save_exchange_data_part(arr, axis);
}

```

Листинг 4. Реализация межпроцессорного двустороннего обмена

Для определения, к какому из типов относится данный узел, используется код, представленный в Листинге 6.

6.4.8 Ввод/Вывод данных

Сохранение данных занимает важное место в моделировании физических процессов. Как отмечено в [12], ввод и вывод исходных данных с внешних дисковых накопителей может осуществляться, как минимум, двумя разными способами:

- ввод/вывод данных выполняет один из процессоров, называемый в дальнейшем «управляющий»;
- ввод/вывод данных выполняется непосредственно каждым процессором.

Несмотря на большую универсальность первого способа, выбран второй способ, так как он является более подходящим в случае обработки данных, объем которых превышает оперативную память управляющего процессорного узла. В этом случае возможны также, по крайней мере, два варианта:

- запись всеми процессорами одного большого файла – каждый процессор пишет свой фрагмент этого файла;
- запись каждым процессором отдельного файла – общее число файлов при этом соответствует числу процессоров.

```

void exchange_direct(double* arr, char axis, double* buf) {
    switch(axis) {
        case 'x': if(sizex > 1) {
            if (rankx % 2 == 0) {
                if (rankx != sizex - 1) send_recv(arr, axis, buf, locNy * locNz, rank + 1, 500);
                if (rankx != 0) recv_send(arr, axis, buf, locNy * locNz, rank - 1, 502);
            } else {
                if (rankx != 0) recv_send(arr, axis, buf, locNy * locNz, rank - 1, 500);
                if (rank) != sizex - 1) send_recv(arr, axis, buf, locNy * locNz, rank + 1, 502);
            }
        } break;
        case 'y': if(sizey > 1) {
            if (ranky % 2 == 0) {
                if (ranky != sizey - 1) send_recv(arr, axis, buf, locNx * locNz, rank + sizex, 504);
                if (ranky != 0) recv_send(arr, axis, buf, locNx * locNz, rank - sizex, 506);
            } else {
                if (ranky != 0) recv_send(arr, axis, buf, locNx * locNz, rank - sizex, 504);
                if (ranky != sizey - 1) send_recv(arr, axis, buf, locNx * locNz, rank + sizex, 506);
            }
        } break;
        case 'z': if(sizez > 1) {
            if (rankz % 2 == 0) {
                if (rankz != sizez - 1)
                    send_recv(arr, axis, buf, locNx * locNy, rank + sizex * sizey, 508);
                if (rankz != 0) recv_send(arr, axis, buf, locNx * locNy, rank - sizex * sizey, 510);
            } else {
                if (rankz != 0) recv_send(arr, axis, buf, locNx * locNy, rank - sizex * sizey, 508);
                if (rankz != sizez - 1)
                    send_recv(arr, axis, buf, locNx * locNy, rank + sizex * sizey, 510);
            }
        } break;
        default: break;
    }
}

```

Листинг 5. Организация межпроцессорных пересылок по всем направлениям

Для лучшей масштабируемости системы, удобства визуализации и обращения к данным выбран первый способ, но при этом возникает проблема синхронизации доступ-

```

if (((rankx != 0 && i == 0) || (rankx != sizex - 1 && i == locNx - 1 && Nx > 1))
    || ((ranky != 0 && j == 0) || (ranky != sizey - 1 && j == locNy - 1 && Ny > 1))
    || ((rankz != 0 && k == 0) || (rankz != sizez - 1 && k == locNz - 1 && Nz > 1)))
    return 0;
else
    return 1;

```

Листинг 6. Определение типа расчетного узла

па к файлу. Приведем пример кода, обеспечивающего корректную запись данных в файл многими процессорами в Листинге 7.

```

for (int cpx = 0; cpx < sizex; cpx++)
    for (int i = 0; i < Nx / sizex + 3; i++)
        for (int cpy = 0; cpy < sizey; cpy++)
            for (int j = 0; j < def.Ny / sizey + 3; j++)
                for (int cpz = 0; cpz < sizez; cpz++)
                {
                    barrier();
                    if ((rank == cpx + cpy * sizex + cpz * sizex * sizey)
                        && (i < locNx) && (j < locNy))
                    {
                        for (int k = 0; k < locNz; k++)
                            if (is_active_point(i, j, k))
                                print_plot_row(i, j, k);
                    }
                }

```

Листинг 7. Синхронизация вывода в файл между процессорами

6.5 Результаты расчетов на кластере

Основные расчеты проводились на кластере К-100 ИПМ им. М.В.Келдыша РАН, структура которого изображена на Рис 6.64. Ускорения и эффективность вычислений были подробно проанализированы на примере трехмерной тестовой задачи, описанной в соответствующем разделе. Время расчета измерялось с помощью счетчика, встроенного в программный код.

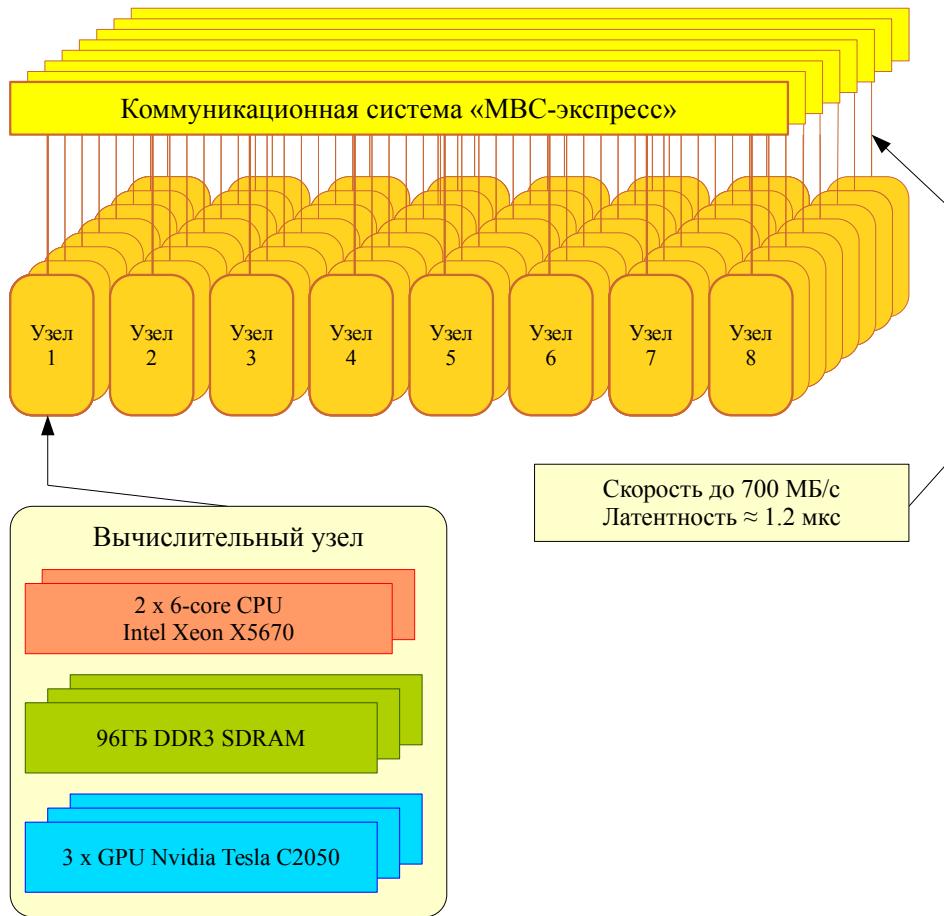


Рис. 6.64. Структура гибридного кластера K100

6.5.1 Результаты применения библиотеки MPI

Исследуем ускорения и эффективности вычислений в зависимости от числа используемых для расчета процессоров. Результаты измерений приведены на графиках (Рис. 6.65, Рис. 6.66). Измерялось время расчета 50 шагов по времени на сетке размером 4 миллиона узлов с помощью счетчика, встроенного в программный код. Расчетная область может быть поделена между CPU геометрически как в одном, так и в двух или трех направлениях. Ступенчатая структура ускорений и, следовательно, скачки эффективности наблюдаются в связи с неодинаковым способом распределения нагрузки между процессорами в зависимости от их числа, а также погрешностью вычислений. Сохранение высокой эффективности вычислений (84-95%) обусловлено значительным превосходством времени, требуемого непосредственно для расчета, по сравнению с временем, необходимым для обмена данными между процессорами.

6.5.2 Результаты использования технологии CUDA

Рассмотрим, как меняется время решения поставленной задачи при проведении расчетов на GPU с помощью CUDA. На Рис. 6.67 изображена зависимость укорения

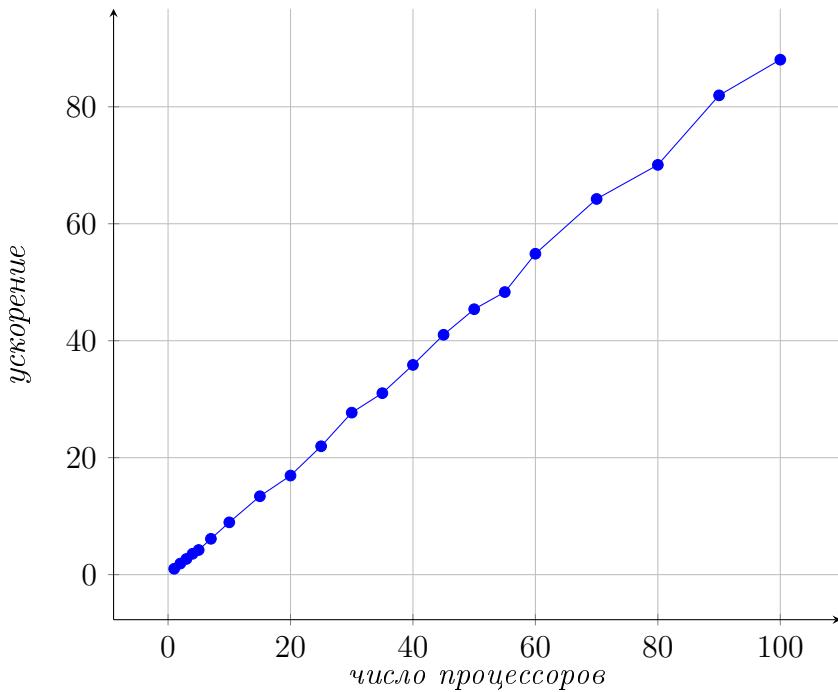


Рис. 6.65. Зависимость ускорения расчета тестовой задачи от числа процессоров

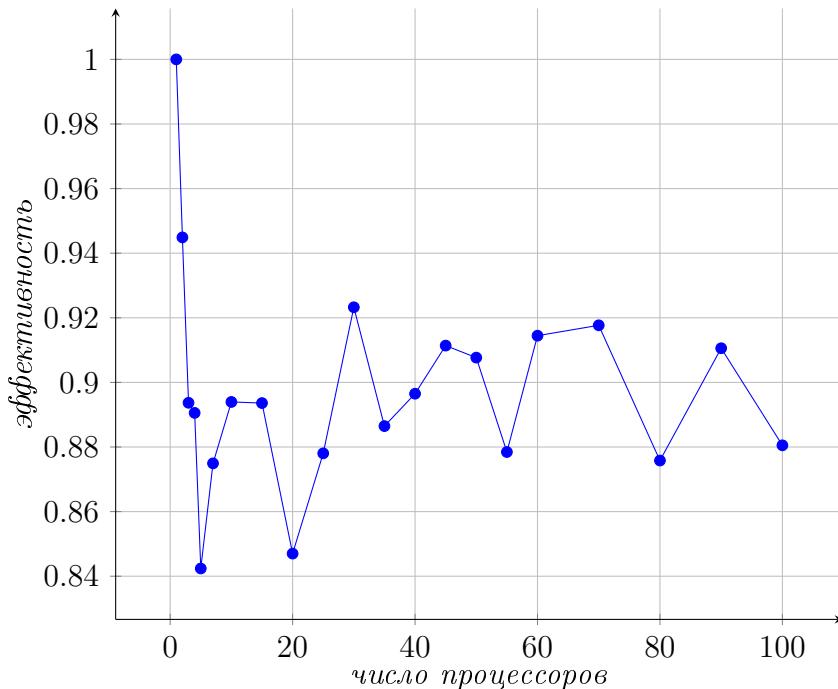


Рис. 6.66. Зависимость эффективности расчета тестовой задачи от числа процессоров

тестового расчета на одном GPU по сравнению с одним CPU. Измерения проводились для различного числа узлов расчетной сетки. Как видно из приведенного на Рис. 6.67 графика, для сеток малого размера (8000 узлов) возможно замедление расчетов, но при увеличении размера сетки наблюдается все большее ускорение расчета. При этом для большей из рассмотренных сеток (2.2 миллиона узлов) по-

лученное ускорение составило 3.5 раза. Небольшое ускорение обусловлено большой сложностью и вложенностью вычислений для каждого из расчетных узлов, так как для GPU доступ к памяти является трудоемкой задачей, а размер кеша мал. Вероятно, путем реорганизации программного кода можно достичнуть больших ускорений, но тогда усложнилась бы поддержка программы, так как стало бы больше отличий между кодом, написанным для CPU, и кодом для GPU, а также уменьшилась бы читаемость кода программного комплекса.

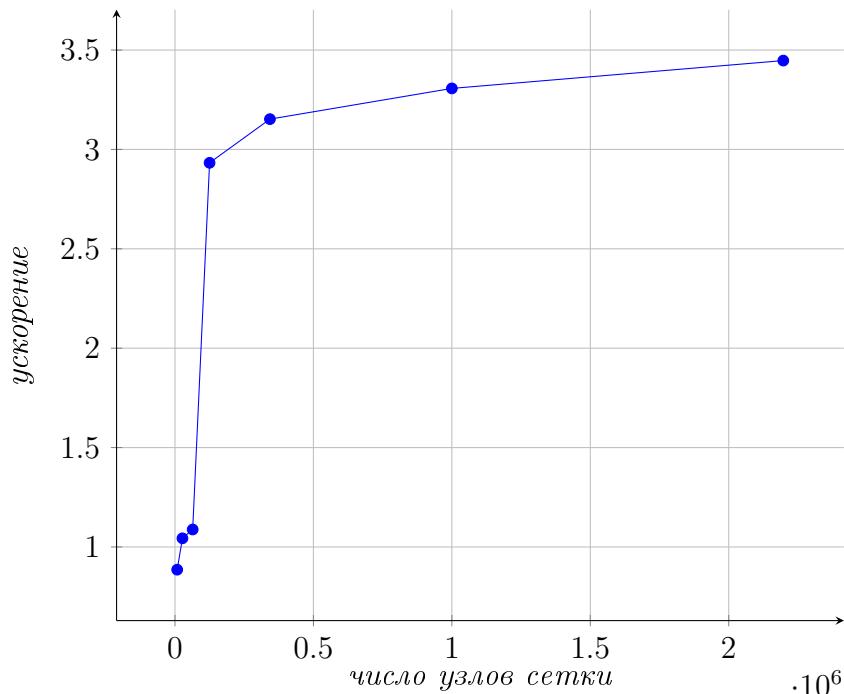


Рис. 6.67. Зависимость ускорения расчета тестовой задачи от числа узлов расчетной сетки на одном GPU по сравнению с одним CPU

6.5.3 Результаты совместного применения библиотек CUDA и MPI

Покажем на Рис. 6.68, как для расчетной сетки размером 8 млн узлов меняется время вычислений при использовании сразу нескольких видеокарт (от трех до двенадцати). Ускорение расчитывается по отношению к расчету на трех видеокартах. Выбрано именно три видеокарты, так как на каждом из узлов кластера расположено три GPU. Из приведенного графика можно видеть, что эффективность вычислений ниже, чем при использовании только MPI (см. Рис. 6.65). Этот эффект можно объяснить тем, что при обмене данными между процессорами необходимо провести копирование значений в необходимых узлах сетки не только в буфер CPU, но и в буфер GPU (для копирования между CPU и GPU) для дальнейшего обмена между CPU.

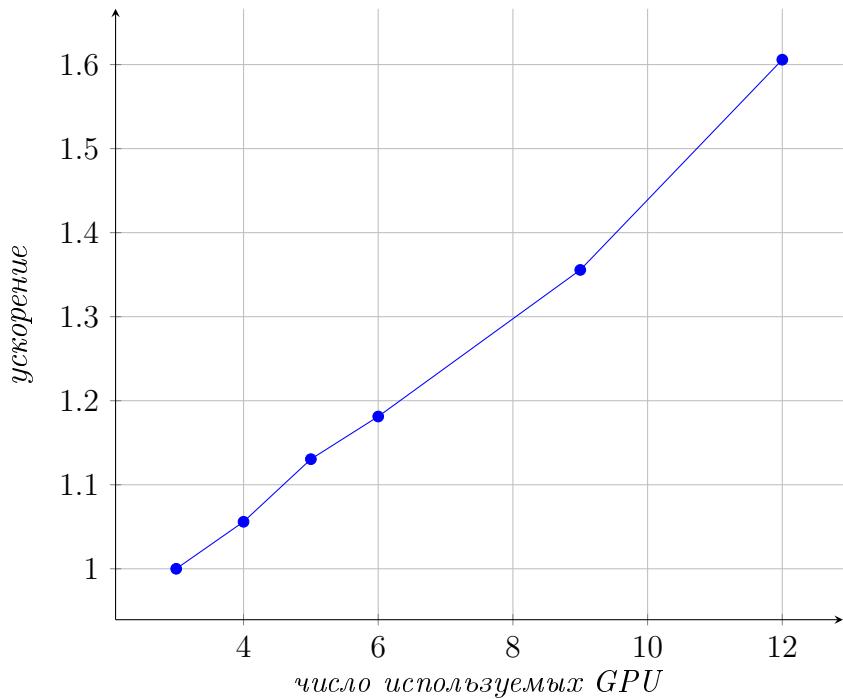


Рис. 6.68. Зависимость ускорения расчета тестовой задачи при использовании различного числа GPU по отношению к расчету на 3 GPU

В заключение следует отметить, что эффективное использование больших вычислительных мощностей является трудоемкой задачей и требует глубокого исследования и понимания архитектуры вычислительных систем.

7 Заключение

В процессе работы над дипломом:

- предложены 1D, 2D и 3D-модели трехфазных течений в пористых средах, допускающие реализацию явными численными методами;
- поставлены и решены модельные задачи просачивания в однородной пористой среде;
- разработан алгоритм расчета;
- алгоритм расчета распараллелен;
- написан программный комплекс на языке программирования C/C++ с использованием библиотек MPI, CUDA для решения задач трехфазной неизотермической фильтрации;
- в необходимом на данном этапе объеме освоена технология программирования на графических платах Nvidia CUDA;
- освоены технологии визуализации данных расчетов;
- программный комплекс протестирован на нескольких тестовых задачах фильтрации;
- сделаны четыре доклада на конференциях [1–4], две из которых – международные [3, 4];
- написаны в соавторстве две статьи [4, 5].

Список литературы

1. Р. Д. Каневская. Математическое моделирование гидродинамических процессов разработки месторождений углеводородов. Москва-Ижевск: Институт компьютерных исследований, 2003. С. 127.
2. Х. Азиз, Э. Сеттари. Математическое моделирование пластовых систем. М.: Недра, 1982. С. 407.
3. J. C. Parker, R. Lenhard, T. Kuppusami. A parametric model for constitutive properties governing multiphase flow in porous media // Water Resources Research. 1987. Vol. 23, no. 4. P. 618–624.
4. M. T. Van Genuchten. A closed-form equation for predicting the hydraulic conductivity of unsaturated soils // Soil Sci. Soc. Am. J. 1980. Vol. 44. P. 892–898.
5. Б. Н. Четверушкин, Л. В. Дородницын. Кинетически-согласованные схемы в газовой динамике // Математическое моделирование. 1999. Т. 11, № 5. С. 84–100.
6. Б. Н. Четверушкин. Кинетические схемы и квазигазодинамическая система уравнений. М.: Макс-пресс, 2004. С. 332.
7. Об одной схеме для решения задач фильтрации / Б. Н. Четверушкин, Д. Н. Морозов, М. А. Трапезникова [и др.] // Математическое моделирование. 2010. Т. 22, № 4. С. 99–109.
8. Использование явных схем для моделирования процесса двухфазной фильтрации / Д. Н. Морозов, М. А. Трапезникова, Б. Н. Четверушкин [и др.] // Математическое моделирование. 2011. Т. 23, № 7. С. 52–60.
9. Н. Н. Калиткин. Численные методы. СПб: СПб, БХВ - Петербург, 2011. С. 592.
10. А. В. Боресков, А. А. Харламов. Основы работы с технологией CUDA. М.: ДМК-Пресс, 2010. С. 232.
11. Д. Сандерс, Э. Кэндрот. Технология CUDA в примерах. Введение в программирование графических процессоров. М.: ДМК-Пресс, 2011. С. 232.
12. М. В. Якововский. Распределенные системы и сети. М.: МГТУ «Станкин», 2000. С. 118.

Список публикаций

1. Д. Н. Морозов, А. А. Люпа. Решение задач фильтрации на гибридных суперкомпьютерах // Труды 54-й научной конференции МФТИ. Управление и прикладная математика. Т. 2. М.: МФТИ, 2011. С. 18–19.
2. А. А. Люпа, Д. Н. Морозов. Алгоритм эффективного использования вычислительных мощностей гибридного кластера на примере задач фильтрации // Труды 55-й научной конференции МФТИ. Управление и прикладная математика. Т. 2. М.: МФТИ, 2012. С. 78–80.
3. Application of Explicit Schemes for the Simulation of the Two and Three Phase Filtration / D. N. Morozov, A. A. Lyupa, N. G. Churbanova et al. // Proceedings of the Third International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering. Stirlingshire, UK: Civil-Comp Press, 2013.
4. Simulation of Multiphase Flows in the Subsurface on GPU-based Supercomputers / M. A. Trapeznikova, N. G. Churbanova, A. A. Lyupa et al. // Parallel Computing: Accelerating Computational Science and Engineering (CSE). IOS Press, 2014. Vol. 25 of *Advances in Parallel Computing*. P. 324–333.
5. Моделирование трехфазной фильтрации явными методами на гибридных вычислительных системах / А. А. Люпа, Д. Н. Морозов, М. А. Трапезникова [и др.] // Математическое моделирование. 2014. Т. 26, № 4. С. 33–43.

Благодарности

Автор выражает благодарность научному руководителю Чурбановой Н.Г. за поддержку и чуткое руководство на протяжении трех лет, участие в ряде грантов и проектов, Трапезниковой М.А. за активное участие в научном руководстве, Савенкову Е.Б. за полезные обсуждения математических моделей фильтрации, Морозову Д.Н. за сотрудничество при разработке алгоритма и программного комплекса и советы по программированию, Подобной О.В., Прончевой Н.Г. и Галигузовой И.И. за участие и решение организационных вопросов. Следует отметить, что данная работа не могла бы состояться без предоставления вычислительных ресурсов ИПМ им. М.В. Келдыша РАН и работы сотрудников института, занимающихся разработкой архитектур вычислительных систем, программного обеспечения и поддержкой кластеров. Также автор благодарен преподавателям МФТИ(ГУ) и сотрудникам ИПМ им. М.В. Келдыша РАН за полезные и интересные лекции и семинарские занятия, привлечение к занятиям научной работой и предоставленные возможности, директора ИПМ им. М.В. Келдыша РАН, академика РАН Четверушкина Б.Н. за отличную организацию и поддержку.