

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра Прикладной математики
(полное название кафедры)

УТВЕРЖДАЮ

Зав. кафедрой

Соловейчик Ю.Г.

(фамилия, имя, отчество)

(подпись)

« » 2023 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

Аверкиной Ольги Андреевны

(фамилия, имя, отчество студента – автора работы)

Реализация численного метода моделирования фильтрации в керне с учетом

многокомпонентности фаз

(тема работы)

Факультет Прикладной математики и информатики

(полное название факультета)

Направление подготовки 01.03.02. Прикладная математика и информатика

(код и наименование направления подготовки бакалавра)

**Руководитель
от НГТУ**

Персова М.Г.

(фамилия, имя, отчество)

д.т.н., профессор

(ученая степень, ученое звание)

(подпись, дата)

**Автор выпускной
квалификационной работы**

Аверкина О.А.

(фамилия, И.О.)

ФПМИ, ПМ-92

(факультет, группа)

(подпись, дата)

Новосибирск, 2023 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра Прикладной математики
(полное название кафедры)

УТВЕРЖДАЮ

Зав. кафедрой

Соловейчик Ю.Г.
(фамилия, имя, отчество)

«21» марта 2023 г.

(подпись)

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ БАКАЛАВРА**

студенту Аверкиной Ольге Андреевне
(фамилия, имя, отчество студента)

Направление подготовки 01.03.02. Прикладная математика и информатика

Факультет Прикладной математики и информатики

Тема Реализация численного метода моделирования фильтрации в керне с учетом
многокомпонентности фаз

Исходные данные (или цель работы):

Разработать программу, реализующую численный метод моделирования процесса
фильтрации в керне с учетом многокомпонентности фаз.

Структурные части работы:

1. Изучение процесса фильтрации.

2. Изучение методов увеличения нефтеотдачи породы с помощью использования
полимеров и поверхностно-активных веществ(ПАВ).

3. Написание программы, реализующую численный метод моделирования процесса
фильтрации.

4. Исследование эффективности методов увеличения нефтеотдачи пласта с помощью использования полимеров и ПАВ.

Задание согласовано и принято к исполнению.

**Руководитель
от НГТУ**

Персова М.Г.
(фамилия, имя, отчество)

д.т.н., профессор
(ученая степень, ученое звание)

21.03.2023 г.
(подпись, дата)

Студент

Аверкина О.А.
(фамилия, имя, отчество)

ФПМИ, ПМ-92
(факультет, группа)

21.03.2023 г.
(подпись, дата)

Тема утверждена приказом по НГТУ № 1336/2 от «21» марта 2023 г.

ВКР сдана в ГЭК №____, тема сверена с данными приказа

(подпись секретаря экзаменационной комиссии по защите ВКР, дата)

(фамилия, имя, отчество секретаря экзаменационной комиссии по защите ВКР)

АННОТАЦИЯ

Отчёт 75 с., 4 ч., 19 рис., 7 табл., 12 источников, 3 прил.

**ДВУХФАЗНАЯ ФИЛЬТРАЦИЯ В ПОРИСТЫХ СРЕДАХ, МЕТОД
КОНЕЧНЫХ ЭЛЕМЕНТОВ, ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ, МЕТОДЫ
УВЕЛИЧЕНИЯ НЕФТЕОТДАЧИ ПЛАСТА**

Цель работы – разработка программы двухфазной фильтрации с учетом многокомпонентности фаз для исследования технологий повышения нефтеотдачи пласта, основанных на применении поверхностно-активных веществ и полимеров.

В результате реализована программа, моделирующая процесс двухфазной фильтрации с учётом многокомпонентности фаз, с помощью которой были исследованы методы увеличения нефтеотдачи пласта. Исследовано влияние концентрации ПАВ и полимеров на процесс нефтедобычи, а также рассмотрены разные режимы закачки ПАВ и полимеров в керн.

СОДЕРЖАНИЕ

| | |
|---|----|
| Введение..... | 6 |
| 1. Математическая модель и численный метод | 7 |
| 1.1. Математическая модель многофазной фильтрации в пористой среде..... | 7 |
| 1.2. Построение конечноэлементной аппроксимации функции давления..... | 10 |
| 1.3. Расчёт потока смеси через границы конечных элементов, шага по времени, перетекающих объёмов фаз и новых значений насыщенности на конечных элементах..... | 12 |
| 1.4. Расчёт новых значений вязкостей и количества остаточной нефти на конечных элементах..... | 15 |
| 2. Описание разработанной программы | 17 |
| 2.1. Основные сведения | 17 |
| 2.2. Структуры данных | 18 |
| 2.3. Модуль расчёта нового состояния ячеек | 19 |
| 3. Тестирование модуля расчёта давления | 26 |
| 4. Исследования..... | 27 |
| 4.1. Поиск оптимальных параметров | 27 |
| 4.2. Исследование методов повышения нефтеотдачи пластов..... | 31 |
| Заключение | 51 |
| Список литературы | 53 |
| Приложение 1. Код программы | 55 |
| Приложение 2. Фрагмент программы отрисовки графиков | 71 |
| Приложение 3. Данные о вязкости фазы воды в зависимости от концентрации полимера..... | 74 |

ВВЕДЕНИЕ

Для эффективной добычи нефти и газа в нефтедобывающей промышленности необходимо знать ряд факторов, которые влияют на процесс. К таким факторам относятся параметры флюидов: вязкость, плотность; а также параметры среды: пористость, структурная проницаемость, изначальные насыщенности породы флюидами и коэффициенты относительной фазовой проницаемости. Информацию о зависимости относительных проницаемостей флюидов от их насыщенностей, необходимую для моделирования процесса нефтедобычи на каждом конкретном месторождении, извлекают из простых экспериментов, сводящихся к одномерным модельным задачам, в которых исследуются различные параметры процесса фильтрации.

Традиционные методы увеличения нефтеотдачи для месторождений высоковязкой и тяжелой нефти нацелены на снижение вязкости и повышение подвижности нефти за счет прогрева пласта. В некоторых случаях применение тепловых методов не дает ожидаемого результата, например, если пласты тонкослоистые или глубокозалегающие. В настоящее время в качестве способа повышения эффективности процесса разработки нефтяных месторождений широкое применение находят химические методы воздействия на пласт. Проблему моделирования таких технологий мы рассмотрим в данной работе, а именно моделирование методов, основанных на закачке поверхностно-активных веществ (ПАВ) и полимеров. При использовании перечисленных технологий в пластовую систему закачиваются химически активные вещества, изменяющие свойства основных фильтрующихся фаз (нефти и воды). Моделирование таких процессов требует учета сложных компонентно-фазовых составов фильтруемых жидкостей и правил их изменения. От способов их учета в численном методе существенно зависит адекватность получаемых результатов.

В данной работе описан метод численного моделирования задачи двухфазной фильтрации для случая несжимаемых изотермических жидкостей.

1. МАТЕМАТИЧЕСКАЯ МОДЕЛЬ И ЧИСЛЕННЫЙ МЕТОД

1.1. МАТЕМАТИЧЕСКАЯ МОДЕЛЬ МНОГОФАЗНОЙ ФИЛЬТРАЦИИ В ПОРИСТОЙ СРЕДЕ

Рассмотрим математическую модель процесса многофазной фильтрации в пористых средах с учётом многокомпонентности фаз. Пористая среда характеризуется следующими параметрами: Φ – структурная пористость породы и K – структурная проницаемость породы.

Будем полагать, что в любой момент времени поры среды полностью заполнены многофазной смесью.

Модель фильтрации может быть однофазной, когда рассматривается процесс течения в пористой среде одной жидкости, и многофазной, когда в среде протекает не одна, а смесь из нескольких жидкостей. Под фазой в этом случае понимается жидкость, входящая в состав фильтруемой смеси, обладающая отличными от других фильтрационными свойствами. Каждая фаза смеси может состоять из нескольких компонент, причём одна компонента может входить в состав нескольких фаз. Пусть m -ая фаза смеси состоит из L^m компонент и пусть χ^{lm} – доля содержания l -ой компоненты в m -ой фазе. Каждая m -ая фаза смеси характеризуется параметрами: ρ^m – плотность (измеряется в $кг / м^3$), η^m – динамическая вязкость (измеряется в $Па \cdot с$), которые могут зависеть от компонентного состава фазы, и k^m – множитель к структурной проницаемости породы, который может зависеть от насыщенностей S^m фаз в среде.

Фильтрация представляет собой движение жидкости в пористой среде под действием перепада давления. Скорость фильтрации многофазного потока описывается законом Дарси:

$$\vec{u} = -K \sum_{m=1}^M \frac{k^m}{\eta^m} \text{grad}(P + P_c^m), \quad (1.1)$$

где \vec{u} – скорость фильтрации, M – количество фаз, P – давление, P_c^m – капиллярное давление фазы m . Записывая для каждой фазы закон сохранения массы, получаем систему уравнений:

$$-\text{div} \left(\rho^m K \frac{k^m}{\eta^m} \text{grad}(P + P_c^m) \right) = \frac{\partial}{\partial t} (\Phi \rho^m S^m) + f, m = \overline{1, M}, \quad (1.2)$$

где f – внутренний источник или сток для давления и равен 0, так как фазы несжимаемые. Отбор или нагнетание фазы в области переместили в краевые условия.

Будем считать, что моделируемый процесс фильтрации – изотермический, то есть температура является постоянной величиной. Фазы – это несжимаемые жидкости, поэтому значения их плотности ρ^m – постоянные величины. Тогда левую и правую части системы уравнений (1.4) можно разделить на ρ^m , а полученные уравнения сложить. Тогда, учитывая, что $\frac{\partial}{\partial t} \left(\Phi \sum_{m=1}^M S^m \right) = \Phi \frac{\partial}{\partial t} \left(\sum_{m=1}^M S^m \right) = \Phi \frac{\partial}{\partial t} \left(\sum_{m=1}^M 1 \right) = 0$, а капиллярное давление P_c^m будем считать настолько малым, что им можно пренебречь, получаем для давления P эллиптическую краевую задачу:

$$-\text{div} \left(\sum_{m=1}^M K \frac{k^m}{\eta^m} \text{grad} P \right) = 0, \quad (1.3)$$

$$P|_{\Gamma_1} = \theta, \quad (1.4)$$

$$P|_{\Gamma_2} = P^*. \quad (1.5)$$

Предварительно определив начальное распределение насыщенностей каждой фазы S^m , будем решать задачу (1.3) – (1.5) методом конечных элементов.

Для этого разобьём расчётную область Ω на ячейки. В каждой ячейке будем считать параметры среды и смеси постоянными. В результате конечноэлементной аппроксимации (использование метода конечных элементов для решения задачи (1.1) – (1.3) описано в п. 1.2) получаем в расчётной области Ω распределение поля давления.

По найденному конечноэлементному решению на каждой грани ячеек расчётной области Ω могут быть вычислены значения скорости фильтрации смеси \vec{u} по соотношению (1.3), по которым могут быть рассчитаны объёмы перетекающей смеси через грани ячеек¹ :

$$V_i = \left| \int_{\Gamma_i} \vec{u} \cdot \vec{n} d\Gamma \right| \Delta t = |Q_i| \Delta t, \quad (1.6)$$

где n – внешняя нормаль к грани Γ конечного элемента, следовательно, положительное значение интеграла будет соответствовать втекающему объёму смеси из элемента ($V_{e,in}$), а отрицательное – вытекающему ($V_{e,out}$).

Вычислив перетекающие объёмы смеси, можно рассчитать новые значения насыщенностей по следующей формуле:

$$S_e^m = \frac{mes(\Omega_e) S_e^m \Phi + V_{e,in}^m - V_{e,out}^m}{mes(\Omega_e) \Phi}, \quad (1.7)$$

где $V_{e,in}^m$ – втекающий, $V_{e,out}^m$ – вытекающий объём m -ой фазы из e -го конечного элемента, которые можно найти по формуле:

$$V_{e,\Gamma_i}^m = V_{e,\Gamma_i} \cdot \alpha_e^m, \quad (1.8)$$

¹ Подробнее о расчёте перетекающих потоков смеси изложено в п. 1.3

где α_e^m – множители, определяющие долю каждой из фаз в каждом конечном элементе.

Зная новое распределение насыщенностей и перетекающие объёмы фаз, можно рассчитать новые массовые доли χ_e^{lm} компоненты l в фазе m ². Далее, через массовую долю полимера в фазе воды $\chi_e^{polymer, water}$ рассчитать новые значения вязкости воды η_e^{water} , а через массовую долю ПАВ в фазе воды $\chi_e^{surfactant, water}$ новые значения остаточной насыщенности нефти $S_{e, residual}^{oil}$.

1.2. ПОСТРОЕНИЕ КОНЕЧНОЭЛЕМЕНТНОЙ АППРОКСИМАЦИИ ФУНКЦИИ ДАВЛЕНИЯ

Эллиптическую краевую задачу (1.3)-(1.5) будем решать относительно давления методом конечных элементов. Пусть функция $P = P(x)$ принадлежит гильбертову пространству H . Решать краевую задачу будем в слабой постановке Галёркина. Для этого потребуем выполнение условия ортогональности невязки решения уравнения (1.3) некоторому пространству пробных функций Υ :

$$\int_{\Omega} -\operatorname{div} \left(\sum_{m=1}^2 K \frac{k^m}{\eta^m} \operatorname{grad} P \right) v d\Omega = 0, \forall v \in \Upsilon. \quad (1.9)$$

Преобразуем выражение (1.9), применяя формулу Грина (интегрирование по частям), и с учётом краевых условий (1.3) получаем:

$$\int_{\Omega} -\operatorname{div} \left(\sum_{m=1}^2 K \frac{k^m}{\eta^m} \operatorname{grad} P \right) \operatorname{grad} v_0 d\Omega + \int_{\Gamma_1} \theta v d\Gamma_1 = 0, \forall v_0 \in H_0^1, \quad (1.10)$$

где H_0^1 – пространство функций, удовлетворяющих нулевым краевым условиям на границе Γ_1 , и которые имеют интегрируемые квадратом первые производные.

² Подробнее о расчёте массовых долей изложено в п. 1.4

Разобьём расчётную область Ω на конечные элементы Ω_e . Искать решение уравнения будем в проекции пространства H на его конечномерное линейное подпространство $V^n = \text{span}\{\Psi_1, \Psi_2, \dots, \Psi_N\}$, где в качестве Ψ_i выберем финитные базисные функции, ассоциированные с узлами конечноэлементной сетки. В этом случае функцию P можно представить в виде линейной комбинации базисных функций Ψ_i с весами q_i :

$$P(x) = \sum_{i=1}^N \Psi_i q_i. \quad (1.11)$$

Тогда, если подставить (1.11) в (1.10) и последовательно заменять пробную функцию v_0 базисными функциями Ψ_i , получим конечноэлементную СЛАУ относительно вектора весов q вида:

$$Aq = b, \quad (1.12)$$

где матрица A и вектор правой части b вычисляются по соотношениям:

$$\begin{aligned} A_{ij} &= \int_{\Omega} \left(K \sum_{m=1}^2 \frac{k^m}{\eta^m} \text{grad} \Psi_j \right) \text{grad} \Psi_i d\Omega, \\ b_i &= \int_{\Gamma_i} \theta \Psi_i d\Gamma_i. \end{aligned} \quad (1.13)$$

Краевые условия первого рода (1.5) учитываются в СЛАУ (1.13) заменами соответствующих строк системы равенствами искомым весов q_i значению краевого условия P^j в j -ом узле.

Задача решается в одномерной постановке. В качестве базисных функций используется линейный лагранжев базис.

Решать СЛАУ (1.12) будем численно с помощью метода LU-разложения[1].

1.3. РАСЧЁТ ПОТОКА СМЕСИ ЧЕРЕЗ ГРАНИЦЫ КОНЕЧНЫХ ЭЛЕМЕНТОВ, ШАГА ПО ВРЕМЕНИ, ПЕРЕТЕКАЮЩИХ ОБЪЁМОВ ФАЗ И НОВЫХ ЗНАЧЕНИЙ НАСЫЩЕННОСТИ НА КОНЕЧНЫХ ЭЛЕМЕНТАХ

Потоком смеси через границу Γ_i конечного элемента Ω_e будем называть интеграл

$$Q_i = \int_{\Gamma_i} \vec{u} n_{\Gamma_i} d\Gamma = \int_{\Gamma_i} \left(K \sum_{m=1}^2 \frac{k^m}{\eta^m} \text{grad} P \right) \cdot \vec{n}_{\Gamma_i} d\Gamma, \quad (1.14)$$

где n_{Γ_i} – нормаль к грани Γ_i . Нормаль к грани зафиксируем в направлении ОХ, тогда для каждого элемента внешняя нормаль слева будет противоположна нормали грани, а справа – совпадать с нормалью грани.

Отметим, что знак интеграла (1.14) всегда больше нуля, так как направление течения жидкости по отношению к элементам области всегда определено и происходит в направлении оси ОХ. Учитывая, что в функция давления представлена линейной комбинации базисных функций ψ_i с весами q_i , то рассчитать интеграл (1.14) по конечно-элементному решению, где Γ_i – конец отрезка (конечного элемента), можно по формуле:

$$Q_i = K \sum_{m=1}^2 \frac{k_i^m}{\eta_i^m} n_{\Gamma_i} d\Gamma. \quad (1.15)$$

Так как в данной работе рассматривается одномерная задача, которая предполагает, что у любой ячейки всего две проницаемые границы (та, через которую фаза втекает и та, через которую фаза вытекает из ячейки), то перетекающие потоки через границы конечных элементов будут равны начальному потоку, то есть:

$$Q_e = \theta \cdot s_{сеч} , \quad (1.16)$$

где $s_{сеч}$ – площадь сечения керна; из чего следует, что величины втекающего и вытекающего объёмов тоже будут равны:

$$V_{e,in} = V_{e,out} . \quad (1.17)$$

Объём m -ой фазы, вытекшей из элемента Ω_e через грань Γ_i за время Δt , вычисляется по соотношению:

$$V_{e, out}^m = \alpha_e^m \cdot |Q_e| \cdot \Delta t , \quad (1.18)$$

где α_e^m – множители, определяющие долю каждой из фаз в каждом конечном элементе, вычисляемые по формуле:

$$\alpha_e^m = \frac{k_e^m}{\eta_e^m \cdot \sum_{k=1}^2 \frac{k_e^k}{\eta_e^k}} . \quad (1.19)$$

При выборе шага по времени Δt нужно учитывать, что суммарный вытекающий объём фазы из ячейки не должен превышать имеющийся объём фазы в ней. Математически это можно записать так:

$$V_{e, out}^m \leq mes(\Omega_e) \Phi S_e^m . \quad (1.20)$$

Шаг по времени будем фиксировать одинаковый для всех элементов расчётной области. Вычислительно эффективнее находить шаг по времени будет по следующей формуле, которая не нарушает предыдущих рассуждений:

$$\Delta t = \min \left\{ \Delta t \mid \frac{mes(\Omega_e) \Phi S_e^m}{\alpha_e^m Q} \right\} , \quad (1.21)$$

где N – количество конечных элементов. Из условия (1.20) видно, что выбор шага по времени зависит от малости величины насыщенности фазы S_e^m , так как необходимо выбрать соответствующий малый шаг по времени, что может повлечь за собой резкое замедление моделирования процесса фильтрации.

Для предотвращения данной проблемы была введена специальная величина – минимальная критическая насыщенность фазы нефти S_{crit} , при которой насыщенность в ячейке настолько мала, что данная фаза будет вытесняться из ячейки в потоке вместе с фазой воды. Если $S_e^{oil} \leq S_{crit}$, то шаг исключается из множества (1.21). Величина минимальной критической насыщенности рассматривается только для фазы нефти потому, что в работе предполагается, что вязкость фазы воды всегда меньше вязкости фазы нефти, то есть фаза воды вытесняет фазу нефти, обратного случая быть не может.

Текущий объём фазы в ячейке можно найти как произведение общего порового пространства этой ячейки и текущей насыщенности фазы в этой ячейке:

$$V_{e, current}^m = mes(\Omega_e) \Phi S_e^m. \quad (1.22)$$

Чтобы получить новый объём фаз в каждой из ячеек, нужно к текущему объёму прибавить притёкший объём и вычесть вытекший:

$$V_{e, current}^m = V_{e, current}^m + V_{e, in}^m - V_{e, out}^m. \quad (1.23)$$

По новым вычисленным объёмам фаз в ячейках можно найти новые насыщенности фаз:

$$S_e^m = \frac{V_{e, current}^m}{mes(\Omega_e) \Phi}. \quad (1.24)$$

1.4. РАСЧЁТ НОВЫХ ЗНАЧЕНИЙ ВЯЗКОСТЕЙ И КОЛИЧЕСТВА ОСТАТОЧНОЙ НЕФТИ НА КОНЕЧНЫХ ЭЛЕМЕНТАХ

Рассчитав новые объёмы фаз и новые насыщенности, можно найти новые значения долей компонент фаз (фаза воды: вода, полимер, ПАВ, фаза нефти: нефть) по формуле:

$$\chi_e^{ml} = \frac{n_e^l M^l}{\sum_{j=1}^k (n_e^j M^j)}, \quad (1.25)$$

где M^l – молярная масса компоненты, k – количество компонент в фазе m ,
а n_e^l – количество вещества, вычисленное по формуле:

$$n_e^{ml} = \frac{\rho^m \cdot (V_{e-1, out}^m \chi_{e-1}^l + mes(\Omega_e) \Phi S_e^m \chi_e^l - V_{e, out}^m \chi_e^l)}{M^l}, \quad (1.26)$$

где ρ^m – плотность фазы.

Зная массовые доли, можем найти новые вязкости фазы воды на элементе η_y^{water} и новые значения остаточной нефти на элементе $S_{e, residual}^{oil}$. Линейно интерполировав дискретно заданную функцию вязкости³, зависящую от массовой доли полимера в фазе воды, получаем новую вязкость воды на элементе⁴. Для того, чтобы найти новые значения остаточной нефти на элементах, нужно знать соотношение количеств веществ нефти и ПАВ-вещества, необходимых для вступления в реакцию⁵.

По новым значениям насыщенности фаз и новым значениям остаточной насыщенности нефти можем вычислить новые коэффициенты k_e^m :

³ См. Приложение 3

⁴ Подробнее о расчёте вязкости изложено в п. 2.4.8

⁵ Подробнее о расчёте остаточной насыщенности изложено в п. 2.4.7

$$\begin{aligned}
k_e^{water} &= k_{\max}^{water} \cdot S^{\beta^{water}}, \\
k_e^{oil} &= k_{\max}^{oil} \cdot (1 - S^{\beta^{oil}}),
\end{aligned}
\tag{1.27}$$

$$S = \frac{S_e^{water} - S_{residual}^{water}}{S_{\max}^{water} - S_{e,residual}^{water}} = \frac{S_e^{water} - S_{residual}^{water}}{(1 - S_{e,residual}^{oil}) - S_{e,residual}^{water}},
\tag{1.28}$$

где $S_{e,residual}^m$ – значения остаточной насыщенности фазы m на элементах.

2. ОПИСАНИЕ РАЗРАБОТАННОЙ ПРОГРАММЫ

2.1. ОСНОВНЫЕ СВЕДЕНИЯ

Программа расчёта всех величин написана на языке C++, среда разработки: Microsoft Visual Studio 2022.

Входные данные программы:

1. Равномерная конечноэлементная сетка:
 - Длина керна.
 - Длина конечного элемента.
2. Сетка по времени:
 - Время фильтрации.
 - Шаг по времени.
3. Информация о свойствах материалов породы:
 - Структурная проницаемость породы.
 - Структурная пористость породы.
4. Информация о свойствах фильтруемых фаз и их компонент:
 - Вязкость нефти.
 - Молярные массы компонент.
 - Массовые доли закачиваемых компонент.
 - Плотной каждой фазы.
 - Таблица с данными о вязкости фазы воды в зависимости от концентрации полимера.
5. Плотность потока с одной стороны керна и значение давления с другой стороны керна.
6. Площадь сечения керна.
7. Начальное распределение насыщенных в области.
8. Значение минимальной критической насыщенности нефти в керне.
9. Значение остаточной насыщенности нефти в керне.

Выходные данные программы:

1. Поля давления по времени.
2. Информация о перетекающих фазах по времени (насыщенность, объёмы, остаточная насыщенность, вязкость фазы воды и тд.).

Модули разработанной программы:

1. Модуль расчёта распределения давления.
2. Модуль расчёта нового состояния ячеек.

Для создания графиков была разработана дополнительная программа на языке Python, среда разработки: PyCharm Community Edition 2022.3.

2.2. СТРУКТУРЫ ДАННЫХ

Структуры данных, используемые для описания состояния модели:

- `vector<double> m` – массив, содержащий молярные массы компонент (размерность равна количеству компонент);
- `vector<double> nu_water` – вязкость фазы воды на каждом конечном элементе (размерность равна количеству конечных элементов);
- `vector<double> density` – плотность фаз (размерность равна количеству фаз);
- `vector<double> k_max` – массив, содержащий максимальные значения структурной проницаемости для каждой из фаз (размерность равна количеству фаз);
- `vector<double> s_res_oil_vec` – массив, содержащий значения остаточной насыщенности нефтью на каждом элементе размер (размерность равна количеству конечных элементов);
- `vector<double> Q` – массив, содержащий значения перетекающих потоков через границы конечных элементов (размерность равна количеству граней конечных элементов);
- `vector<double> nodes` – массив, содержащий координаты узлов расчётной области (размерность равна количеству узлов расчётной области);

- `vector<double>` `pp` – массив, содержащий значения давления на каждом конечном элементе (размерность равна количеству конечных элементов);
- `vector<vector<double>>` `Q_m` – массив, содержащий значения перетекающих потоков фаз (размерность равна произведению количества фаз и количества граней конечных элементов);
- `vector<vector<double>>` `V_out` – массив, содержащий значения вытекающих объёмов фаз (размерность равна произведению количества фаз и количества граней конечных элементов);
- `vector<vector<double>>` `V_m` – массив, содержащий текущие значения объёмов фаз (размерность равна произведению количества фаз и количества конечных элементов);
- `vector<vector<double>>` `k` – массив, содержащий множители к структурной проницаемости для каждой фазы (размерность равна произведению количества фаз и количества узлов конечных элементов);
- `vector<vector<double>>` `alfa` – массив, содержащий значения долей вытекающих фаз (размерность равна произведению количества фаз и количества узлов конечных элементов);
- `vector<vector<double>>` `x` – массив, содержащий значения массовых долей каждой из компонент (размерность равна количеству компонент);
- `vector<vector<double>>` `amount_of_substance` – массив, содержащий значения количества вещества каждой компоненты (размерность равна количеству компонент).

2.3. МОДУЛЬ РАСЧЁТА НОВОГО СОСТОЯНИЯ ЯЧЕЕК

Общий алгоритм расчёта нового состояния ячеек

Шаг 1. Расчёт перетекающего потока смеси $Q_e, e = \overline{1, N}$ через грани конечных элементов по формуле (1.16).

Шаг 2. Расчёт перетекающих потоков фаз $Q_e^m, e = \overline{1, N}, m = \overline{1, 2}$.

Шаг 3. Расчёт шага по времени Δt .

Шаг 4. Расчёт перетекающих объёмов фаз $V_{e, out}^m, e = \overline{1, N}, m = \overline{1, 2}$.

Шаг 5. Расчёт новых насыщенных фаз в ячейках $S_e^m, e = \overline{1, N}, m = \overline{1, 2}$.

Шаг 6. Расчёт количества вещества компонент в ячейках $n_e^l, e = \overline{1, N}, l = \overline{1, 4}$.

Шаг 7. Расчёт массовых долей компонент в ячейках $\chi_e^l, e = \overline{1, N}, l = \overline{1, 4}$.

Шаг 8. Расчёт новых значений остаточной насыщенности нефти в ячейках $S_{e, residual}^{oil}, e = \overline{1, N}$.

Шаг 9. Расчёт новых значений вязкости фазы воды в ячейках $\eta_e^{water}, e = \overline{1, N}$.

2.4.1. Подмодуль расчёта перетекающий поток фаз

Перетекающие потоки фаз используются для расчёта шага по времени, а далее, с учётом выбранного шага по времени вычисляются перетекающие потоки фаз, поэтому в разработанной программе расчёт перетекающих потоков фаз выполняется в отдельном подмодуле один раз за одну итерацию пересчёта состояния ячеек.

Входные данные. Конечноэлементная сетка, значения перетекающих потоков смеси Q_e .

Выходные данные. Массив $\{Q_e^m\}, e = \overline{1, N}, m = \overline{1, 2}$ перетекающих потоков фаз.

Алгоритм расчёта. В цикле по конечным элементам $\Omega_e, e = \overline{1, N}$ для каждой фазы вычисляются вытекающие из текущего элемента потоки фаз по следующей формуле:

$$Q_e^m = \alpha_e^m Q_e, \quad (2.1)$$

где α_e^m определяется соотношением (1.19).

2.4.2. Подмодуль расчёта шага по времени

Входные данные. Массив $\{Q_e^m\}, e = \overline{1, N}, m = \overline{1, 2}$ перетекающих потоков фаз, массив $\{S_e^m\}, e = \overline{1, N}, m = \overline{1, 2}$ насыщенностей фаз в элементах.

Выходные данные. Шаг по времени Δt .

Алгоритм расчёта. В цикле по конечным элементам $\Omega_e, e = \overline{1, N}$ выполняется следующее:

- a. если текущая насыщенность нефти на элементе $S_e^{oil} > S_{crit}^{oil}$, то вычисляем Δt по соотношению (1.21);
- b. иначе переходим к следующему элементу.

2.4.3. Подмодуль расчёта перетекающих объёмов фаз

Входные данные. Массив $\{Q_e^m\}, e = \overline{1, N}, m = \overline{1, 2}$ перетекающих потоков фаз, шаг по времени Δt .

Выходные данные. Массив $\{V_{e, out}^m\}, e = \overline{1, N}, m = \overline{1, 2}$ перетекающих объёмов фаз.

Алгоритм расчёта. В цикле по конечным элементам $\Omega_e, e = \overline{1, N}$ для каждой фазы вычисляются объёмы вытекающих фаз по формуле:

$$V_{e, out}^m = Q_e^m \cdot \Delta t. \quad (2.2)$$

2.4.4. Подмодуль расчёта новых значений насыщенности фаз

Входные данные. Массив $\{V_{e, out}^m\}, e = \overline{1, N}, m = \overline{1, 2}$ перетекающих объёмов фаз, массив $\{S_e^m\}, e = \overline{1, N}, m = \overline{1, 2}$ насыщенностей фаз в элементах.

Выходные данные. Массив $\{S_e^m\}, e = \overline{1, N}, m = \overline{1, 2}$ новых насыщенностей фаз.

Алгоритм расчёта. В цикле по конечным элементам $\Omega_e, e = \overline{1, N}$ для каждой фазы:

- а. вычисляются текущие объёмы фаз на элементе $V_{e, current}^m$ по формулам (1.22), (1.23);
- б. вычисляются новые насыщенности фаз на элементе S_e^m по формуле (1.24);
- с. если $V_{e, out}^{oil} > mes(\Omega_e)\Phi(S_e^{oil} - S_{e, residual}^{oil})$, то:

$$V_{e, out}^{water} = V_{e, out}^{water} + V_{e, out}^{oil}, \quad (2.3)$$

$$V_{e, out}^{oil} = mes(\Omega_e)\Phi(S_e^{oil} - S_{e, residual}^{oil}), \quad (2.4)$$

$$V_{e, out}^{water} = V_{e, out}^{water} - V_{e, out}^{oil}. \quad (2.5)$$

2.4.5. Подмодуль расчёта количества вещества компонент

Входные данные. Массив $\{V_{e, out}^m\}, e = \overline{1, N}, m = \overline{1, 2}$ перетекающих объёмов фаз, массив $V_{e, current}^m, e = \overline{1, N}, m = \overline{1, 2}$ текущих объёмов фаз на элементах, массив $\{M^l\}, l = \overline{1, 4}$ молярных масс компонент, массив $\rho^m, m = \overline{1, 2}$ плотности фаз, массив $\chi_e^l, e = \overline{1, N}, l = \overline{1, 4}$ массовых долей каждой компоненты.

Выходные данные. Массив $\{n_e^l\}, e = \overline{1, N}, l = \overline{1, 4}$ количества вещества для каждой компоненты.

Алгоритм расчёта. В цикле по конечным элементам $\Omega_e, e = \overline{1, N}$:

- а. для фазы воды по каждой компоненте (вода, полимер, ПАВ) вычисляется количества вещества по формуле (1.26);
- б. для фазы нефти вычисляется количество вещества по формуле:

$$n_e^{oil} = \frac{V_{e, current}^{oil} \cdot \rho^{oil}}{M^{oil}}. \quad (2.6)$$

2.4.6. Подмодуль расчёта массовых долей компонент

Входные данные. Массив $\{n_e^l\}, e = \overline{1, N}, l = \overline{1, 4}$ количества вещества для каждой компоненты, массив $\{M^l\}, l = \overline{1, 4}$ молярных масс компонент.

Выходные данные. Массив $\{\chi_e^l\}, e = \overline{1, N}, l = \overline{1, 4}$ массовых долей компонент смеси.

Алгоритм расчёта. В цикле по конечным элементам $\Omega_e, e = \overline{1, N}$ для каждой компоненты смеси вычисляются значения массовых долей компонент χ_e^l по формуле (1.26).

2.4.7. Подмодуль расчёта значений остаточной насыщенности нефти в ячейках

Входные данные. Массив $\{n_e^l\}, e = \overline{1, N}, l = \overline{1, 4}$ количества вещества компонент смеси, количество моль нефти и ПАВ-вещества ($moles_{required}^{oil}$, $moles_{required}^{surfactant}$) необходимых для осуществления реакции, массив $\{S_e^{residual}\}, e = \overline{1, N}$ остаточных насыщенностей нефти в ячейках.

Выходные данные. Массив $\{S_e^{residual}\}, e = \overline{1, N}$ новых остаточных насыщенностей нефти в ячейках.

Алгоритм расчёта. В цикле по конечным элементам $\Omega_e, e = \overline{1, N}$:

- если массовая доля ПАВ-вещества на элементе больше критического значения (минимальная величина доли ПАВ, при которой может произойти реакция), то есть $\chi_e^{surfactant} > \chi_{crit}^{surfactant}$, то:
- вычисляется количество частей нефти, которое может вступить в реакцию по формуле:

$$portion^{oil} = \left[\frac{n_e^{oil}}{moles^{oil}} \right] \quad (2.7)$$

и соответствующее количество моль ПАВ-вещества, необходимых для реакции:

$$moles^{\text{surfactant}} = moles_{\text{required}}^{\text{surfactant}} \cdot portion^{\text{oil}}; \quad (2.8)$$

с. если $moles^{\text{surfactant}} > n_e^{\text{surfactant}}$, что значит, что на предлагаемое количество нефти не хватает ПАВ-вещества, то вычисляется количество частей ПАВ, которое может вступить в реакцию по формуле:

$$portion^{\text{surfactant}} = \left[\frac{n_e^{\text{surfactant}}}{moles^{\text{surfactant}}} \right] \quad (2.9)$$

и соответствующее количество моль нефти, необходимых для реакции:

$$moles^{\text{oil}} = moles_{\text{required}}^{\text{oil}} \cdot portion^{\text{surfactant}}; \quad (2.10)$$

иначе $moles^{\text{oil}} = moles_{\text{required}}^{\text{oil}} \cdot portion^{\text{oil}};$

d. далее можем посчитать объём нефти, вступившей в реакцию (иными словами, объём нефти, который теперь станет подвижным):

$$V_e^{\text{oil}} = \frac{moles^{\text{oil}} \cdot M^{\text{oil}}}{\rho^{\text{oil}}}; \quad (2.11)$$

e. исходя из вычисленной величины можно найти новое значение остаточной насыщенности нефти на элементе:

$$S_e^{\text{residual}} = S_e^{\text{residual}} - \frac{V_e^{\text{oil}}}{mes(\Omega_e)\Phi}. \quad (2.12)$$

2.4.8. Подмодуль расчёта новых значений вязкости фазы воды в ячейках

Входные данные. Массив $\{\chi_e^l\}, e = \overline{1, N}, l = \overline{1, 4}$ массовых долей компонент смеси, данные о вязкости воды в зависимости от концентрации полимера.

Выходные данные. Массив $\{\eta_e^{water}\}, e = \overline{1, N}$ новых значений вязкости фазы воды в ячейках.

Алгоритм расчёта. В цикле по конечным элементам $\Omega_e, e = \overline{1, N}$:

- a. для фазы воды в таблице с данными о вязкости воды в зависимости от концентрации находим ближайшие справа и слева значения массовой доли $(\chi_{i+1}^{polymer}, \chi_i^{polymer})$ по отношению к текущему значению массовой доли на элементе $\chi_e^{polymer}$;
- b. с помощью линейной интерполяции вычисляются новые значения вязкости:

$$\eta_e^{water} = \eta^{water}(\chi_i^{polymer}) + \frac{\eta^{water}(\chi_{i+1}^{polymer}) - \eta^{water}(\chi_i^{polymer})}{\chi_{i+1}^{polymer} - \chi_i^{polymer}} (\chi_e^{polymer} - \chi_i^{polymer}). \quad (2.13)$$

3. ТЕСТИРОВАНИЕ МОДУЛЯ РАСЧЁТА ДАВЛЕНИЯ

Для того, чтобы убедиться в правильности работы модуля расчёта поля давления, проверим, что для используемых линейных базисных функций порядок аппроксимации соответствует теоретически ожидаемому. Ожидаемый порядок аппроксимации – первый.

Зададим функцию $P(x)$, расчётную область разобьём на 10 конечных элементов при $x \in [0;5]$. На обеих границах зададим первые краевые условия $P = P^*$. Погрешность будем вычислять следующим образом:

$$\delta = \frac{\sum_{i=0}^N |P^*(x_i) - P(x_i)|}{N}, \quad (3.1)$$

где N – количество узлов в сетке (в данном случае $N = 11$), i – номер узла сетки, $P^*(x_i)$ – аналитическое значение функции в точке x_i , а $P(x_i)$ – численное значение функции в этой же точке.

В таблице 3.1 приведены результаты работы модуля расчёта давления, необходимые для определения порядка аппроксимации. На полиномах нулевой и первой степени погрешность условно равна нулю ($\delta \neq 0.0$ из-за погрешности вычислений), а на полиноме второй степени уже можем видеть значительную погрешность. Следовательно, теоретически ожидаемый порядок аппроксимации подтверждён.

Таблица 3.1 – Поиск порядка аппроксимации

| P^* , аналитическая функция | δ , погрешность полученного решения |
|-------------------------------|--|
| 2 | 2,83E-16 |
| x | 4,34E-16 |
| x*x | 3,75E+00 |

4. ИССЛЕДОВАНИЯ

4.1. ПОИСК ОПТИМАЛЬНЫХ ПАРАМЕТРОВ

В данном разделе производились тестирование модуля расчёта нового состояния ячеек с целью выбора оптимальных значений таких параметров, как шаг по времени и шаг разбиения сетки.

Общие входные данные для последующих тестов: длина керна $l = 1$ м, площадь сечения керна $s_{сеч} = 0.00114$ м², структурная пористость $\Phi = 0.215$, структурная проницаемость $K = 0.0349$ мкм², $k_{\max}^{oil} = 0.425$, $k_{\max}^{water} = 0.05$, $\beta^{oil} = 2.0$, $\beta^{water} = 1.5$, критическое значение насыщенности для нефти $S_{crit}^{oil} = 0.31$, остаточные значения насыщенности одинаковы и равны $S_{residual}^{oil} = 0.3$, начальное распределение насыщенностей равномерно по всей длине и равно $S_e^{oil} = 0.7$, $S_e^{water} = 0.3$, вязкость нефти $\eta^{oil} = 0.005$, поток на левой границе $\theta = 3.0e - 5$ м³/с, плотность фазы воды $\rho^{water} = 1170$, нефти – $\rho^{oil} = 873$, молярные массы компонент: воды – $M^{water} = 18.0e - 3$, полимера – $M^{polymer} = 0.2$, ПАВ – $M^{surfactant} = 435.0e - 3$, нефти – $M^{oil} = 250.0e - 3$. В керн не вкачивали полимер и ПАВ, то есть $\chi_e^{polymer} = 0$ и $\chi_e^{surfactant} = 0$.

Для поиска оптимального шага разбиения сетки выберем произвольный шаг по времени $\Delta t = 10$ с. Количество конечных элементов, на которые разбита расчётная область, с каждым разом будем увеличивать в 2 раза, начиная с пяти конечных элементов.

Нефть вытеснялась в течение 10 часов. Общий объём подвижной нефти можем найти следующим образом:

$$\begin{aligned} V_{mobile}^{oil} &= l \cdot s_{сеч} \cdot \Phi \cdot (S_{e,t=0}^{oil} - S_{residual}^{oil}) = \\ &= 1 \cdot 0.00114 \cdot 0.215 \cdot (0.7 - 0.3) = 9.804e - 5 \text{ м}^3. \end{aligned} \quad (4.1)$$

Рисунок 4.1 показывает графики суммарного объёма вытесненной нефти для расчётной области, разделённой на 5, 10 и 20 конечных элементов. Для модели, где расчётная область разделена на 5 конечных элементов, суммарный объём вытекшей нефти равен $V_{total}^{oil} = 9.58786e-05 \text{ м}^3$, на 10 конечных элементов – $V_{total}^{oil} = 9.62557e-05 \text{ м}^3$, на 20 конечных элементов – $V_{total}^{oil} = 9.64699e-05 \text{ м}^3$.

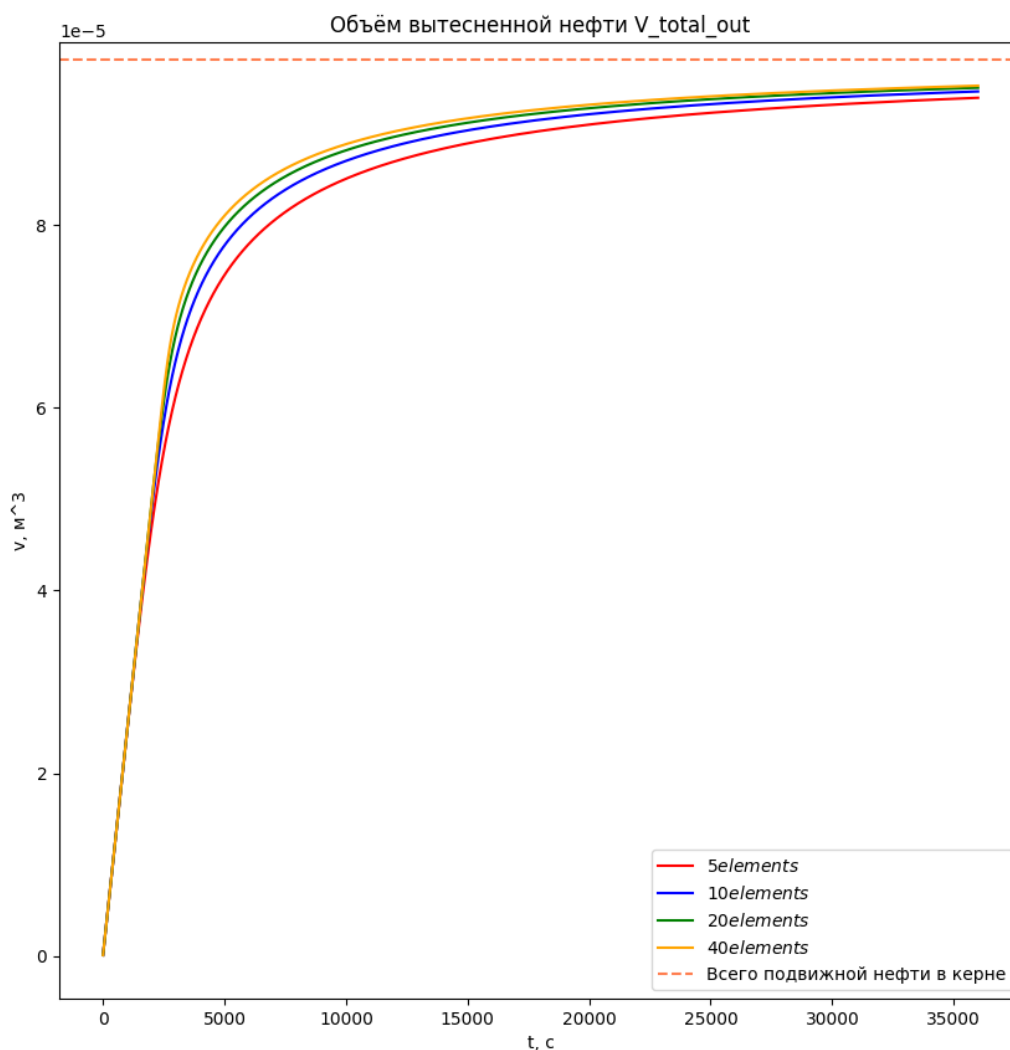


Рисунок 4.1 – Объём вытесненной нефти в зависимости от времени с дроблением сетки.

Из рисунка 4.2 видно, что разница между моделью, разделённой на 20 конечных элементов и моделью, разделённой на 40 конечных элементов, меньше 3%. Таким образом для дальнейших расчётов можно использовать модель, разделённую на 20 конечных элементов.

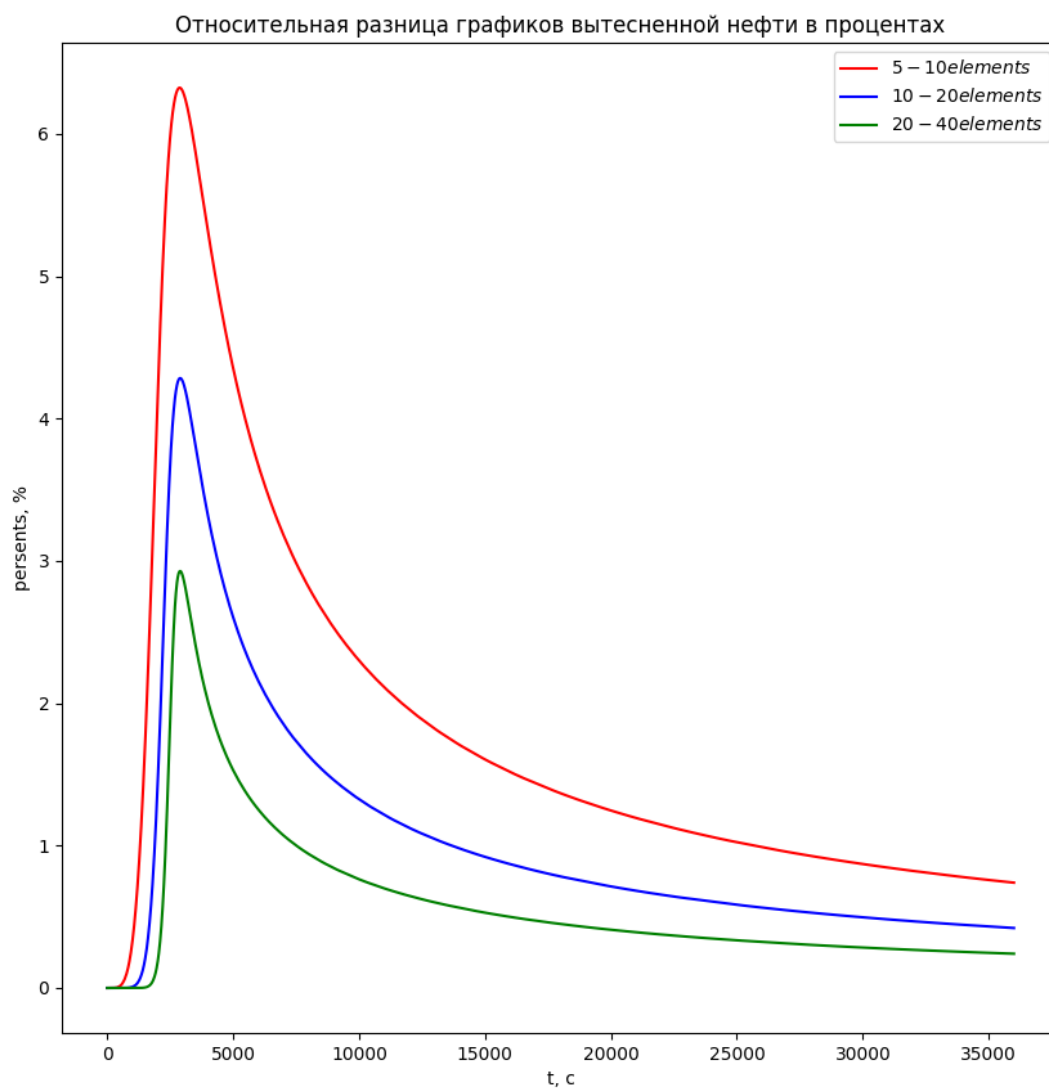


Рисунок 4.2 – Разница между графиками в процентном соотношении.

Для поиска оптимального шага по времени, учитывая результаты предыдущего исследования, разделим расчётную область на 20 конечных элементов. Шаг по времени Δt с каждым разом будем уменьшать в 2 раза начиная с 80 секунд.

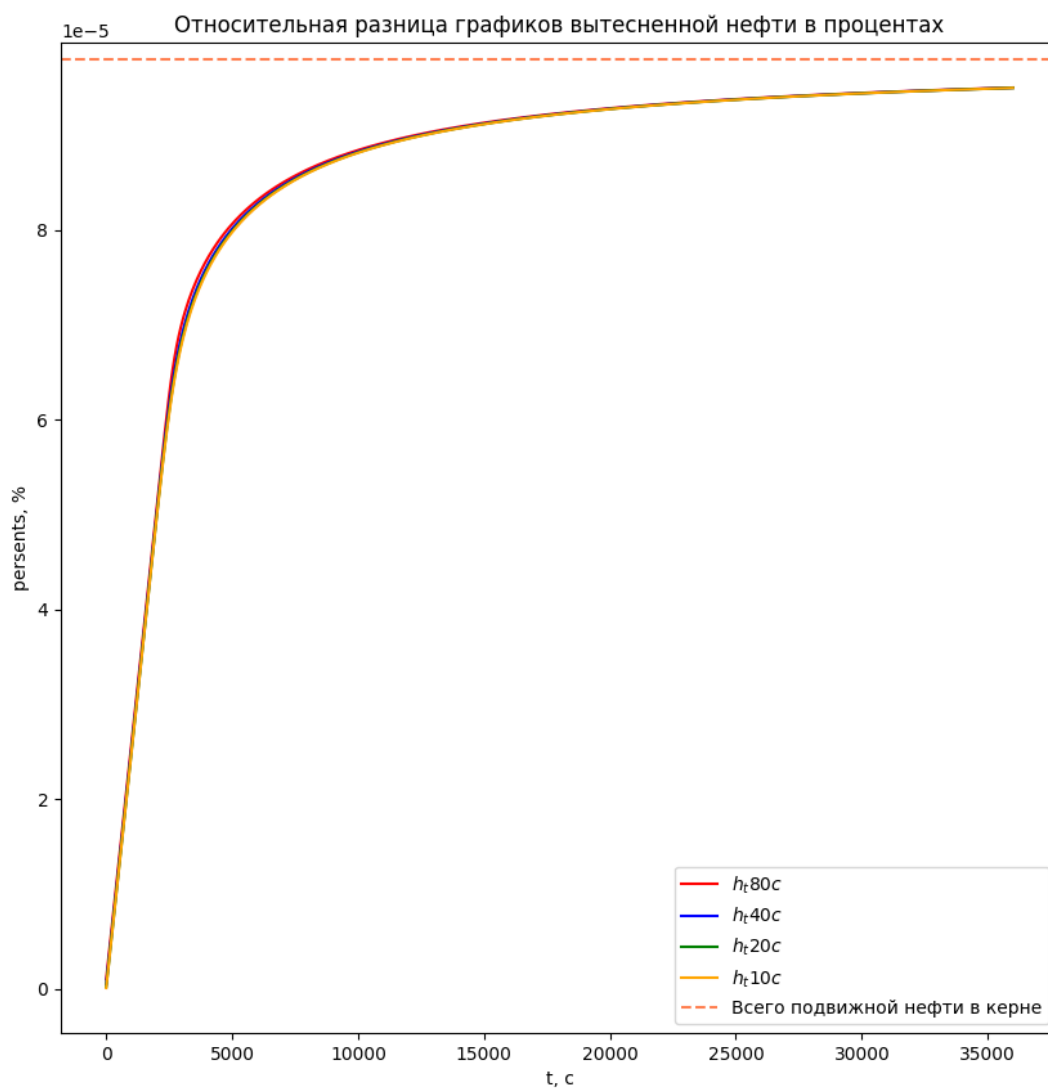


Рисунок 4.3 – График вытесненной нефти в зависимости от времени с увеличением шага по времени.

Из рисунка 4.4 видно, что разница между решением с шагом по времени, равным 20 секунд и решением с шагом по времени, равным 10 секунд разница меньше 3%, поэтому в дальнейших тестах и исследованиях будем использовать $\Delta t = 20$ с.

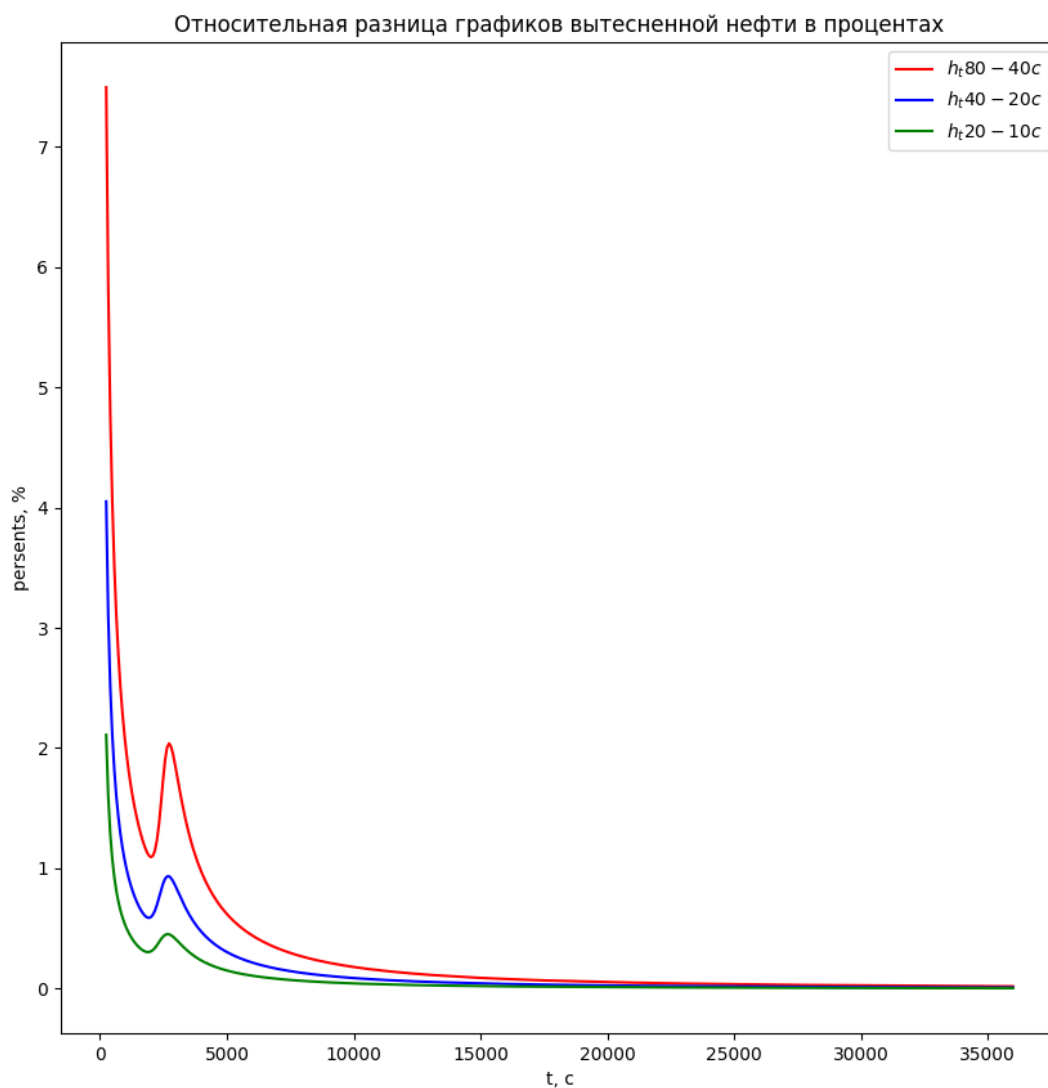


Рисунок 4.4 – Разница между графиками в процентном соотношении

4.2. ИССЛЕДОВАНИЕ МЕТОДОВ ПОВЫШЕНИЯ НЕФТЕОТДАЧИ ПЛАСТОВ

В данном разделе приведены результаты трёх численных экспериментов. В первом проводилось моделирование многофазного потока в пористой среде в технологиях нефтедобычи, использующих полимерное заводнение, во втором проводилось моделирование многофазного потока, используя технологию увеличения количества подвижной нефти в керне при контакте ПАВ с неподвижной нефтью, а в третьем была исследована эффективность применения сразу двух методов одновременно.

Общие входные данные для последующих исследований: длина керна $l = 1$ м, площадь сечения керна $S_{сеч} = 0.00114$ м², структурная пористость $\Phi = 0.215$, структурная проницаемость $K = 0.0349$ мкм², $k_{max}^{oil} = 0.425$, $k_{max}^{water} = 0.05$, $\beta^{oil} = 2.0$, $\beta^{water} = 1.5$, критическое значение насыщенности для нефти $S_{crit}^{oil} = 0.31$, остаточные значения насыщенности одинаковы и равны $S_{residual}^{oil} = 0.3$, начальное распределение насыщенностей равномерно по всей длине и равно $S_e^{oil} = 0.7$, $S_e^{water} = 0.3$, вязкость нефти $\eta^{oil} = 0.005$, поток на левой границе $\theta = 3.0e - 5$ м³/с, плотность фазы воды $\rho^{water} = 1170$, нефти – $\rho^{oil} = 873$, молярные массы компонент: воды – $M^{water} = 18.0e - 3$, полимера – $M^{polymer} = 0.2$, ПАВ – $M^{surfactant} = 435.0e - 3$, нефти – $M^{oil} = 250.0e - 3$, $\Delta t = 20$ с, расчётная область разбита на 20 конечных элементов с шагом 0.05 м.

4.2.1 Исследование зависимости объёма вытесненной нефти от концентрации полимера в фазе воды.

В данном исследовании из керна вытеснялась вся подвижная нефть в течении 36000 секунд.

Рисунок 4.5 показывает накопительный объём вытекшей нефти для разных концентраций полимера в воде при закачке. На рисунках 4.6 – 4.7 изображены распределения насыщенностей нефти на конечных элементах через каждые 200 секунд начиная с момента времени $t = 0$ с до $t = 4200$ с.

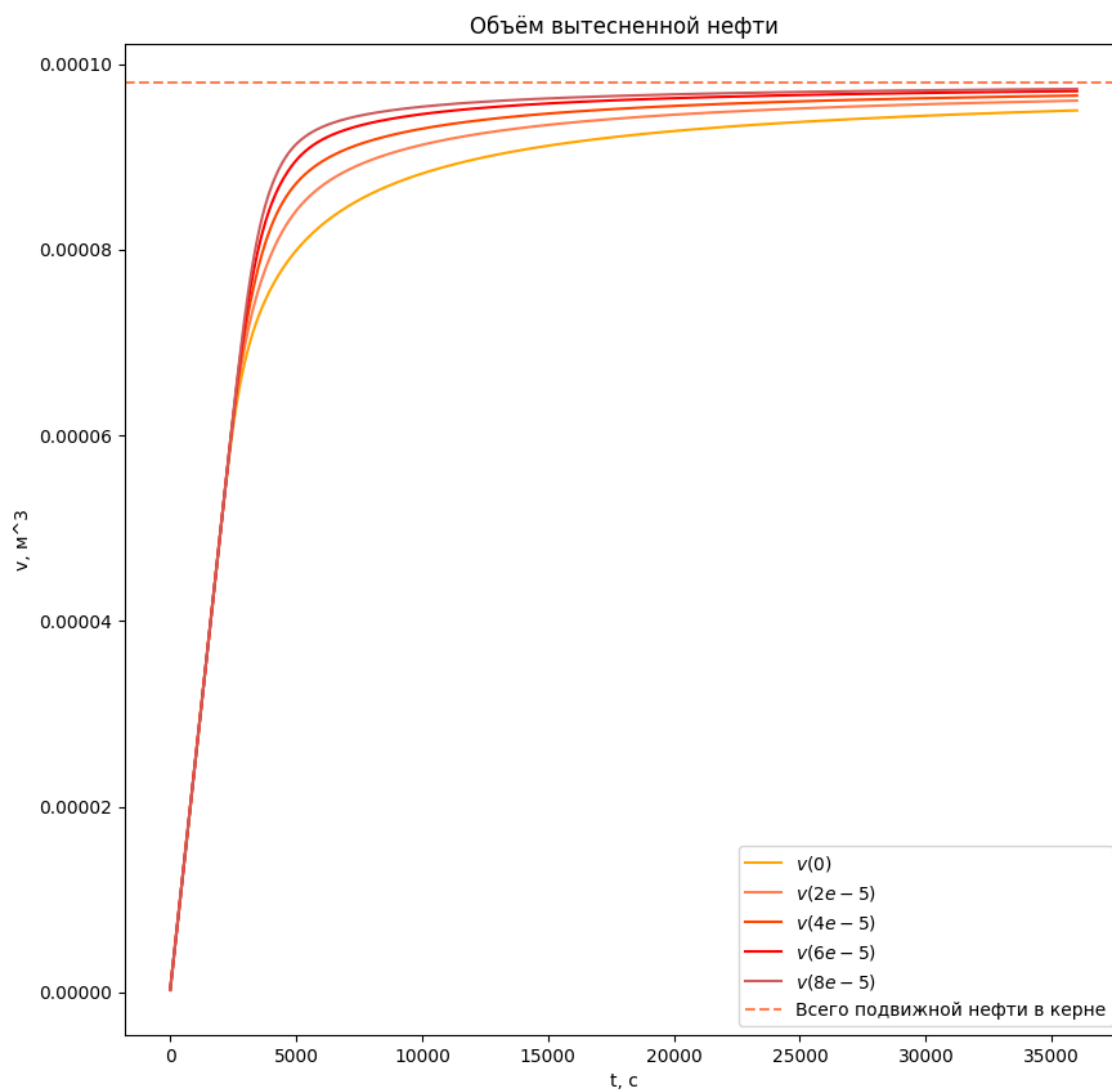


Рисунок 4.5 – Объём вытесненной нефти с увеличением концентрации полимера.

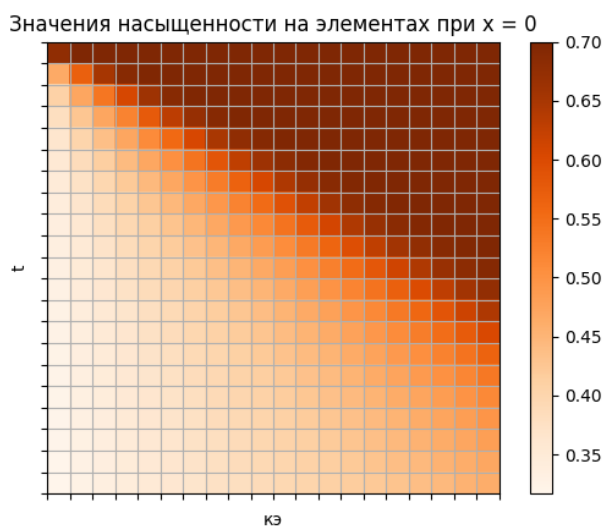


Рисунок 4.6 – Распределение насыщенности нефти без закачки полимера.

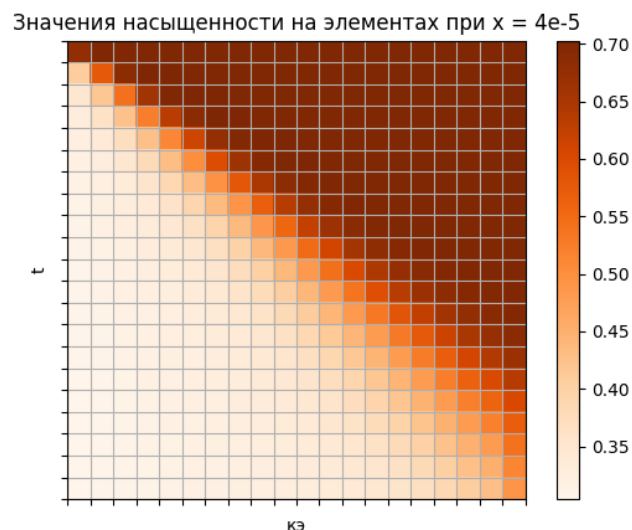


Рисунок 4.7 – Распределение насыщенности нефти при концентрации полимера в составе закачиваемой смеси $4e-5$.

Исходя из рисунка 4.8 объём дополнительно добытой нефти существенно увеличивается при закачке раствора полимера от 0 до $1.5e-7 \text{ м}^3$. С дальнейшим увеличением объёма закачиваемого раствора полимера (до $5.0e-7 \text{ м}^3$) объём дополнительно добытой нефти также увеличивается, однако её прирост на единицу объёма закаченного полимера довольно резко снижается. Сопоставив данные результаты с таблицей 4.1 можно сделать вывод, что для данной модели значение оптимальной концентрации полимера при закачке лежит в диапазоне от $1.0e-4$ до $1.5e-4$.

Таблица 4.1 – Данные о добытой нефти в зависимости от количества закаченного полимера.

| Массовая доля полимера | Масса закаченного полимера, кг | Объем закаченного полимера = V_1 , м ³ | Объем вытесненной нефти, м ³ | Объем дополнительно добытой нефти = V_2 , м ³ |
|------------------------|--------------------------------|---|---|--|
| 0,00E+00 | 0,00E+00 | 0,00E+00 | 9,4994200E-05 | |
| 2,00E-05 | 2,8810080E-05 | 2,4624000E-08 | 9,6041500E-05 | 1,0473000E-06 |
| 4,00E-05 | 5,7620160E-05 | 4,9248000E-08 | 9,6588900E-05 | 1,5947000E-06 |
| 6,00E-05 | 8,6430240E-05 | 7,3872000E-08 | 9,7069600E-05 | 2,0754000E-06 |
| 8,00E-05 | 1,1524032E-04 | 9,8496000E-08 | 9,7302600E-05 | 2,3084000E-06 |
| 1,00E-04 | 1,4405040E-04 | 1,2312000E-07 | 9,7447700E-05 | 2,4535000E-06 |
| 1,50E-04 | 2,1607560E-04 | 1,8468000E-07 | 9,7634500E-05 | 2,6403000E-06 |
| 2,00E-04 | 2,8810080E-04 | 2,4624000E-07 | 9,7729300E-05 | 2,7351000E-06 |
| 2,50E-04 | 3,6012600E-04 | 3,0780000E-07 | 9,7785400E-05 | 2,7912000E-06 |
| 3,00E-04 | 4,3215120E-04 | 3,6936000E-07 | 9,7822200E-05 | 2,8280000E-06 |
| 3,50E-04 | 5,0417640E-04 | 4,3092000E-07 | 9,7849100E-05 | 2,8549000E-06 |
| 4,00E-04 | 5,7620160E-04 | 4,9248000E-07 | 9,7869600E-05 | 2,8754000E-06 |

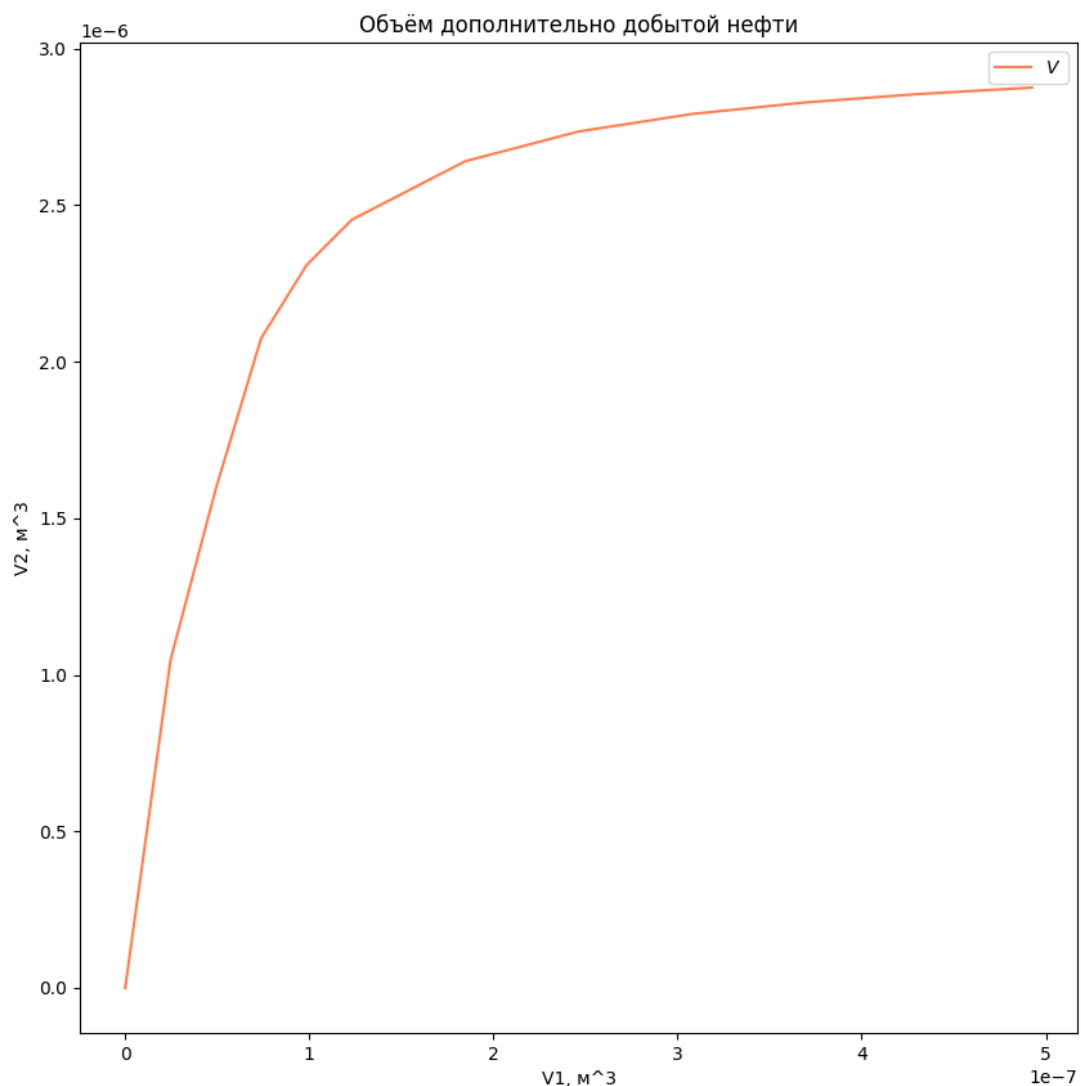


Рисунок 4.8 – График зависимости дополнительно добытой нефти от объема закаченного полимера.

4.2.2 Исследование зависимости объема вытесненной нефти от концентрации ПАВ в воде.

Рисунок 4.9 показывает накопительный объем вытесненной нефти при разных концентрациях ПАВ в фазе воды при закачке. На рисунках 4.10 – 4.13 изображены распределения насыщенности остаточной нефти на конечных элементах через каждые 1800 секунд при разных концентрациях ПАВ в составе закачиваемой смеси, начиная с момента времени $t = 0$ с до $t = 37800$ с.

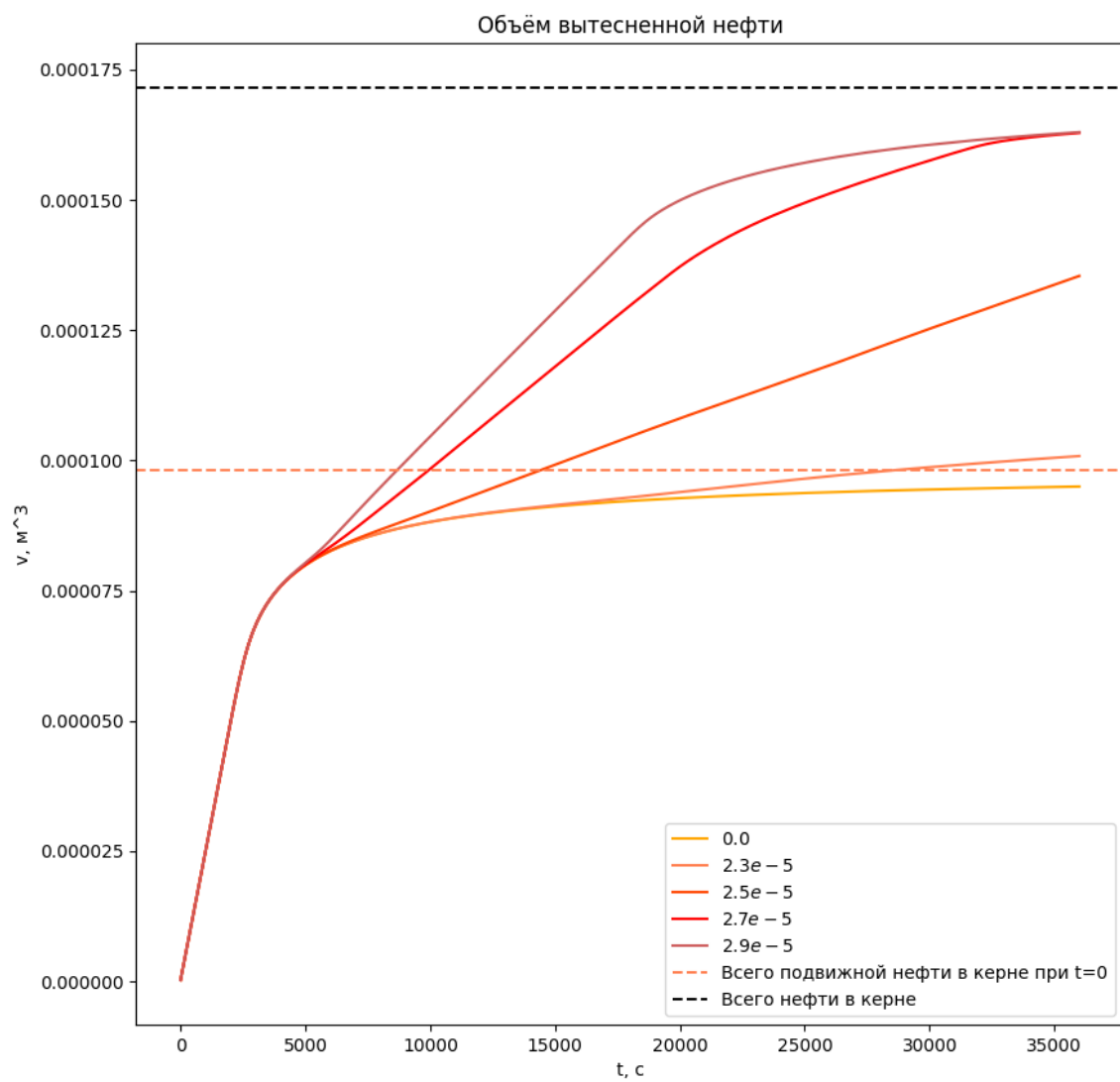


Рисунок 4.9 – Объём вытесненной нефти с увеличением концентрации ПАВ в составе закачиваемой смеси.

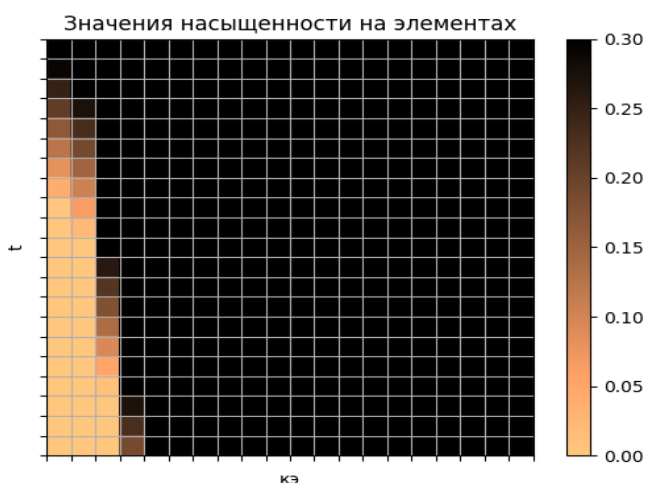


Рисунок 4.10 – Распределение насыщенности остаточной нефти при концентрации ПАВ в составе закачиваемой смеси 2.3×10^{-5} .

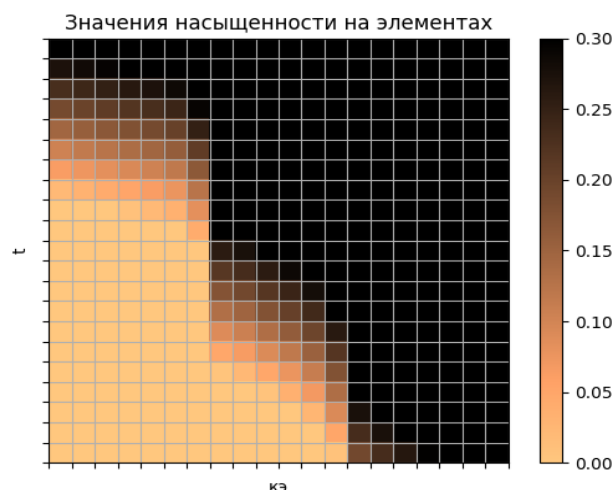


Рисунок 4.11 – Распределение насыщенности остаточной нефти при концентрации ПАВ в составе закачиваемой смеси 2.5×10^{-5} .

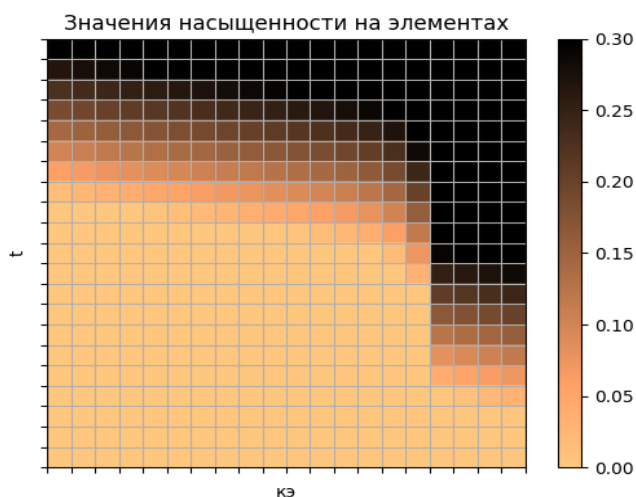


Рисунок 4.12 – Распределение насыщенности остаточной нефти при концентрации ПАВ в составе закачиваемой смеси 2.7×10^{-5} .

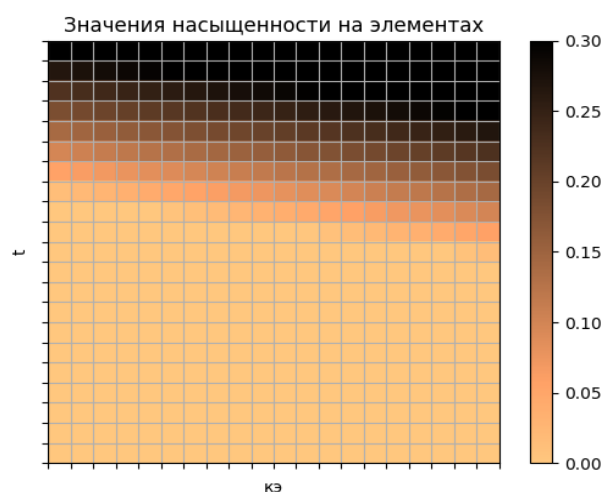


Рисунок 4.13 – Распределение насыщенности остаточной нефти при концентрации ПАВ в составе закачиваемой смеси 2.9×10^{-5} .

Исходя из рисунка 4.14 и таблицы 4.2 можем отметить скачок объёма дополнительно добытой нефти от $5.8628000 \times 10^{-6} \text{ м}^3$ до $4.0436800 \times 10^{-5} \text{ м}^3$ при увеличении объёма закачиваемого ПАВ от $2.8317600 \times 10^{-8} \text{ м}^3$ до $3.0780000 \times 10^{-8} \text{ м}^3$. Из этих данных следует, что для данной модели значение оптимальной концентрации ПАВ при закачке равно 2.5×10^{-5} .

Таблица 4.2 – Данные о количестве добытой нефти в зависимости от количества закаченного ПАВ.

| Массовая доля ПАВ | Масса закаченного ПАВ, кг | Объем закаченного ПАВ, м ³ | Объем вытесненной нефти, м ³ | Объем дополнительно добытой нефти, м ³ |
|-------------------|---------------------------|---------------------------------------|---|---|
| 0,00E+00 | 0,0000000E+00 | 0,0000000E+00 | 9,4994200E-05 | |
| 2,30E-05 | 3,3131592E-05 | 2,8317600E-08 | 1,0085700E-04 | 5,8628000E-06 |
| 2,35E-05 | 3,3851844E-05 | 2,8933200E-08 | 1,0391000E-04 | 8,9158000E-06 |
| 2,40E-05 | 3,4572096E-05 | 2,9548800E-08 | 1,1366500E-04 | 1,8670800E-05 |
| 2,45E-05 | 3,5292348E-05 | 3,0164400E-08 | 1,2320800E-04 | 2,8213800E-05 |
| 2,50E-05 | 3,6012600E-05 | 3,0780000E-08 | 1,3543100E-04 | 4,0436800E-05 |
| 2,70E-05 | 3,8893608E-05 | 3,3242400E-08 | 1,6289700E-04 | 6,7902800E-05 |
| 2,90E-05 | 4,1774616E-05 | 3,5704800E-08 | 1,6303700E-04 | 6,8042800E-05 |
| 3,20E-05 | 4,6096128E-05 | 3,9398400E-08 | 1,6307200E-04 | 6,8077800E-05 |
| 3,40E-05 | 4,8977136E-05 | 4,1860800E-08 | 1,6317500E-04 | 6,8180800E-05 |
| 3,60E-05 | 5,1858144E-05 | 4,4323200E-08 | 1,6342800E-04 | 6,8433800E-05 |
| 3,80E-05 | 5,4739152E-05 | 4,6785600E-08 | 1,6370300E-04 | 6,8708800E-05 |
| 4,00E-05 | 5,7620160E-05 | 4,9248000E-08 | 1,6394800E-04 | 6,8953800E-05 |
| 4,50E-05 | 6,4822680E-05 | 5,5404000E-08 | 1,6437500E-04 | 6,9380800E-05 |

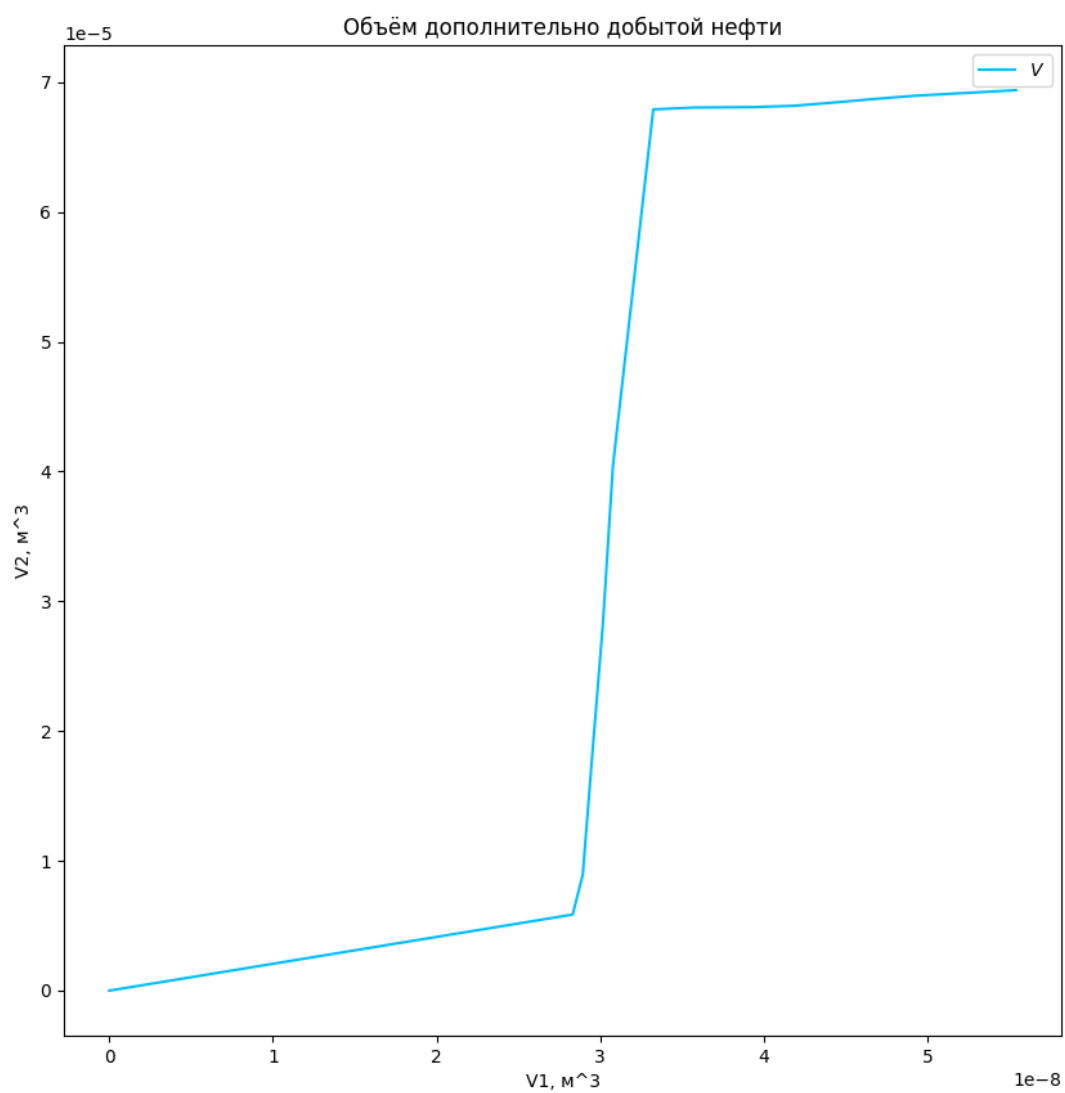


Рисунок 4.14 – График зависимости дополнительно добытой нефти от объема закаченного ПАВ.

4.2.3 Исследование эффективности закачки комбинации растворов ПАВ и полимера.

Рисунок 4.15 показывает накопительный объём вытесненной нефти в случае, когда при закачке не использовались ПАВ или полимер (график 0), в случае использования при закачке только ПАВ (график 1), только полимера (график 2) и комбинации ПАВ и полимера (график 3). В таблице 4.3 приведены данные об общем количестве вытесненной нефти и о количестве дополнительно вытесненной нефти для каждого из трёх графиков. Из рисунка 4.15 и таблицы 4.3 можем заметить, что для данной модели при одновременной закачке растворов ПАВ и полимера общее количество вытесненной нефти существенно увеличивается, а именно в 1.25 раз по сравнению со случаем использования только ПАВ и в 1.74 раз по сравнению со случаем использования только полимера); объём дополнительно добытой нефти также увеличивается в 1.85 раз и 3.1 раз для каждого из случаев соответственно.

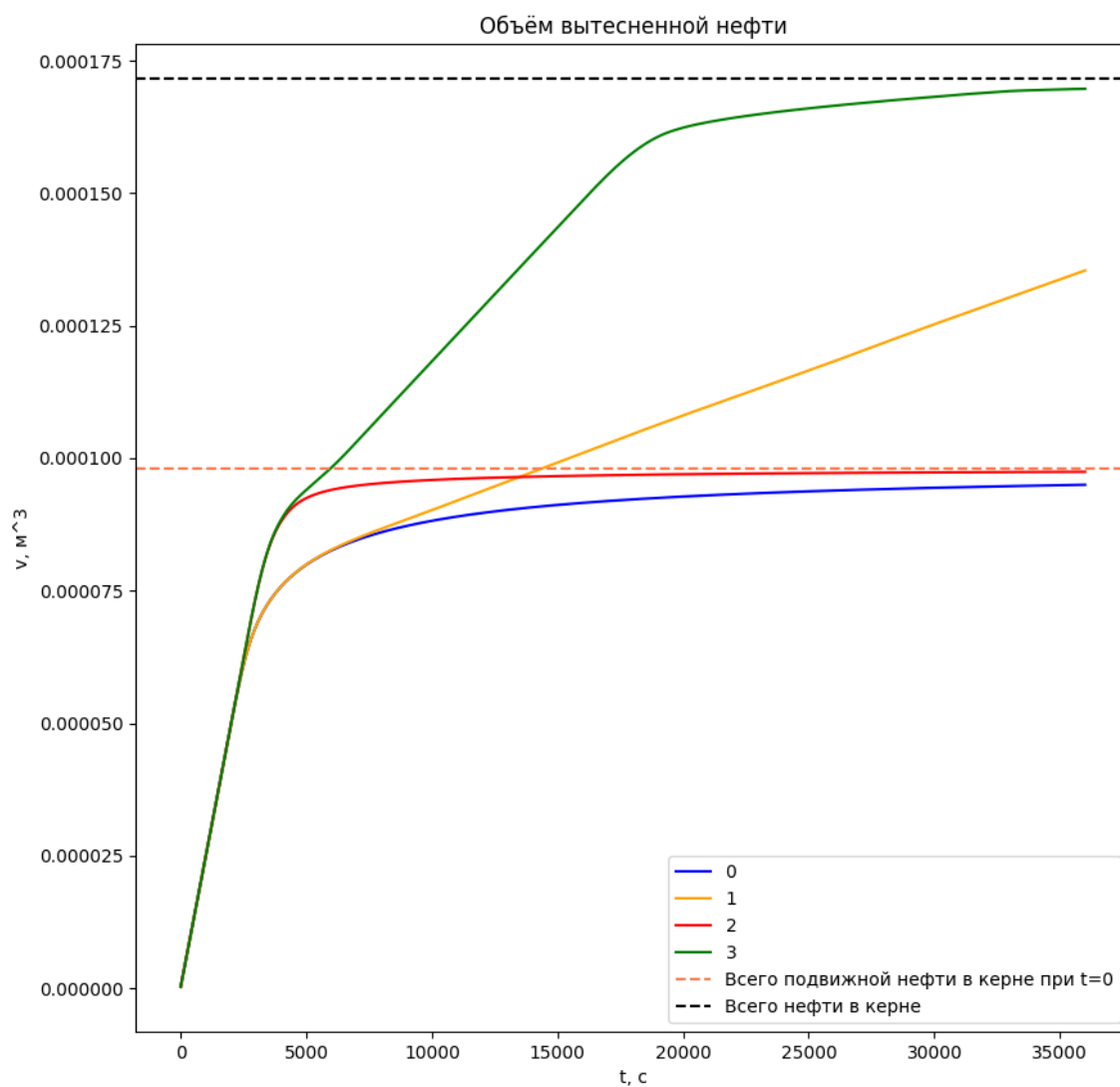


Рисунок 4.15 – Объем вытесненной нефти.

Таблица 4.3 – Сравнительная таблица с данными о количестве вытесненной нефти.

| | Массовая доля поли- мера | Массо- вая доля ПАВ | Масса зака- ченного поли- мера, кг | Масса зака- ченного ПАВ, кг | Объем вытес- ненной нефти, м³ | Объем дополни- тельно добытой нефти, м³ |
|---|--------------------------------|---------------------------|--|-----------------------------------|-------------------------------------|---|
| 0 | 0,00E+00 | 0,00E+00 | | | 9,4994200E-05 | |
| 1 | 0,00E+00 | 2,50E-05 | 0,0000000E+00 | 3,6012600E-05 | 1,3543100E-04 | 4,0436800E-05 |
| 2 | 1,00E-04 | 0,00E+00 | 1,4405040E-04 | 0,0000000E+00 | 9,7447700E-05 | 2,4535000E-06 |
| 3 | 1,00E-04 | 2,50E-05 | 1,4405040E-04 | 3,6012600E-05 | 1,6971000E-04 | 7,4715800E-05 |

4.2.4 Исследование эффективности разных режимов закачки полимера.

Рассмотрим три режима закачки полимера. При первом режиме массовая доля закачиваемого вещества была постоянна и равнялась $1,12E-04$, то есть $\chi^{polymer} = const = 1,12E-04$. При втором режиме начальная массовая доля полимера в момент времени $t=0$ была задана $9,90E-06$ и с каждым часом увеличивалась в 1,5 раза. При третьем режиме начальная массовая доля полимера была задана $3,80E-04$ и с каждым часом уменьшалась в 1,5 раза. Перечисленные параметры были подобраны таким образом, чтобы объёмы закаченного полимера для всех режимов за всё время было примерно одинаковыми. В данном исследовании ПАВ при закачке не использовался.

На рисунке 4.16 для каждого из трёх режимов закачки изображён график объёма вытесненной нефти в зависимости от времени. Можно заметить, что первые 2 часа второй режим значительно уступает первому и третьему по объёму добытой нефти, а далее прирост добытой нефти для каждого из режимов уменьшается и разница между графиками тоже уменьшается.

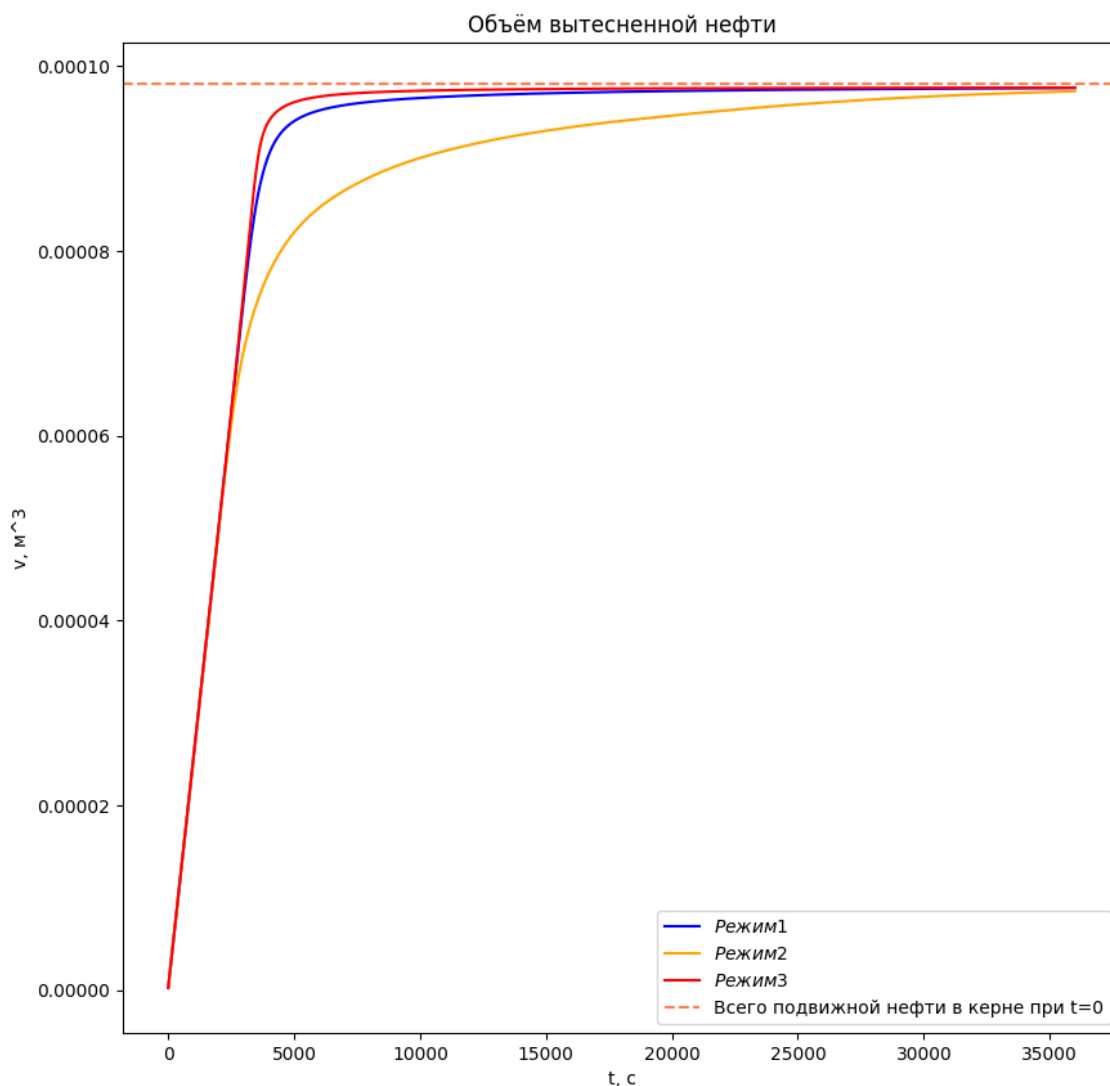


Рисунок 4.16 – Объём вытесненной нефти в соответствии с выбранным режимом закачки полимера.

Исходя из данных таблицы 4.4 по объёму дополнительно добытой нефти все способы закачки дают примерно одинаковый результат. То есть, можно сделать вывод, что для более быстрой добычи нефти эффективнее всего закачивать полимер либо с убыванием концентрации, либо равномерно (с одинаковой концентрацией), но в долговременной перспективе результаты применения каждого из режимов несущественно отличаются друг от друга.

Таблица 4.4 – Данные о количестве добытой нефти.

| № Режима | Концентрация полимера при t=0 | Масса закачен- ного полимера, кг | Объём зака- ченного поли- мера, м ³ | Объём вытес- ненной нефти, м ³ | Объём дополни- тельно добытой нефти, м ³ |
|-------------|-------------------------------------|--|--|---|---|
| 1 | 1,12E-04 | 1,6133645E-04 | 1,378944E-07 | 9,7447700E-05 | 2,4535000E-06 |
| 2 | 9,90E-06 | 1,6161991E-04 | 1,381367E-07 | 9,7296500E-05 | 2,3023000E-06 |
| 3 | 3,80E-04 | 1,6136967E-04 | 1,379228E-07 | 9,7680400E-05 | 2,6862000E-06 |

В следующем исследовании были рассмотрены те же режимы закачки полимера с тем различием, что теперь помимо полимера в керн равномерно закачивали ПАВ с массовой долей 2,5E-5.

На рисунке 4.17 можем наблюдать результаты работы программы для всех трёх способов закачки полимера. Следует отметить, что с 1-го и до 7-го часа считая от начала закачки первый и третий режимы значительно опережают второй режим по скорости добычи нефти.

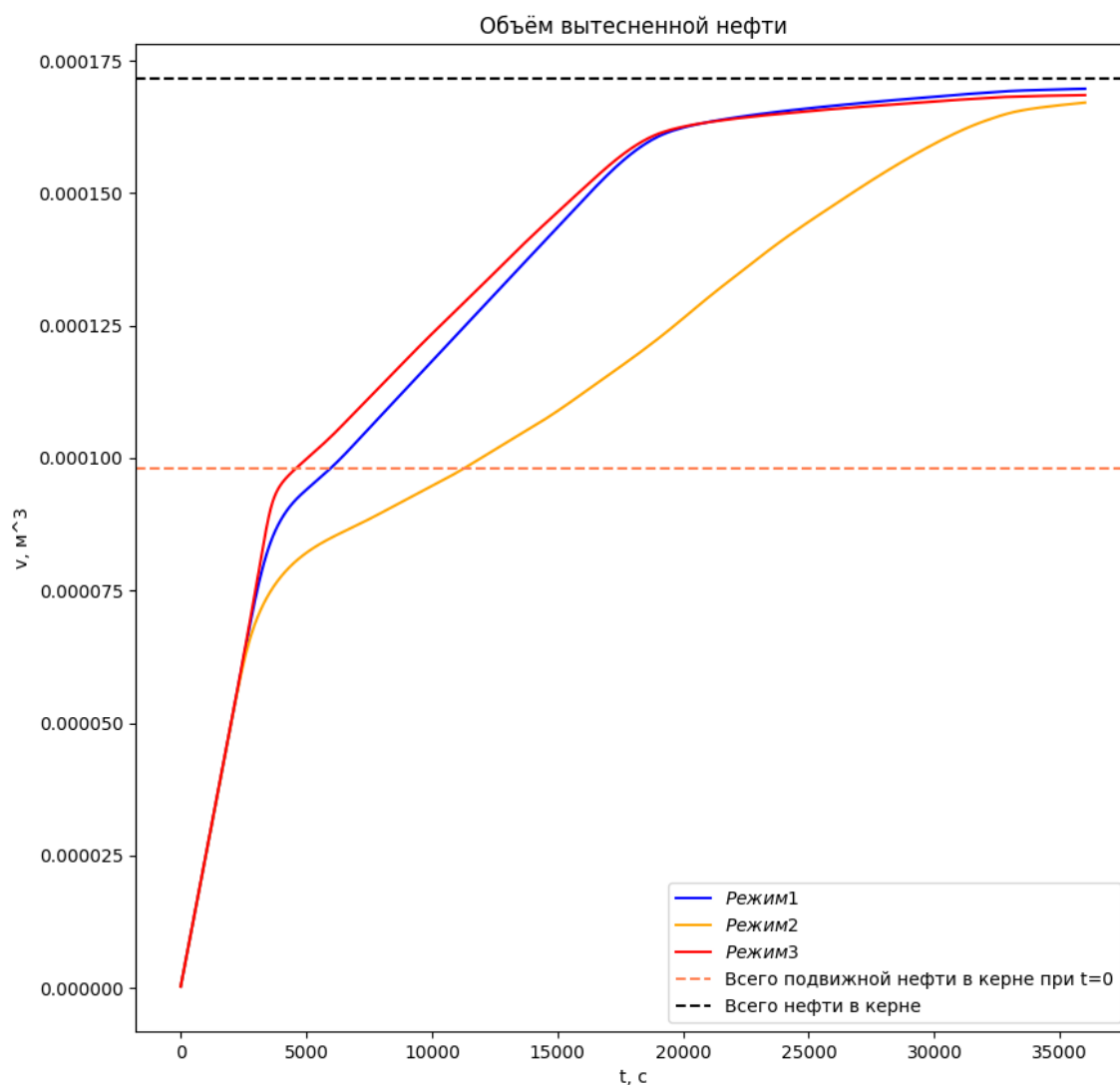


Рисунок 4.17 – Объём вытесненной нефти в соответствии с выбранным режимом закачки полимера.

Исходя из данных таблицы 4.5 по объёму дополнительно добытой нефти все способы закачки дают примерно одинаковый результат. Можно сделать вывод, что для более быстрой добычи нефти эффективнее всего закачивать полимер либо с убыванием концентрации, либо равномерно (с одинаковой концентрацией). Использование ПАВ при закачке увеличивает разрыв в скорости добычи нефти, но в долговременной перспективе результаты применения каждого из режимов несущественно отличаются друг от друга.

Таблица 4.5 – Данные о количестве добытой нефти.

| № Режима | Концентрация полимера при $t=0$ | Масса закачен- ного полимера, кг | Объём зака- ченного поли- мера, м ³ | Объём вытес- ненной нефти, м ³ | Объём дополни- тельно добытой нефти, м ³ |
|-------------|---------------------------------------|--|--|---|---|
| 1 | 1,12E-04 | 1,6133645E-04 | 1,378944E-07 | 1,6971000E-04 | 7,4715800E-05 |
| 2 | 9,90E-06 | 1,6161991E-04 | 1,381367E-07 | 1,6710300E-04 | 7,2108800E-05 |
| 3 | 3,80E-04 | 1,6136967E-04 | 1,379228E-07 | 1,6851900E-04 | 7,3524800E-05 |

4.2.5 Исследование эффективности разных режимов закачки ПАВ.

В данных исследованиях рассмотрены три режима закачки ПАВ. При первом режиме массовая доля закачиваемого вещества была постоянна и равнялась $2,5E-05$, то есть $\chi^{\text{sufractant}} = \text{const} = 2,5E-05$. При втором режиме начальная массовая доля полимера в момент времени $t=0$ была задана $1,00E-05$ и с каждым часом увеличивалась в 1,2 раза. При третьем режиме начальная массовая доля полимера была задана $5,00E-05$ и с каждым часом уменьшалась в 1,2 раза. Перечисленные параметры были подобраны таким образом, чтобы объёмы закаченного ПАВ для всех режимов за всё время было примерно одинаковыми. В данном исследовании полимер при закачке не использовался.

На рисунке 4.18 для каждого из трёх режимов закачки изображён график объёма вытесненной нефти в зависимости от времени. Можем заметить, что третий режим существенно опережает первый и второй по скорости добычи нефти. При этом первый режим по сравнению со вторым значительно эффективнее позволяет извлекать нефть в период с 2 до 9 часов считая со времени начала закачки ПАВ; далее первый режим начинает проигрывать второму. Довольно резкий изгиб второго графика можно объяснить тем, что только к моменту времени $t \approx 26000$ в ячейках накопилось достаточное количество молей ПАВ для того, чтобы неподвижная нефть начала активно вступать в реакцию и становиться подвижной.

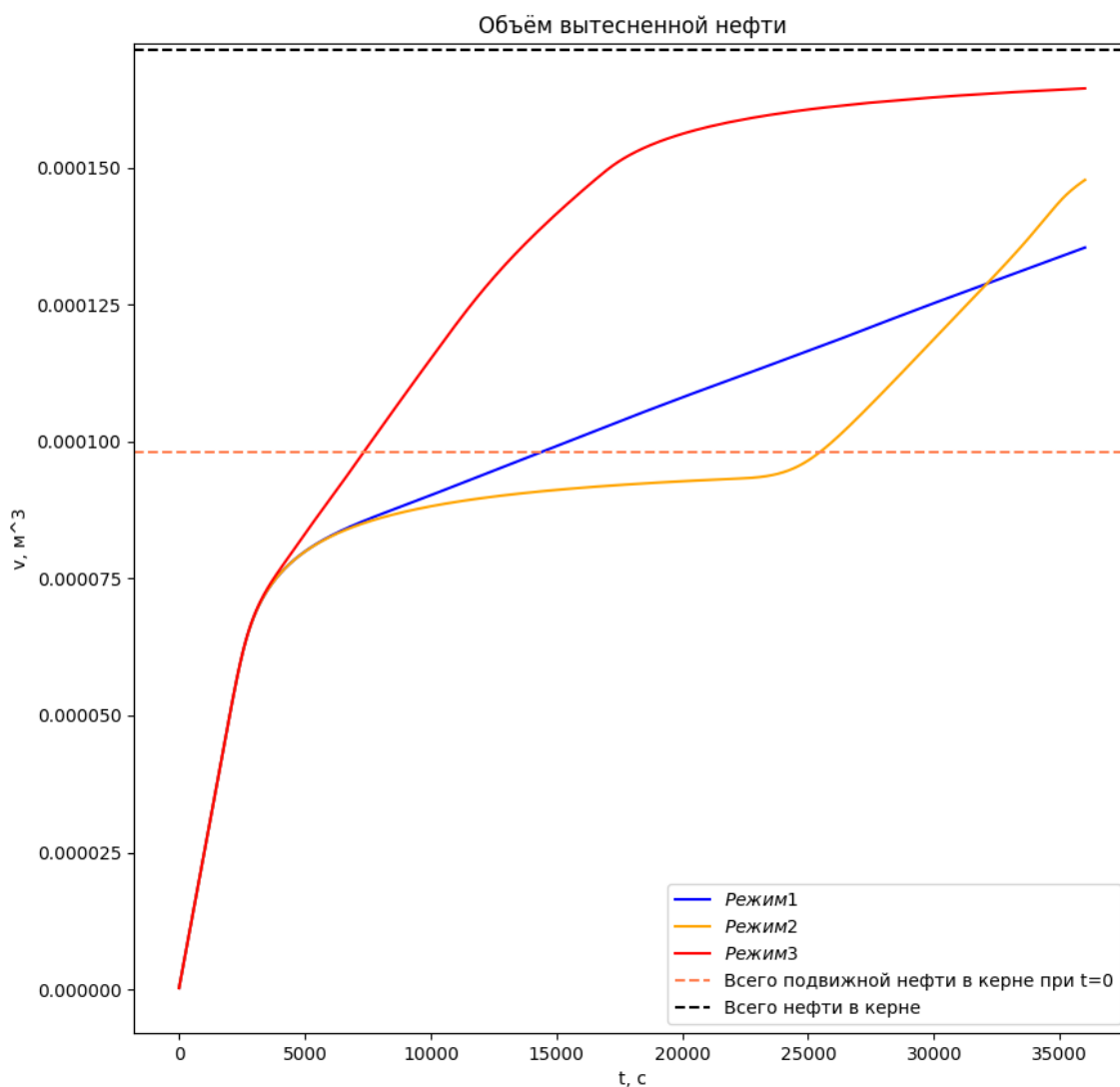


Рисунок 4.18 – Объём вытесненной нефти в соответствии с выбранным режимом закачки ПАВ.

Исходя из данных таблицы 4.6 можем утверждать, что использование третьего режима закачки ПАВ позволяет в 1,74 и 1,32 раза увеличить объём дополнительно добытой нефти по сравнению с первым и вторым режимами соответственно. Подытоживая вышесказанное, можно сделать вывод, что эффективнее всего при закачке ПАВ использовать третий режим.

Таблица 4.6 – Данные о количестве добытой нефти.

| № Режима | Массовая доля ПАВ при $t=0$ | Масса закачен- ного ПАВ, кг | Объём зака- ченного ПАВ, м^3 | Объём вытес- ненной нефти, м^3 | Объём дополни- тельно добытой нефти, м^3 |
|-------------|-----------------------------------|-----------------------------------|---|---|---|
| 1 | 2,50E-05 | 3,6012600E-05 | 3,078000E-08 | 1,3543100E-04 | 4,0436800E-05 |
| 2 | 1,00E-05 | 3,7393585E-05 | 3,196033E-08 | 1,4777100E-04 | 5,2776800E-05 |
| 3 | 5,00E-05 | 3,6235637E-05 | 3,097063E-08 | 1,6448000E-04 | 6,9485800E-05 |

В следующем исследовании были рассмотрены те же режимы закачки ПАВ с той разницей, что теперь помимо ПАВ в керн ещё равномерно закачивали полимер с массовой долей 2,5E-5.

Исходя из рисунка 4.19 и данных таблицы 4.7 можно сделать вывод, что третий режим закачки ПАВ при равномерной закачке полимера в керн даёт существенный выигрыш в скорости добычи нефти и несущественный выигрыш в объёме суммарно добытой и дополнительно добытой нефти за всё время.

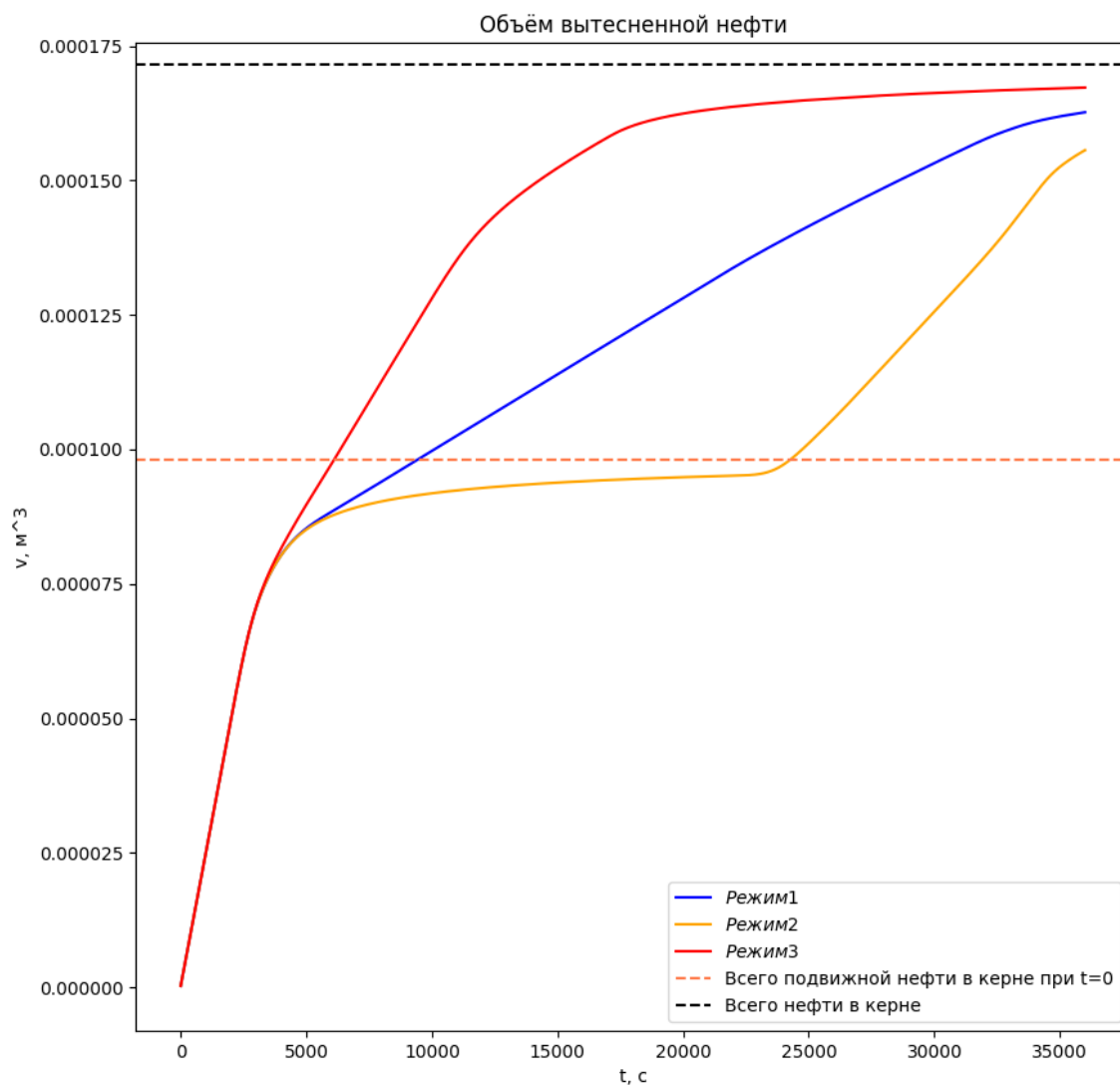


Рисунок 4.19 – Объём вытесненной нефти в соответствии с выбранным режимом закачки ПАВ.

Таблица 4.7 – Данные о количестве добытой нефти.

| № Режима | Массовая доля ПАВ при t=0 | Масса закачен- ного ПАВ, кг | Объём закачен- ного полимера, м ³ | Объём вытес- ненной нефти, м ³ | Объём дополни- тельно добытой нефти, м ³ |
|-------------|---------------------------------|-----------------------------------|--|---|---|
| 1 | 2,50E-05 | 3,6012600E-05 | 3,078000E-08 | 1,6266500E-04 | 6,7670800E-05 |
| 2 | 1,00E-05 | 3,7393585E-05 | 3,196033E-08 | 1,5561700E-04 | 6,0622800E-05 |
| 3 | 5,00E-05 | 3,6235637E-05 | 3,097063E-08 | 1,6725800E-04 | 7,2263800E-05 |

ЗАКЛЮЧЕНИЕ

В рамках данной работы:

1. Описана математическая модель процессов многофазной фильтрации в пористых средах для случая изотермических и несжимаемых жидкостей и разработана программа, моделирующая данный процесс.
2. Реализован модуль расчёта поля давления, при вычислении которого используются линейные базисные функции на равномерных конечноэлементных сетках. При тестировании данного модуля были подтверждены теоретические представления о порядке аппроксимации для выбранного базиса.
3. Реализован модуль расчёта нового состояния ячеек и для выбранных модельных данных подобраны оптимальные значения таких параметров, как шаг по времени и шаг разбиения сетки.
4. Исследованы зависимости объёмов всей вытесненной нефти и дополнительно добытой нефти от концентрации закачиваемого полимерного раствора. В ходе исследования для выбранных модельных данных был найден диапазон оптимальных значений концентрации полимера.
5. Исследованы зависимости объёмов всей вытесненной нефти и дополнительно добытой нефти от концентрации растворённого ПАВ в воде при закачке. В ходе исследования для выбранных модельных данных было найдено оптимальное значение концентрации ПАВ.

6. Рассмотрен случай одновременного применения ПАВ и полимера при закачке. Результат исследования показал, что для выбранной модели общее количество вытесненной нефти увеличивается в 1.25 раз по сравнению со случаем закачки с водой только ПАВ и в 1.74 раз по сравнению со случаем закачки с водой только полимера; объём дополнительно добытой нефти также увеличивается в 1.85 раз и 3.1 раз для каждого из случаев соответственно, что существенно повышает эффективность применения каждого из веществ. В дальнейшем возможно изучение экономической выгоды при различных способах закачки ПАВ и полимеров в пласт.
7. Исследованы разные режимы закачки полимера в керн: равномерная закачка, закачка с увеличением концентрации и с уменьшением концентрации. Результаты исследований показали, что по объёму дополнительно добытой нефти все способы закачки дают примерно одинаковый результат. Но для более быстрой добычи нефти эффективнее всего закачивать полимер либо с убыванием концентрации, либо равномерно (с одинаковой концентрацией). Использование ПАВ при закачке увеличивает разрыв в скорости добычи нефти между режимами, но в долгосрочной перспективе результаты применения каждого из режимов несущественно отличаются друг от друга.
8. Исследованы разные режимы закачки ПАВ в керн: равномерная закачка, закачка с увеличением концентрации и с уменьшением концентрации. Результате было выяснено, что третий режим существенно опережает первый и второй по скорости добычи нефти. Причем при закачке только ПАВ (без использования полимера) третий режим позволяет в 1,74 и 1,32 раза увеличить объём дополнительно добытой нефти по сравнению с первым и вторым режимами соответственно; а при использовании полимера при закачке ни один из режимов не даёт значительного выигрыша ни в объёме всей вытесненной нефти, ни в объёме дополнительно добытой нефти.

СПИСОК ЛИТЕРАТУРЫ

1. Соловейчик, Ю.Г. Метод конечных элементов для скалярных и векторных задач / Ю.Г. Соловейчик, М.Э. Рояк, М.Г. Персова. – Новосибирск: НГТУ, 2007. – 869 с.
2. Персова М. Г., Соловейчик Ю. Г., Патрушев И. И., Овчинникова А. С. Численное моделирование нефтедобычи с применением ПАВ-полимерного заводнения // Известия Саратовского университета. Новая серия. Серия Математика. Механика. Информатика. 2021. Т. 21, вып. 4. С. 544-558.
3. Патрушев И. И. Разработка и реализация вычислительных схем решения трёхмерной задачи многофазной фильтрации: выпускная квалификационная работа: 01.04.02 / И. И. Патрушев; науч. рук. М. Г. Персова; Новосиб. гос. техн. ун-т. - Новосибирск, 2018. – 60 с.
4. Азиз Х., Сеттари Э. Математическое моделирование пластовых систем: Пер. с англ. – М.: Недра, 1982. – 407 с.
5. Исследование эффективности использования полимерного заводнения в условиях наличия высокопроводящих слоев в задачах моделирования месторождений / Н.А. Крашенинник, А.А. Пешкичева, И.И. Патрушев // Наука. Технологии. Инновации: сборник научных трудов: [Всероссийская научная конференция молодых ученых], г. Новосибирск, в 9-и частях - Новосибирск: Изд-во НГТУ, 2019. Ч. 2. - 2019. - 261 с. : ил., табл.; с. 132 – 135. - ISBN 978-5-7782-4008-7.
6. Особенности построения и тестирования конечноэлементных СЛАУ для одномерного уравнения эллиптического типа: учебное пособие / А. Г. Задорожный. – Новосибирск: Изд-во НГТУ, 2022. – 88 с.
7. Willhite Paul G. Waterflooding – Society of Petroleum Engineers Richardson, TX, 1986.

8. Телков В.П., Каримов А.К., Мостаджеран М.Г., Ким С.В., Юсупходжаев М.А. Повышение нефтеизвлечения на месторождениях высоковязкой и тяжелой нефти с помощью полимерного заводнения // Нефтяное хозяйство. 2018. № 5. С. 60-63.
9. Щербакова, А. С. Разработка месторождений с применением ПАВ-полимерного заводнения / А. С. Щербакова. – Текст: непосредственный // Молодой ученый. – 2017. – № 36 (170). – С. 37-40.
10. Лэйк Л. Основы методов увеличения нефтеотдачи. – Остин: Университет Техас-Остин, 2005. – 449 с.
11. Мордвинов В.А., Поплыгин В.В., Поплыгина И.С. Варианты полимерного заводнения залежи с высоковязкой нефтью // Вестник Пермского национального исследовательского политехнического университета. Геология. Нефтегазовое и горное дело. – 2015. – № 14. – С. 39-51. БОТ: 10.15593/2224-9923/2015.14.5
12. Мазо А.Б., Поташев К.А., Баушин В.В., Булыгин Д.В. Расчет полимерного заводнения нефтяного пласта по модели фильтрации с фиксированной трубкой тока. Георесурсы. 2017. Т. 19. № 1. С. 15-20.

ПРИЛОЖЕНИЕ 1. КОД ПРОГРАММЫ

```
#include<iostream>
#include<fstream>
#include<vector>
#include<Set>
#include<math.h>
#include<conio.h>

using namespace std;

#define test3

#ifdef test0
double test(double x)
{
    return 2;
}
#endif
#ifdef test1
double test(double x)
{
    return x;
}
#endif
#ifdef test2
double test(double x)
{
    return exp(x);
}
#endif
#ifdef test3
double test(double x)
{
    return x * x;
}
#endif

static class Filtration_using_polymer_and_surfactants
{
public:

    const int max_iter = 10000;
    const double eps = 1e-10;
    const double eps1 = 1e-8;
    const double eps_s = 1e-8;

    int n; // количество узлов
    int ke = 0; // номер кэ
    double a_time, b_time; // начало и конец по сетки по времени
    double a, b, h, k_discharge; // начало и конец области, шаг
    разбиения, коэф разрядки
    double K, F; // структурная проницаемость, структурная пористость
    double nu_oil; // вязкость нефти
    double Q_0; // начальный поток
```

```

double s_s; // площадь сечения керна
double dt, dt1, t; // шаг по времени
double total_V_out_oil = 0, total_V_out_water = 0;
double s_crit_oil, s_res_oil; // минимальная и остаточные
насыщенности
vector<double> summ_v, M, nu_water, density, k_max, s_transitional,
s_res_oil_vec;
vector<vector<double>> Q_m, V_m, s, V_out, k, alfa,
discrete_nu_func, x, amount_of_substance;

vector<double> nodes;
vector<double> pp, Q;
vector<double> global_b;
vector<vector<double>> G;
vector<int> ia, ja;
vector<double> al, au, di, f;
vector<vector<int>> elems;
vector<double> y_vec, q;

ofstream fpressure;
ofstream fcalc;
ofstream ftotal_oil;
ofstream ftotal_water;
ofstream fxi;
ofstream ft_grid;
ofstream fnu_out;
ofstream fs_res;
ofstream ftick;
ofstream fparity;
ofstream fpress;

double Function_nu_in_point(double x_polymer)
{
    if (x_polymer < discrete_nu_func[discrete_nu_func.size() /
2][0])
    {
        if (x_polymer < discrete_nu_func[discrete_nu_func.size()
/ 4][0])
        {
            for (int i = discrete_nu_func.size() / 4 - 1; i >=
0; i--)
            {
                if (x_polymer >= discrete_nu_func[i][0])
                {
                    return discrete_nu_func[i][1] +
(discrete_nu_func[i + 1][1] - discrete_nu_func[i][1])
* (x_polymer -
discrete_nu_func[i][0]) / (discrete_nu_func[i + 1][0] -
discrete_nu_func[i][0]);
                }
            }
        }
        else
        {
            for (int i = discrete_nu_func.size() / 4 + 1; i <=
discrete_nu_func.size() / 2; i++)
            {
                if (x_polymer <= discrete_nu_func[i][0])

```



```

        {
            return discrete_nu_func[i - 1][1] +
(discrete_nu_func[i][1] - discrete_nu_func[i - 1][1])
                * (x_polymer - discrete_nu_func[i -
1][0]) / (discrete_nu_func[i][0] - discrete_nu_func[i - 1][0]);
        }
    }
}
else
{
    if (x_polymer < discrete_nu_func[discrete_nu_func.size()
- discrete_nu_func.size() / 4][0])
    {
        for (int i = discrete_nu_func.size() -
discrete_nu_func.size() / 4 - 1; i >= discrete_nu_func.size() / 2; i--)
        {
            if (x_polymer >= discrete_nu_func[i][0])
            {
                return discrete_nu_func[i][1] +
(discrete_nu_func[i + 1][1] - discrete_nu_func[i][1])
                    * (x_polymer -
discrete_nu_func[i][0]) / (discrete_nu_func[i + 1][0] -
discrete_nu_func[i][0]);
            }
        }
    }
    else
    {
        for (int i = discrete_nu_func.size() -
discrete_nu_func.size() / 4 + 1; i < discrete_nu_func.size(); i++)
        {
            if (x_polymer <= discrete_nu_func[i][0])
            {
                return discrete_nu_func[i - 1][1] +
(discrete_nu_func[i][1] - discrete_nu_func[i - 1][1])
                    * (x_polymer - discrete_nu_func[i -
1][0]) / (discrete_nu_func[i][0] - discrete_nu_func[i - 1][0]);
            }
        }
    }
}

void Reading_viscosity_of_water(string nu_water)
{
    ifstream f_nu_water;

    f_nu_water.open(nu_water + ".txt");

    int discrete_nu_func_size;
    f_nu_water >> discrete_nu_func_size;
    discrete_nu_func.resize(discrete_nu_func_size);
    for (int i = 0; i < discrete_nu_func_size; i++)
    {
        discrete_nu_func[i].resize(2);
        f_nu_water >> discrete_nu_func[i][0] >>
discrete_nu_func[i][1];
    }
}

```

```

    }
}

void Reading_grid(string grid)
{
    ifstream f_grid;

    f_grid.open(grid + ".txt");

    f_grid >> a >> b >> h >> k_discharge;
    f_grid >> a_time >> b_time >> dt;
}

void Reading_parameters(string parameters)
{
    ifstream f_param;

    f_param.open(parameters + ".txt");

    f_param >> K;
    f_param >> F;
    f_param >> nu_oil;
    f_param >> M[0] >> M[1] >> M[2] >> M[3];
    f_param >> x[1][0] >> x[2][0];
    f_param >> s_s;
    f_param >> Q_0;
    f_param >> density[0] >> density[1];
    f_param >> s_crit_oil;
    f_param >> s_res_oil;
}

void Building_grid()
{
    cout << "Grid:";
    for (double w = a; w < b - eps1; w += h)
    {
        nodes.push_back(w);
        h *= k_discharge;
        cout << endl << w << " ";
    }

    nodes.push_back(b);
    n = nodes.size();
    cout << endl << b << endl;
}

void Resize()
{
    s.resize(2), k.resize(2), alfa.resize(2),
amount_of_substance.resize(4), x.resize(3);
    for (size_t m = 0; m < 2; m++)
    {
        s[m].resize(n);
        k[m].resize(n);
        alfa[m].resize(n);
    }
    for (size_t j = 0; j < x.size(); j++)
        x[j].resize(n);
}

```

```

    for (size_t j = 0; j < amount_of_substance.size(); j++)
        amount_of_substance[j].resize(n);

    k_max.resize(2);
    G.resize(2);
    M.resize(4);
    summ_v.resize(2);
    nu_water.resize(n);
    density.resize(2);
    q.resize(n);
    di.resize(n);
    ia.resize(n + 1);
    pp.resize(n);
    global_b.resize(n, 0.0);
    Q.resize(n);
    elems.resize(n - 1);
    s_res_oil_vec.resize(n);
    s_transitional.resize(n);

    int j = 0;
    for (int l = 0; l < n - 1; l++)
    {
        elems[l].resize(2);
        elems[l][0] = j;
        elems[l][1] = j + 1;
        j++;
    }

    Q_m.resize(2), V_m.resize(2), V_out.resize(2);
    for (int l = 0; l < 2; l++)
    {
        Q_m[l].resize(n);
        V_m[l].resize(n);
        V_out[l].resize(n);
    }
}

void Initialization()
{
    for (size_t e = 0; e < n; e++)
    {
        s[0][e] = 0.3;
        s[1][e] = 0.7;
        k[0][e] = s[0][e];
        k[1][e] = s[1][e];
        s_res_oil_vec[e] = s_res_oil;
    }
    x[0][0] = 1.0 - x[1][0] - x[2][0];
    nu_water[0] = Function_nu_in_point(x[1][0]);
    for (size_t e = 1; e < n; e++)
    {
        x[2][e] = 0.0;
        x[1][e] = 0.0;
        x[0][e] = 1.0;
        nu_water[e] = Function_nu_in_point(x[1][e]);
    }
    k_max[0] = 1.0;

```

```

        k_max[1] = 1.0;
    }

    void Portrait()
    {
        vector<set<int>> portrait(n);
        int ja_size = 0;
        for (int k = 0; k < nodes.size(); k++)
        {
            for (int i = 0; i < elems.size(); i++)
            {
                for (int j = 0; j < 2; j++)
                {
                    if (elems[i][0] == k || elems[i][1] == k)
                        if (elems[i][j] < k) // пробегаемся по
элементу, если есть узел меньше, то добавляем его
                        portrait[k].insert(elems[i][j]);
                }
            }

            //Подсчёт кол-ва элементов для ja
            //cout << "\t----Portrait----" << endl;
            for (int i = 0; i < portrait.size(); i++)
            {
                //cout << i << ": ";
                for (auto j = portrait[i].begin(); j !=
portrait[i].end(); j++)
                {
                    //cout << *j << " ";
                    ja_size++;
                }
                //cout << endl;
            }

            ja.resize(ja_size);

            // Заполнение векторов ia и ja
            for (int i = 2; i < ia.size(); i++)
                ia[i] = ia[i - 1] + portrait[i - 1].size();

            for (int index = 0, i = 1; i < portrait.size(); i++)
                for (int temp : portrait[i]) // темп берёт в себя каждое
значение портрета
                    ja[index++] = temp;
        }

        void Building_G_local()
        {
            for (int ii = 0; ii < 2; ii++)
                G[ii].resize(2);
            G[0][0] = K * (k[0][ke] / nu_water[ke] + k[1][ke] / nu_oil) /
(1.0 * h);
            G[0][1] = -K * (k[0][ke] / nu_water[ke] + k[1][ke] / nu_oil) /
(1.0 * h);
            G[1][0] = -K * (k[0][ke] / nu_water[ke] + k[1][ke] / nu_oil) /
(1.0 * h);

```

```

        G[1][1] = K * (k[0][ke] / nu_water[ke] + k[1][ke] / nu_oil) /
(1.0 * h);
    }

    void Boundary_conditions_1(int t_cur) // cтp 237
    {
        double B = 1e+10, sum = 0;
        double x1, x2;
        x1 = nodes[0];
        x2 = nodes[nodes.size() - 1];
        double res1 = test(x1) * 101325.0;
        double res2 = test(x2) * 101325.0;
        di[nodes.size() - 1] = B;
        global_b[nodes.size() - 1] = B * res2;
        /*di[0] = B;
        global_b[0] = B * res1;*/
    }

    void Boundary_conditions_2()
    {
        global_b[0] += Q_0;
        //global_b[0] -= K * (k[0][0] / nu_water[0] + k[1][0] /
nu_oil);
    }

    void LU_decomposition()
    {
        //cout << endl << endl << "          Start LU_decomposition" <<
endl;

        int j0 = 0;
        int j = 0;
        int jbeg = 0;
        int jend = 0;
        int j0j = 0;
        int jjbeg = 0;
        int jjend = 0;
        int indau = 0;
        int indal = 0;
        double cu = 0;
        double cl = 0;
        //int ii = 0;

        for (int i = 1; i < n; ++i)
        {
            double sumdi = 0;
            j0 = i - (ia[i + 1] - ia[i]);
            for (int ii = ia[i]; ii < ia[i + 1]; ++ii)
            {
                j = ii - ia[i] + j0;
                jbeg = ia[j];
                jend = ia[j + 1];
                if (jbeg < jend)
                {
                    j0j = j - (jend - jbeg);
                    jjbeg = max(j0, j0j);
                    jjend = min(j, i - 1);
                    cl = 0;

```

```

        for (int k = 0; k < jjend - jjbeg; k++)
        {
            indau = ia[j] + jjbeg - j0j + k;
            indal = ia[i] + jjbeg - j0 + k;
            cl = cl + au[indau] * al[indal];
        }
        al[ii] = al[ii] - cl;
        cu = 0;
        for (int k = 0; k < jjend - jjbeg; k++)
        {
            indal = ia[j] + jjbeg - j0j + k;
            indau = ia[i] + jjbeg - j0 + k;
            cu = cu + au[indau] * al[indal];
        }
        au[ii] = au[ii] - cu;
    }
    au[i] = au[i] / di[j];
    sumdi += al[ii] * au[ii];
}
di[i] -= sumdi;
}

/*cout << endl << "di: ";
for (int i = 0; i < n; i++)
    cout << di[i] << ' ';

cout << endl << "al: ";
for (int i = 0; i < ia[n]; i++)
    cout << al[i] << ' ';

cout << endl << "au: ";
for (int i = 0; i < ia[n]; i++)
    cout << au[i] << ' ';
cout << endl;

cout << endl << "          End LU_decomposition" << endl <<
endl;*/
}

void straight_run()
{
    //cout << endl << "          Start straight_run" << endl;
    y_vec.resize(n);

    double q = 0;
    for (int i = 0; i < n; i++)
    {
        int m = ia[i + 1];
        int w = ia[i];
        int j0 = i - (m - w);
        y_vec[i] = global_b[i];
        int j = 0;
        for (int k = w; k < m; k++, j++)
        {
            q = al[k] * y_vec[j0 + j];
            y_vec[i] -= q;
        }
    }
}

```

```

        y_vec[i] /= di[i];
    }

    /*cout << endl << "y_vec: ";
    for (int i = 0; i < n; i++)
        cout << y_vec[i] << ' ';
    cout << endl;

    cout << endl << "        End straight_run" << endl;*/
}

void reverse()
{
    //cout << endl << endl << "        Start reverse" << endl;

    int p = 0;
    double w = 0;
    for (int i = n - 2; i >= 0; i--)
    {
        for (int k = ia[i + 1]; k < ia[i + 2]; k++)
        {
            p = k + i + 1 - ia[i + 2];
            w = au[k] * y_vec[i + 1];
            y_vec[p] -= au[k] * y_vec[i + 1];
        }
    }

    fpressure << endl << "res_vec: " << endl;
    for (int i = 0; i < n; i++)
    {
        fpressure << fixed;
        fpressure.precision(5);
        fpressure << scientific;
        fpressure << y_vec[i] << endl;
    }

    q = y_vec;
    //cout << endl << "        End reverse" << endl;
}

void Global_Matrix_and_b(int t)
{
    int k = 0;
    al.resize(ia[n], 0); au.resize(ia[n], 0);

    for (int k = 0; k < nodes.size() - 1; k++) // по каждому
элементу
    {
        Building_G_local();

        for (int i = 0; i < 2; i++)
        {
            int i_i = elems[k][i];
            di[k + i] += G[i][i];
            for (int j = 0; j < i; j++)
            {
                int j_j = elems[k][j];
                if (i_i < j_j)

```

```

        swap(i_i, j_j);
        for (k = ia[i_i]; ja[k] - j_j != 0; k++);
        al[k] += G[i][j]; // для глобальной матрицы A
    }
    }
    ke++;
}

/*cout << endl << endl << "al: ";
for (int i = 0; i < al.size(); i++)
    cout << al[i] << " ";

cout << endl << "di: ";
for (int i = 0; i < di.size(); i++)
    cout << di[i] << " ";

cout << endl << "global_b: ";
for (int i = 0; i < global_b.size(); i++)
    cout << global_b[i] << " ";*/

Boundary_conditions_2();
Boundary_conditions_1(t);

au = al;

/*cout << endl << endl << "al: ";
for (int i = 0; i < al.size(); i++)
    cout << al[i] << " ";

cout << endl << "di: ";
for (int i = 0; i < di.size(); i++)
    cout << di[i] << " ";

cout << endl << "global_b: ";
for (int i = 0; i < global_b.size(); i++)
    cout << global_b[i] << " ";*/

LU_decomposition();
straight_run();
reverse();

for (int i = 0; i < n; i++)
    global_b[i] = di[i] = 0;

for (int i = 0; i < ia[n]; i++)
    al[i] = 0;
}

void calc_of_phase_flows()
{
    alfa[0][0] = 1.0;
    alfa[1][0] = 0.0;
    for (int e = 1; e < n; e++)
    {
        alfa[0][e] = k[0][e] / nu_water[e] / (k[0][e] /
nu_water[e] + k[1][e] / nu_oil);
        alfa[1][e] = k[1][e] / nu_oil / (k[0][e] / nu_water[e] +
k[1][e] / nu_oil);
    }
}

```



```

    }
    if (isnan(alfa[0][1]) == true)
    {
        cout << "!!";
    }

    Q[0] = Q_0 * s_s;
    for (int e = 1; e < n; e++) // расчёт перетекающего потока
    смеси через грани кэ
    {
        //Q[e] = Q[0];
        Q[e] = s_s * K * (k[0][e] / nu_water[e] + k[1][e] /
    nu_oil) * (pp[e] - pp[e - 1]) / (nodes[e] - nodes[e - 1]);
    }

    for (int e = 0; e < n; e++) // расчёт перетекающего потока
    смеси через грани кэ
    {
        for (int m = 0; m < 2; m++)
        {
            Q_m[m][e] = alfa[m][e] * fabs(Q[0]);
        }
    }

    void calc_time_step()
    {
        for (int e = 1; e < n; e++)
        {
            if (s[1][e] > s_crit_oil)
            {
                dt1 = fabs(nodes[e] - nodes[e - 1]) * s_s *
    (s[1][e] - s_res_oil) * F / Q_m[1][e];
                if (dt1 < dt)
                    dt = dt1;
            }
        }
        fcalc << endl << "dt: " << dt;
    }

    void calc_of_phase_volumes_and_saturations()
    {
        for (int e = 0; e < n; e++)
        {
            for (int m = 0; m < 2; m++)
            {
                V_out[m][e] = Q_m[m][e] * dt;
            }
        }

        fcalc << endl << "V_water_in: " << V_out[0][0];
        fcalc << endl << "V_oil_in: " << V_out[1][0] << endl;

        bool flag = false;
        double fraction_of_oil = 0.0;
        for (int e = 1; e < n; e++)
        {

```

```

V_m[0][e] = fabs(nodes[e] - nodes[e - 1]) * s_s *
s[0][e] * F;
V_m[0][e] += V_out[0][e - 1] - V_out[0][e];
V_m[1][e] = fabs(nodes[e] - nodes[e - 1]) * s_s *
s[1][e] * F;
V_m[1][e] += V_out[1][e - 1] - V_out[1][e];

for (int m = 0; m < 2; m++)
{
    s[m][e] = V_m[m][e] / (fabs(nodes[e] - nodes[e -
1]) * F * s_s);
}
if (s[0][e] < 0)
{
    cout << "!!";
}
if (s[1][e] == 0.31)
{
    cout << "!!";
}

if (V_out[1][e] > fabs(nodes[e] - nodes[e - 1]) * F * s_s
* (s[1][e] - s_res_oil_vec[e]))
{
    double V_out_start = V_out[1][e];
    double V_out_mix = V_out[1][e] + V_out[0][e];

    V_out[0][e] += V_out[1][e];
    V_out[1][e] = fabs(nodes[e] - nodes[e - 1]) * F *
s_s * (s[1][e] - s_res_oil_vec[e]);
    V_out[0][e] -= V_out[1][e];
    double V_out_mix_res = V_out[1][e] + V_out[0][e];

    s[1][e] = s_res_oil_vec[e];
}

fcalc << endl << "V[0][" << e << "]: " << V_m[0][e];
fcalc << endl << "V[1][" << e << "]: " << V_m[1][e];
fcalc << endl << "s[0][" << e << "]: " << s[0][e];
fcalc << endl << "s[1][" << e << "]: " << s[1][e];
fcalc << endl << "Vpor = " << fabs(nodes[e] - nodes[e -
1]) * s_s * F << endl;

if (s[0][e] + s[1][e] > 1.0 + eps_s)
{
    int z = 1;
}
}

fcalc << endl << endl << "V_water_out: " << V_out[0][n - 1];
fcalc << endl << "V_oil_out: " << V_out[1][n - 1];

for (int m = 0; m < 2; m++)
{
    summ_v[m] = 0;
    for (int e = 0; e < V_out.size(); e++)
        summ_v[m] += V_out[m][e];
}

```

```

    }

    void calc_of_new_x()
    {
        double s_mobile = 0;
        double mass;
        double portion_oil, portion_surfactant;
        double amount_moll_oil, amount_moll_surfactant;
        for (int e = 1; e < n; e++) // количество вещества (вода,
полимер, пав, нефть)
        {
            for (int l = 0; l < 3; l++) // вода, полимер, пав
                amount_of_substance[l][e] = (density[0] *
V_out[0][e - 1] * x[l][e - 1] + density[0] * x[l][e] * (fabs(nodes[e] -
nodes[e - 1]) * F * s_s * s[0][e] - V_out[0][e])) / M[l];

            mass = V_m[l][e] * density[l];
            amount_of_substance[3][e] = mass / M[3];

            for (int l = 0; l < 3; l++)
                x[l][e] = amount_of_substance[l][e] * M[l] /
(amount_of_substance[0][e] * M[0] + amount_of_substance[1][e] * M[1] +
amount_of_substance[2][e] * M[2]);

            //x[0][e] = 1.0 - x[1][e] - x[2][e];

            if (x[2][e] > 0.5e-5)
            {
                portion_oil = int(amount_of_substance[3][e] /
0.00002);

                amount_moll_surfactant = 0.0000005 * portion_oil;
                amount_moll_oil = portion_oil * 0.00002;
                if (amount_moll_surfactant >
amount_of_substance[2][e])
                {
                    portion_surfactant =
int(amount_of_substance[2][e] / 0.0000005);
                    amount_moll_oil = 0.00002 *
portion_surfactant;
                }

                double V_new = amount_moll_oil * M[3] / density[1];
                s_res_oil_vec[e] -= V_new / (fabs(nodes[e] -
nodes[e - 1]) * F * s_s);
                /*if (s_res_oil_vec[e] < 0.4)
                    int y = 0;*/
                if (s_res_oil_vec[e] < 0.0)
                    s_res_oil_vec[e] = 0;
            }
            s_mobile += (s[1][e] - s_res_oil_vec[e]) * F * s_s *
fabs(nodes[1] - nodes[0]);
        }
        fs_res << s_mobile << endl;
        for (int e = 1; e < n - 1; e++) // вязкость
            nu_water[e] = Function_nu_in_point(x[1][e]);
        fnu_out << nu_water[n - 2] << endl;
    }

```

```

void Calc() // расчёт нового состояния ячеек
{
    calc_of_phase_flows();
    calc_time_step();
    calc_of_phase_volumes_and_saturations();
    calc_of_new_x();

    /*cout << endl << endl << "nu_water: ";
    for (int e = 0; e < n - 1; e++)
    {
        cout << nu_water[e] << " ";
    }*/

    fxi << endl;
    for (int e = 1; e < n; e++)
    {
        fxi << fixed;
        fxi.precision(3);
        fxi << scientific << x[1][e] << " ";
    }

    total_V_out_oil += V_out[1][n - 1];
    total_V_out_water += V_out[0][n - 1];

    ftotal_oil << total_V_out_oil << endl;
    ftotal_water << total_V_out_water << endl;
    fcalc << endl << endl << "total_V_out_oil: " <<
total_V_out_oil;
    fcalc << endl << endl << "total_V_out_water: " <<
total_V_out_water;
    fcalc << endl << "Суммарный перетекающий объём воды: " <<
summ_v[0];
    fcalc << endl << "Суммарный перетекающий объём нефти: " <<
summ_v[1];

    fcalc << endl << endl << "Перетекающие объёмы для каждой фазы
по границе каждого кэ:" << endl;
    for (int i = 0; i < 2; i++)
    {
        for (int j = 0; j < V_out[i].size(); j++)
        {
            fcalc << fixed;
            fcalc.precision(5);
            fcalc << scientific << V_out[i][j] << " ";
        }
        fcalc << endl;
    }
}

Filtration_using_polymer_and_surfactants(string t_grid, string
parameters, string grid, string nu_water,
string pressure, string saturation, string total_oil, string
total_water, string x, string nu_out,
string tick, string parity, string s_res, string press)
{
    setlocale(LC_ALL, "Russian");
    Reading_grid(grid);
    Building_grid();

```

```

Resize();
Reading_parameters(parameters);
Reading_viscosity_of_water(nu_water);
Initialization();
Portrait();

fpressure.open(pressure + ".txt");
fcalc.open(saturation + ".txt");
ftotal_oil.open(total_oil + ".txt");
ftotal_water.open(total_water + ".txt");
fxi.open(x + ".txt");
fnu_out.open(nu_out + ".txt");
ft_grid.open(t_grid + ".txt");
fs_res.open(s_res + ".txt");
ftick.open(tick + ".txt");
fparity.open(parity + ".txt");
fpress.open(press + ".txt");
double time_tick = 0;
double tick_ = 200;
int parity_index = 0;

for (t = a_time; t <= b_time; t += dt)
{
    if (time_tick == tick_)
    {
        time_tick = 0;
    }
    fcalc << endl << "Время: " << t;
    ft_grid << t << endl;
    fpressure << endl << "Время: " << t;
    fxi << endl << "Время: " << t;
    ke = 0;
    double ss;
    Global_Matrix_and_b(t);
    pp = q;
    Calc();
    if (parity_index % 1 == 0)
    {
        fparity << total_V_out_oil << endl;
    }
    if (time_tick == 0)
    {
        for (size_t e = 1; e < s[1].size(); e++)
        {
            ftick << s[1][e] << " ";
            //ftick << s_res_oil_vec[e] << " ";
        }
        ftick << endl;
    }
    for (size_t e = 0; e < k[0].size(); e++)
    {
        s_transitional[e] = (s[0][e] - 0.0) / (1.0 -
s_res_oil_vec[e] - 0.0);

        k[0][e] = 0.05 * pow(s_transitional[e], 1.5);
        k[1][e] = 0.425 * pow(1.0 - s_transitional[e],
2.0);

        if (isnan(k[0][e]) == true)

```

```

        {
            cout << "!!";
        }
        if (k[0][e] >= 1.0 or k[1][e] > 1.0)
        {
            cout << "!!";
        }
        if (e % 1 == 0)
        {
            fpress << y_vec[e] << endl;
        }
    }
    time_tick += dt;
    parity_index++;
}
fpressure.close();
fcalc.close();
ftotal_oil.close();
ftotal_water.close();
fxi.close();
ft_grid.close();
fnu_out.close();
fs_res.close();
ftick.close();
fparity.close();
fpress.close();
}

};

int main()
{
    Filtration_using_polymer_and_surfactants solve =
    Filtration_using_polymer_and_surfactants("t_grid", "parameters",
        "grid", "nu_water", "pressure", "saturation", "total_oil",
        "total_water", "x", "nu_out", "tick", "parity", "s_res", "press");
    Filtration_using_polymer_and_surfactants solve1 =
    Filtration_using_polymer_and_surfactants("t_grid1", "parameters1",
        "grid", "nu_water", "pressure", "saturation1", "total_oil1",
        "total_water1", "x", "nu_out1", "tick1", "parity1", "s_res1", "press1");
    Filtration_using_polymer_and_surfactants solve2 =
    Filtration_using_polymer_and_surfactants("t_grid2", "parameters2",
        "grid", "nu_water", "pressure", "saturation2", "total_oil2",
        "total_water2", "x", "nu_out2", "tick2", "parity2", "s_res2", "press2");
    Filtration_using_polymer_and_surfactants solve3 =
    Filtration_using_polymer_and_surfactants("t_grid3", "parameters3",
        "grid", "nu_water", "pressure", "saturation3", "total_oil3",
        "total_water3", "x", "nu_out3", "tick3", "parity3", "s_res3", "press3");
    Filtration_using_polymer_and_surfactants solve4 =
    Filtration_using_polymer_and_surfactants("t_grid3", "parameters4",
        "grid", "nu_water", "pressure", "saturation4", "total_oil4",
        "total_water4", "x", "nu_out3", "tick4", "parity3", "s_res3", "press3");
    system("C:\\Users\\olga\\source\\repos\\Neft'\\pythonProject1\\main
.py");
}

```

ПРИЛОЖЕНИЕ 2. ФРАГМЕНТ ПРОГРАММЫ ОТРИСОВКИ ГРАФИКОВ

```
import numpy as np
import numpy.random
import matplotlib
import matplotlib.pyplot as plt
from numpy import loadtxt
import csv
from matplotlib.ticker import NullLocator, LinearLocator
import matplotlib.ticker as ticker

ft = "C:\\Users\\olga\\source\\repos\\Neft'\\t_grid.txt"
f2 = "C:\\Users\\olga\\source\\repos\\Neft'\\total_oil.txt"
f3 = "C:\\Users\\olga\\source\\repos\\Neft'\\total_water.txt"
f4 = "C:\\Users\\olga\\source\\repos\\Neft'\\s_res.txt"
f21 = "C:\\Users\\olga\\source\\repos\\Neft'\\total_oil1.txt"
f31 = "C:\\Users\\olga\\source\\repos\\Neft'\\total_water1.txt"
f22 = "C:\\Users\\olga\\source\\repos\\Neft'\\total_oil2.txt"
f32 = "C:\\Users\\olga\\source\\repos\\Neft'\\total_water2.txt"
f23 = "C:\\Users\\olga\\source\\repos\\Neft'\\total_oil3.txt"
f33 = "C:\\Users\\olga\\source\\repos\\Neft'\\total_water3.txt"
f24 = "C:\\Users\\olga\\source\\repos\\Neft'\\total_oil4.txt"
f34 = "C:\\Users\\olga\\source\\repos\\Neft'\\total_water4.txt"
f25 = "C:\\Users\\olga\\source\\repos\\Neft'\\total_oil5.txt"
f35 = "C:\\Users\\olga\\source\\repos\\Neft'\\total_water5.txt"
fnu = "C:\\Users\\olga\\source\\repos\\Neft'\\nu_out.txt"
fnu1 = "C:\\Users\\olga\\source\\repos\\Neft'\\nu_out1.txt"
fnu2 = "C:\\Users\\olga\\source\\repos\\Neft'\\nu_out2.txt"
fnu3 = "C:\\Users\\olga\\source\\repos\\Neft'\\nu_out3.txt"
fnu4 = "C:\\Users\\olga\\source\\repos\\Neft'\\nu_out4.txt"
fnu5 = "C:\\Users\\olga\\source\\repos\\Neft'\\nu_out5.txt"
fpercent = "C:\\Users\\olga\\source\\repos\\Neft'\\percent.txt"
fpercent1 = "C:\\Users\\olga\\source\\repos\\Neft'\\percent1.txt"
fpercent2 = "C:\\Users\\olga\\source\\repos\\Neft'\\percent2.txt"
fparity = "C:\\Users\\olga\\source\\repos\\Neft'\\parity.txt"
fparity1 = "C:\\Users\\olga\\source\\repos\\Neft'\\parity1.txt"
fparity2 = "C:\\Users\\olga\\source\\repos\\Neft'\\parity2.txt"
fparity3 = "C:\\Users\\olga\\source\\repos\\Neft'\\parity3.txt"
fpercent_time = "C:\\Users\\olga\\source\\repos\\Neft'\\percent_time.txt"
fpercent_time1 =
"C:\\Users\\olga\\source\\repos\\Neft'\\percent_time1.txt"
fpercent_time2 =
"C:\\Users\\olga\\source\\repos\\Neft'\\percent_time2.txt"
fpress = "C:\\Users\\olga\\source\\repos\\Neft'\\press.txt"
fpress1 = "C:\\Users\\olga\\source\\repos\\Neft'\\press1.txt"
fpress2 = "C:\\Users\\olga\\source\\repos\\Neft'\\press2.txt"

with open(ft, 'r') as file:
    t = [float(line.strip()) for line in file]
with open(f2, 'r') as file:
    oil = [float(line.strip()) for line in file]
with open(f3, 'r') as file:
    water = [float(line.strip()) for line in file]
with open(f4, 'r') as file:
```

```

s_res = [float(line.strip()) for line in file]
with open(f21, 'r') as file:
    oil1 = [float(line.strip()) for line in file]
with open(f31, 'r') as file:
    water1 = [float(line.strip()) for line in file]
with open(f22, 'r') as file:
    oil2 = [float(line.strip()) for line in file]
with open(f32, 'r') as file:
    water2 = [float(line.strip()) for line in file]
with open(f23, 'r') as file:
    oil3 = [float(line.strip()) for line in file]
with open(f33, 'r') as file:
    water3 = [float(line.strip()) for line in file]
with open(f24, 'r') as file:
    oil4 = [float(line.strip()) for line in file]
with open(f34, 'r') as file:
    water4 = [float(line.strip()) for line in file]
with open(f25, 'r') as file:
    oil5 = [float(line.strip()) for line in file]
with open(f35, 'r') as file:
    water5 = [float(line.strip()) for line in file]
with open(fnu, 'r') as file:
    nu = [float(line.strip()) for line in file]
with open(fnu1, 'r') as file:
    nu1 = [float(line.strip()) for line in file]
with open(fnu2, 'r') as file:
    nu2 = [float(line.strip()) for line in file]
with open(fnu3, 'r') as file:
    nu3 = [float(line.strip()) for line in file]
with open(fnu4, 'r') as file:
    nu4 = [float(line.strip()) for line in file]
with open(fnu5, 'r') as file:
    nu5 = [float(line.strip()) for line in file]
with open(fpercent, 'r') as file:
    percent = [float(line.strip()) for line in file]
with open(fpercent1, 'r') as file:
    percent1 = [float(line.strip()) for line in file]
with open(fpercent2, 'r') as file:
    percent2 = [float(line.strip()) for line in file]
with open(fparity, 'r') as file:
    parity = [float(line.strip()) for line in file]
with open(fparity1, 'r') as file:
    parity1 = [float(line.strip()) for line in file]
with open(fparity2, 'r') as file:
    parity2 = [float(line.strip()) for line in file]
with open(fparity3, 'r') as file:
    parity3 = [float(line.strip()) for line in file]
with open(fpercent_time, 'r') as file:
    percent_time = [float(line.strip()) for line in file]
with open(fpercent_time1, 'r') as file:
    percent_time1 = [float(line.strip()) for line in file]
with open(fpercent_time2, 'r') as file:
    percent_time2 = [float(line.strip()) for line in file]
with open(fpress, 'r') as file:
    press = [float(line.strip()) for line in file]
with open(fpress1, 'r') as file:
    press1 = [float(line.strip()) for line in file]
with open(fpress2, 'r') as file:

```



```

press2 = [float(line.strip()) for line in file]

plt.figure(10, figsize=(10, 10))
plt.title("Объём вытесненной нефти")
plt.plot(t, oil, color='orange', label='$1$')
plt.plot(t, oil1, color='red', label='$2$')
plt.plot(t, oil2, color='green', label='$3$')
plt.axhline(y=9.804e-5, color='coral', linestyle='--', label='Всего подвижной нефти в керне при t=0')
plt.axhline(y=0.00017157, color='black', linestyle='--', label='Всего нефти в керне')
plt.xlabel('t, с')
plt.ylabel('v, м^3')
plt.legend()
plt.savefig("oil_out_test.png")

```

ПРИЛОЖЕНИЕ 3. ДАННЫЕ О ВЯЗКОСТИ ФАЗЫ ВОДЫ В ЗАВИСИМОСТИ ОТ КОНЦЕНТРАЦИИ ПОЛИМЕРА

| Массовая доля полимера | Вязкость фазы воды |
|---------------------------|-----------------------|
| 0,00E+00 | 0,00133250000000 |
| 1,00E-05 | 0,00165862753686 |
| 2,00E-05 | 0,00213104894727 |
| 4,00E-05 | 0,00303972312666 |
| 8,00E-05 | 0,00649200476877 |
| 1,60E-04 | 0,01386511999990 |
| 3,20E-04 | 0,02961204735040 |
| 6,40E-04 | 0,06324311281040 |
| 1,28E-03 | 0,13506973262000 |
| 2,56E-03 | 0,28847145340100 |
| 5,12E-03 | 0,61609494453800 |
| 1,02E-02 | 1,31580777304000 |