

**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский политехнический университет Петра Великого**

Институт прикладной математики и механики

Кафедра теоретическая механика

Работа допущена к защите

Зав. кафедрой, д.ф.-м.н., проф.

\_\_\_\_\_ А. М. Кривцов

" " \_\_\_\_\_

**ДИССЕРТАЦИЯ**

**на соискание академической степени**

**МАГИСТРА**

**Тема: Моделирование процессов фильтрации в нефтяных пластах**

Направление: 010403 – Механика и математическое моделирование

Выполнила студентка гр. 63604/2

Научные руководители

Э.Р. Гадельшина

И.С. Каешков

В.А. Кузькин

Санкт-Петербург

2017

## РЕФЕРАТ

39 с., 6 рис., 2 табл., 2 приложения, 25 источников

Поле давлений между скважинами, численная схема, метод конечных элементов,  
метод прогонки.

Настоящая работа посвящена моделированию поля давления в нефтяном пласте между двумя скважинами – нагнетательной и добывающей. Численно решается уравнение диффузии на равномерной сетке методом прогонки. Реализовано с помощью программного кода, написанного на C#.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	6
ОБЗОР ЛИТЕРАТУРЫ.....	8
1 УРАВНЕНИЕ ДИФФУЗИИ.....	10
1.1 ДВУМЕРНАЯ ПОСТАНОВКА.....	13
1.2 МЕТОД КОНЕЧНЫХ РАЗНОСТЕЙ.....	14
1.3 АЛГОРИТМ ЧИСЛЕННОГО РЕШЕНИЯ (МЕТОД РАСЩЕПЛЕНИЯ ПО ПРОСТРАНСТВЕННЫМ КООРДИНАТАМ).....	16
1.4 РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ МЕТОДОМ ПРОГОНКИ.....	19
1.5 СРАВНЕНИЕ ПОЛУЧЕННОГО РЕЗУЛЬТАТА С ПРИВЕДЕННЫМ В ОПУБЛИКОВАННОМ ИСТОЧНИКЕ.....	21
2 СВОЙСТВА РАЗНОСТНОЙ СХЕМЫ.....	26
2.1 СХОДИМОСТЬ.....	26
2.2 УСТОЙЧИВОСТЬ.....	27
ЗАКЛЮЧЕНИЕ.....	29
ПРИЛОЖЕНИЕ 1.....	30
ПРИЛОЖЕНИЕ 2.....	32

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

- 1) Азиз. Х., Сеттари Э. Математическое моделирование пластовых систем, Москва-Ижевск, Институт компьютерных исследований, 2004 – 416с.
- 2) Амусин Б.З., Фадеев А.Б. Метод конечных элементов при решении задач горной геомеханики, М., Недра, 1975 – 144с.
- 3) Белевич М.Ю. Математические методы решения океанологических задач, Санкт-Петербург, РГГМУ, 2008 – 191с.
- 4) Беляев Н.М., Рядно А.А. Метод нестационарной теплопроводности, М. Высшая школа. 1978 - 328с.
- 5) Дмитриевский М.В., Плиткина Ю.А. Математическое моделирование полей давлений с целью определения трещины гидроразрыва, УДК 519.63+519.65, 7 с.
- 6) Каневская Р.Д. Математическое моделирование гидродинамических процессов разработки месторождений углеводородов, Москва-Ижевск, Институт компьютерных исследований, 2002 – 140с.
- 7) Костюченко С.В. Математическое моделирование полей давлений в нефтяных резервуарах с произвольными системами скважин различных профилей // Нефтяное хозяйство. 2000. №10. С.70-77.
- 8) Кузнецов Г.В., Шеремет М.А. Разностные методы решения задач теплопроводности: учебное пособие, Томск, Изд-во ТПУ, 2007. – 172 с.
- 9) Лыков А.В. Теория теплопроводности / А.В. Лыков. – М. : Высшая школа, 1967. – 600 с.
- 10) Мартинсон Л.К., Малов Ю.И. Дифференциальные уравнения математической физики, М., МГТУ им. Н.Э. Баумана, 2002 – 368 с.
- 11) Маслов Е.А. Численное решение двумерных нестационарных уравнений теплопроводности, Томский государственный архитектурно-строительный университет, Томск, 2008, 29с.
- 12) Меркулова Н.Н., Михайлов М.Д. Разностные схемы для обыкновенных дифференциальных уравнений, Томск, ТГУ, 2004 – 122с.
- 13) Патанкар С.В. Численное решение задач теплопроводности и конвективного теплообмена при течении в каналах, М, Изд-во МЭИ, 2003. - 312 с.
- 14) Пискунов Н.С. Дифференциальное и интегральное исчисления (том II). М. Интеграл-пресс. 2002 - 410с.

- 15) Рихтмайер Р. Мортон К. Разностные методы решения краевых задач, М. Издательство Мир. 1972 – 380 с.
- 16) Сабитов К.Б. Уравнения математической физики, М. Высшая школа. 2003 – 255 с.
- 17) Самарский А.А. Введение в теорию разностных схем, М. Наука, 1971 – 553 с.
- 18) Самарский, А.А. Теория разностных схем / А.А. Самарский – М. : Наука, 1977. – 656 с.
- 19) Форсайт Дж., Вазов В. Разностные методы решения дифференциальных уравнений в частных производных, М. Иностранная литература. 1963 - 488с.
- 20) Щелкачев В.Н. Основы и приложения теории неустановившейся фильтрации: Монография: В 2 ч., М, Нефть т газ, 1995. – ч. 1. – 586 с.
- 21) Якимов. А.С. Аналитический метод решения уравнений математической физики Издательство томского университета, 2010
- 22) Яненко Н.Н. Метод дробных шагов многомерных задач математической физики, Новосибирск. Наука. 1967 - 195с
- 23) Xu J., Jiang R. et al. Non-Darcy flow numerical simulation for low-permeability reservoirs, SPE 154890, 2012 – 7с.
- 24) Yu R., Bian Y. et al. Nonlinear flow numerical simulation of low-permeability reservoir, Central South University Press and Springer - Verlag Berlin Heidelberg, 2012 – 8с.
- 25) [www.unity3d.ru/](http://www.unity3d.ru/)

## ВВЕДЕНИЕ

Энергетическое состояние нефтяной залежи, т.е. пластовое давление - является важным показателем, используемым для правильного и эффективного проектирования ее дальнейшей эксплуатации. Основным инструментом, при помощи которого оценивается энергетическое состояние нефтяных пластов, являются карты давлений, или, как их еще называют – карты изобар. Карта изобар описывает поле пластовых давлений, сложившихся на определенную дату. Из анализа таких карт получают следующую информацию:

1. снижение или увеличение пластовых давлений в отдельных зонах пласта;
2. зоны отбора нефти;
3. зоны влияния нагнетательных скважин;
4. средневзвешенные пластовые давления, рассчитываемые по зонам отбора;
5. направление движения контура нефтеносности.

В зависимости от метода, используемого для построения поля давлений, могут учитываться различные исходные данные:

- замеры забойных давлений - замеры, произведенные на забоях скважин во время их работы;
- замеры пластовых давлений - замеры, произведенные на забоях скважин через некоторое время после их остановки, необходимое для того, чтобы давление в достаточной степени стабилизировалось;
- дебиты / приемистости вертикальных, наклонных и горизонтальных скважин;
- начальное пластовое давление, заданное на контуре питания  $\Gamma$  - давление, которое имел пласт до начала его эксплуатации;
- геометрические и фильтрационные параметры трещин гидроразрыва пласта (ГРП);
- зоны отсутствия фильтрации - геометрическое место зон отсутствия коллекторов, линии непроницаемых разломов и т.д.;
- фильтрационные параметры пласта – коэффициенты проницаемости, вязкости, пьезопроводности и т.д.

В настоящее время для построения таких карт не существует полностью корректной общепризнанной методики. Все существующие подходы можно условно разделить на две группы, каждая из которых имеет свои преимущества и недостатки:

1. Интерполяция пластовых давлений.

В качестве исходной информации для подобных методов служат замеры пластовых давлений в скважинах, а также значения давлений, заданные на контуре питания. Заметим, что подобная информация не может быть получена на определенную дату одновременно по всем скважинам, т.к. для этого пришлось бы временно остановить разработку всего месторождения, поэтому замеры пластовых давлений, полученные на предыдущую дату, пересчитываются на дату построения карты. В результате этого пересчета в данные вносятся дополнительная погрешность. Для получения карты полученные исходные данные пересчитываются в узлы равномерной сетки одним из интерполяционных методов. Достоинствами методики являются простота программирования алгоритмов и использования полученных программ пользователями. К недостаткам можно отнести тот факт, что получаемая картина часто физически противоречива. Для достижения правдоподобно выглядящего результата обычно приходится прибегать к добавлению в расчет новых, искусственно создаваемых исследователем данных - как правило это контура, с заданными на них значениями пластовых давлений.

## 2. Математическое моделирование процессов фильтрации.

Основу данного подхода составляет создание математической модели, описывающей процесс фильтрации жидкости в пласте [8,9]. Обычно такая модель описывается системой дифференциальных уравнений с заданными начальными и граничными условиями. В зависимости от типа выбранной модели варьируется вид и количество уравнений. Существующие математические модели позволяют учитывать массу разнородной информации, что можно рассматривать и как преимущество и как недостаток: часто приходится адаптировать модель к исходным данным, варьируя ее параметрами. Этот процесс очень трудоемкий и занимает основное время построения фильтрационной модели.

Данная работа посвящена изучению именно этого способа получения поля давления – математическому моделированию. Для этого рассмотрим двумерную задачу, численно решим нестационарное уравнение диффузии, исследуем свойства полученной численной схемы, такие как сходимость и устойчивость.

## ОБЗОР ЛИТЕРАТУРЫ

Перед тем как приступить к моделированию процессов фильтрации потребовалось изучить значительный объем литературы, включающий в себя как учебные методички, так и объемные монографии известных исследователей.

Одним из основоположников метода конечных разностей в нашей стране был А.А. Самарский, долгие годы читавший лекции по численным методам в МГУ для студентов физического и механико-математического факультетов. На основе этих лекций им были написаны такие книги, как «Введение в теорию разностных схем» [17] и «Теория разностных схем» [18] содержащие результаты многолетних исследований.

Была рассмотрена книга известного ученого профессора Миннесотского университета (США) Патанкара С.В. [13], преподававшего численные методы решения задач о теплопереносе и течении жидкости. В своей книге автор описывает все тонкости написанной им компьютерной программы CONDUCT, разработанной для решения задач гидродинамики и теплообмена.

Так как в данной работе решается уравнение диффузии, частным случаем которого является уравнение теплопроводности были рассмотрены книги Лыкова А.В. «Теория теплопроводности» [9] и Беляева Н.М, и Рядно А.А. «Методы нестационарной теплопроводности» [4], где подробно рассматриваются решения задач нестационарной теплопроводности различных тел несколькими методами (разделения переменных, операционные, интегральные преобразования Фурье и Хенкеля).

Среди специалистов – нефтяников широкой известностью обладает книга профессоров Канадского университета Азиз Х. и Сеттари.Э. [1]. В ней изложены теоретические основы математического моделирования пластовых систем и описаны методы решения уравнений фильтрации с помощью компьютера. Даны рекомендации по конструированию математических и компьютерных моделей, их анализ и примеры программных систем.

Монография Каневской Р.Д. [6] написана на основе курса лекций, прочитанных автором в Российском Государственном Университете нефти и газа им. И. М. Губкина. В пособии рассмотрены математические модели фильтрации и их использование для моделирования разработки месторождений нефти и газа.

В книге Амусина Б.З. и Фадеева А.Б. [2] рассмотрен метод конечных элементов применительно к решению задач горной геомеханики.



В крупной монографии Щелкачева В.Н. [20] описываются уравнения движения упругой жидкости в упругой пористой среде, их решение различными методами.

Также были рассмотрены некоторые учебные пособия [3], [8], [12], в которых рассматриваются элементы теории разностных схем, разбирается метод конечных разностей для задач теплопроводности в различных постановках.

Решение дифференциальных уравнений рассматривалось в [10], [14], [19], [22], [21], подробное решение краевых задач рассматривалось в [15], [16].

Для подтверждения корректной работы выбранного метода решения, была смоделирована ситуация, рассмотренная в методичке, составленной Масловым Е.А. для выполнения лабораторных работ по курсу «Математическое моделирование плазмохимических процессов» [11]. Корректность подтверждалась путем сравнения полученных и приведенных в литературе результатов.

Визуализация решения рассматриваемой в данной работе задачи была осуществлена с помощью инструмента для разработки двух- и трехмерных приложений и игр. Была использована бесплатная версия Unity 5, свободно распространяющаяся на официальном сайте разработчика [25].

Так как после защиты магистерской диссертации планируется продолжить работу над моделированием процессов фильтрации и пластовых систем, были рассмотрены статьи о моделировании фильтрации в низкопроницаемых коллекторах [23], [24] и литература по переориентации трещины повторного гидроразрыва пласта [5], [7].

## 1 УРАВНЕНИЕ ДИФФУЗИИ

Уравнение диффузии представляет собой частный вид дифференциального уравнения в частных производных. Бывает нестационарным и стационарным.

Нестационарное уравнение диффузии классифицируется как параболическое дифференциальное уравнение. Оно описывает распространение растворимого вещества вследствие диффузии или перераспределение температуры тела в результате теплопроводности. Для нахождения единственного решения уравнение рассматривается в совокупности с начальными и краевыми условиями. Для аналитического решения параболических уравнений в бесконечной области (задача Коши для параболического уравнения) используют специальную интегральную формулу. Для аналитического решения параболических уравнений в конечной области может применяться метод разделения переменных Фурье. Для численного решения параболических уравнений используют метод конечных элементов, метод конечных разностей и метод конечных объемов, а так же их комбинации и другие численные методы, подходящие под решаемую задачу.

В случае, когда ставится задача по нахождению установившегося распределения плотности или температуры (например, в случае, когда распределение источников не зависит от времени), из нестационарного уравнения выбрасывают члены уравнения, связанные со временем. Тогда получается стационарное уравнение теплопроводности, относящееся к классу эллиптических уравнений. Поскольку эллиптические уравнения не зависят от времени, то для них задаются, только краевые условия. Для аналитического решения применяют метод разделения переменных Фурье, метод функции Грина и метод потенциалов. [12]

В смысле интерпретации при решении уравнения диффузии речь идет о нахождении зависимости концентрации вещества (или иных объектов) от пространственных координат и времени, причем задан коэффициент (в общем случае также зависящий от пространственных координат и времени), характеризующий проницаемость среды для диффузии. При решении уравнения речь идет о нахождении зависимости искомой величины от пространственных координат и времени. Общий вид уравнения в декартовых координатах:

$$\lambda \frac{\partial q}{\partial t} = \frac{\partial}{\partial x} \left( \Gamma \frac{\partial q}{\partial x} \right) + \frac{\partial}{\partial y} \left( \Gamma \frac{\partial q}{\partial y} \right) + \frac{\partial}{\partial z} \left( \Gamma \frac{\partial q}{\partial z} \right) + S \quad (1)$$

Где  $\Gamma$  - обобщенный коэффициент диффузии;  $q$  – искомая переменная;  $\lambda$  - величина, характеризующая материал;  $S$  – мощность внутренних источников в единице объема

Для упрощения задачи принимаем допущение, что  $\lambda$  и  $\Gamma$  постоянны и не зависят от переменной, координаты и времени.

Данное уравнение (1) описывает множество различных ситуаций, для того чтобы описать один конкретный случай, необходимо определить геометрические, физические и граничные условия.

Геометрические условия определяют форму и размеры тела, в котором протекает изучаемый процесс. Физические условия определяют характеристики материала ( $\lambda$ ). Начальные условия описывают поле значений искомой величины ( $q$ ) в начальный момент времени.

$$t=0, q=f(x, y, z) \text{ или } q=q_0=const \quad (2)$$

Граничные условия (ГУ) определяют особенности протекания процесса на поверхности рассматриваемого объекта (области) и могут быть заданы несколькими способами:

1. ГУ первого рода – задается распределение температуры на границе рассматриваемого объекта для каждого момента времени

$$q=q_w(x, y, z, t) \text{ или } q=q_w=const \quad (3)$$

Где  $q_w$  – значение переменной на границе области.

2. ГУ второго рода – задается значение диффузионного потока для каждой точки границы объекта в любой момент времени:

$$-\left(\Gamma \frac{\partial q}{\partial \hat{n}}\right)_w = q_w(x, y, z, t), \text{ часто } q_w=const \quad (4)$$

Где  $\hat{n}$  – нормаль к границе

3. ГУ третьего рода – задается взаимосвязь между диффузионным потоком от границы рассматриваемого объекта и диффузионным потоком из окружающей среды.

$$-\left(\Gamma \frac{\partial q}{\partial \dot{n}}\right)_w = \alpha(q_w - q^e) \quad (5)$$

Где  $q^e$  – характеризует окружающую среду.

4. ГУ четвертого рода – для определения диффузионного взаимодействия между элементами, имеющими различные физические характеристики, задаем равенство потоков и равенство переменных по обе стороны от границы раздела:

$$\begin{cases} -\left(\Gamma_1 \frac{\partial q_1}{\partial \dot{n}}\right)_b = -\left(\Gamma_2 \frac{\partial q_2}{\partial \dot{n}}\right)_b \\ q_1(x_b, y_b, z_b, t) = q_2(x_b, y_b, z_b, t) \end{cases} \quad (6)$$

Где  $x_b, y_b, z_b$  - координаты границы раздела сред;  $q_1, q_2$  – характеризуют соприкасающиеся среды. Это условие применяется, например, при решении задач теплопроводности для многослойных пластин.

Дифференциальное уравнение (1) вместе с условиями однозначности дает полную математическую формулировку краевой задачи диффузии.

При решении конкретных задач нестационарной диффузии можно, применяя методы математического моделирования, можно добиться существенного упрощения общей математической постановки. Если для рассматриваемого процесса:

$$\frac{\partial^2 P}{\partial x^2} \gg \frac{\partial^2 P}{\partial y^2} \gg \frac{\partial^2 P}{\partial z^2} \quad (7)$$

то можно ограничиться одномерным нестационарным уравнением диффузии, когда искомая переменная зависит только от одной координаты и времени. Все прочие коэффициенты, отвечающие за характеристики пласта и флюида, считаем независимыми от координат и времени. В этом случае физическая картина проста и математические вычисления минимальны. Однако при решении производственных задач сложно представить

себе ситуацию, когда одномерной задачи было бы достаточно, и она бы корректно описывала все происходящие процессы.

Тем не менее, существует много физических явлений, которые могут быть удовлетворительно представлены в двух измерениях, и обычно качественные особенности многих практических задач могут быть изучены таким образом. Если считать пласт достаточно тонким, чтобы можно было пренебречь воздействием гравитационных сил на жидкость, находящуюся в пласте, то течение можно считать двумерным. Добавление третьего измерения в программу делает ее более сложной в использовании из-за того, что задача усложняется и требует гораздо больше времени и памяти для вычисления. [5]

Аналитическим решением уравнения (1) является выражение для  $q$ , зависящего от  $x$ . Численное решение находится в виде численных значений  $q$  в конечном числе расчетных точек. Дискретные значения переменной  $q$  в каждой расчетной точке  $i$  находятся из системы алгебраических уравнений, которые называются дискретными аналогами дифференциального уравнения (1).

При увеличении числа расчетных точек численное решение становится более корректным и приближается к точному. Для одномерной задачи уравнение обычно можно решить аналитически, но для сложных многомерных задач очень трудно или вообще невозможно получить аналитическое решение, в таком случае альтернативой является численный метод. Преимущество численного метода заключается в замещении дифференциального уравнения системой алгебраических уравнений, которую можно решить с помощью компьютера.[13]

Исходя из выше перечисленного, рассмотрим решение двумерного уравнения диффузии методом конечных разностей.

Для того чтобы обобщенное уравнение (1) описывало процесс фильтрации, нужно осуществить следующую замену:

$$q = P; \lambda = \phi c; \Gamma = k / \mu \quad (8)$$

Где  $P$  – пластовое давление;  $\phi$  – пористость пласта;  $c$  – сжимаемость породы;  $k$  – проницаемость пласта;  $\mu$  – вязкость флюида.

## 1.1 Двумерная постановка

Перейдем к рассмотрению двумерной задачи, теперь искомая переменная  $P$  зависит от координаты  $x$  и координаты  $y$ . Нестационарное распространение импульса описывается следующим уравнением, записанным в декартовой системе координат:

$$c \frac{\partial P(x, y, t)}{\partial t} = \frac{k}{\mu} \left( \frac{\partial^2 P(x, y, t)}{\partial x^2} + \frac{\partial^2 P(x, y, t)}{\partial y^2} \right) \quad (9)$$

Где  $P$  – пластовое давление;  $\phi$  – пористость пласта;  $c$  – сжимаемость породы (1/кгс/см<sup>2</sup>);  $k$  – проницаемость пласта (мД);  $\mu$  – вязкость флюида (сП);  $x, y$  – координаты (м);  $t$  – координата времени (сут).

Это уравнение устанавливает связь между временным и пространственным изменением давления в любой точке рассматриваемого пространства.

## 1.2 Метод конечных разностей

Сформулированное уравнение (9) с соответствующими краевыми условиями (начальными и граничными) будем решать численно. Численным решением называется решение, полученное в виде таблицы чисел.

При решении дифференциального уравнения в частных производных наиболее часто используется метод конечных разностей (МКР). Идея МКР решения краевых задач весьма проста и видна уже из самого названия: вместо производных в дифференциальном уравнении используются их конечно-разностные аппроксимации. При построении дискретных аппроксимаций краевых дифференциальных задач нужно стремиться увязать две, возможно, противоречивые цели: хорошее качество аппроксимации и эффективное устойчивое решение получающихся при этом алгебраических систем.

При использовании МКР для задач теплопроводности твердое тело представляют в виде совокупности узлов. Аппроксимируя частные производные дифференциального уравнения (1) конечными разностями, получаем систему линейных алгебраических уравнений для определения давления как локальной характеристики в каждом узле сетки. Полученная система является незамкнутой, для ее замыкания используют разностное представление граничных условий. В результате получают замкнутую систему линейных алгебраических уравнений, которую решают численными методами [8].

В данной работе анализируется задача распространения давления в нефтяном пласте между скважинами. Рассматривается симметричный элемент разработки прямоугольной

формы (размерами  $L_x \times L_y$ ), в противоположных углах которого находятся скважины – нагнетательная и добывающая. Скважины моделируются линейным источником ограниченной длины ( $l_{inj}$ ,  $l_{prod}$ ) т.к. предполагается, что на добывающей был проведен гидроразрыв пласта (ГРП), а на нагнетательной образовалась трещина автоГРП. Схема рассматриваемого элемента представлена на рис. 1

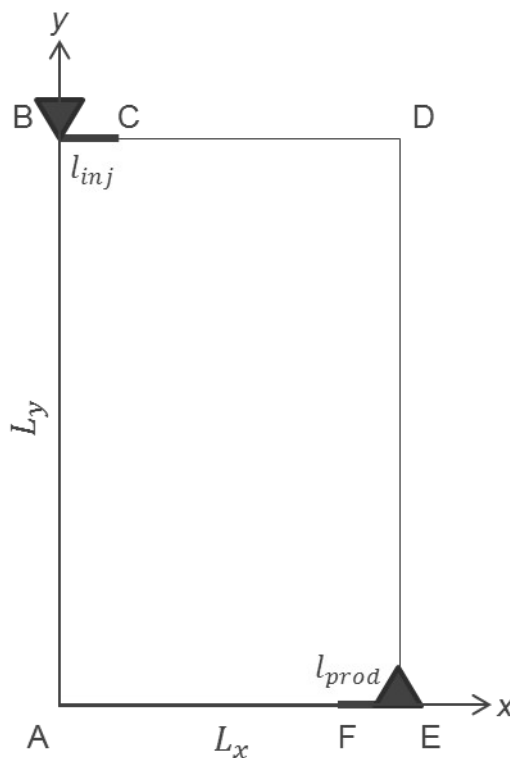


Рис.1 Схема рассматриваемого элемента

При постановке задачи были приняты следующие допущения:

1. Пласт и поток однородны, все физические характеристики одинаковы в любой точке элемента;
2. Давление внутри трещины одинаково и равно скважинному.

Конкретизируем нашу задачу задав геометрические ( $L_x$ ,  $L_y$ ), физические ( $\phi$ ,  $c$ ,  $k$ ,  $\mu$ ), начальные ( $P_{res}$  при  $t=0$ ) условия:

- $L_x = 300\text{м}$  – длина рассматриваемой области по оси  $x$ ;
- $L_y = 500\text{м}$  – длина рассматриваемой области по оси  $y$ ;
- $l_{inj} = 50\text{м}$  – полудлина трещина автоГРП на нагнетательной скважине;
- $l_{prod} = 50\text{м}$  – полудлина трещины ГРП на добывающей скважине;
- $\phi = 0.15$  – пористость моделируемого пласта;

- $c = 1.7 \cdot 10^{-4}$  ш/кгс/см<sup>2</sup> – сжимаемость моделируемого пласта;
- $k = 10$  мД – проницаемость моделируемого пласта;
- $\mu = 0.8$  сП – вязкость моделируемой жидкости, рассматриваем однофазное течение;
- $P_{res}$  (при  $t = 0$ ) = 250 атм – пластовое давление в начальный момент времени, одинаково для всех узлов.

Теперь необходимо задать граничные условия. В месте нахождения скважин с трещинами (BC и FE на рис.1) будет происходить обмен импульсами, распространение давления в пласт:

$$P_{inj} \left( \frac{\partial P(x, y, t)}{\partial x} \right) = \frac{k}{\mu} \left( \frac{\partial P(x, y, t)}{\partial x} \right) = \frac{\sum_i P_i}{P_i} (i - P(x, y, t)) \quad (10)$$

$$P_{inj} \left( \frac{\partial P(x, y, t)}{\partial y} \right) = \frac{k}{\mu} \left( \frac{\partial P(x, y, t)}{\partial y} \right) = \frac{\sum_i P_i}{P_i} (i - P(x, y, t))$$

Где

По всем остальным границам (AB, CD, DE, FA на рис.1) задаем условие неперетока и симметрии, т.е. градиент давления равняется нулю.

$$\frac{k}{\mu} \left( \frac{\partial P(x, y, t)}{\partial x} \right) = 0 \quad (11)$$

$$\frac{k}{\mu} \left( \frac{\partial P(x, y, t)}{\partial y} \right) = 0$$

### 1.3 Алгоритм численного решения (метод расщепления по пространственным координатам)



Для численного решения задачи воспользуемся методом Писмена – Рекфорда (метод расщепления по пространственным координатам) [8, 17]. Для аппроксимации дифференциального уравнения (9) разностным методом введем пространственно-временную сетку с координатами  $x_i = i \cdot h_x$ ;  $y_j = j \cdot h_y$ ,  $t_k = k \cdot \tau$ , где  $h_x$ ,  $h_y$  – шаги по пространству,  $\tau$  – шаг по времени;  $i = 0, \dots, N_x$ ,  $j = 0, \dots, N_y$ , и  $k = 0, \dots, K$ . Таким образом, вся расчетная область покрывается сеткой (рис. 2) [8].

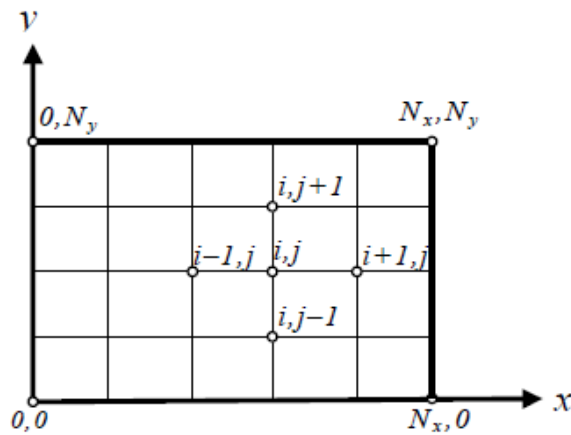


Рис.2 Разностная сетка области решения

Введем следующее обозначение:

$$P(x_i, y_j, t_k) = P_{i,j}^k$$

Дискретизацию уравнения (2) будем проводить на основе локально-одномерной схемы А.А. Самарского, которая является абсолютно устойчивой и обладает свойством суммарной аппроксимации. Суть этого подхода состоит в том, что шаг по времени реализуется в два этапа – на промежуточном (полушаге  $\tau/2$ ) временном шаге проводим дискретизацию двумерного уравнения (2) только в направлении оси  $x$  и получаем одномерное уравнение, после его решения проводим вновь дискретизацию уравнения (2), но уже в направлении оси  $y$  и, решая полученное одномерное уравнение, определяем поле температуры на целом шаге по времени. Представим (2) в разностном виде, используя неявную схему на каждом полушаге по времени [3]:

$$\varphi c \frac{P_{i,j}^{k+1/2} - P_{i,j}^k}{\tau/2} = \frac{k}{\mu} \left( \frac{P_{i+1,j}^{k+1/2} - 2P_{i,j}^{k+1/2} + P_{i-1,j}^{k+1/2}}{h_x^2} \right) \quad (12)$$

$$\varphi c \frac{P_{i,j}^{k+1} - P_{i,j}^{k+1/2}}{\tau/2} = \frac{k}{\mu} \left( \frac{P_{i,j+1}^{k+1} - 2P_{i,j}^{k+1} + P_{i,j-1}^{k+1}}{h_y^2} \right) \quad (13)$$

Аппроксимируем граничные условия, записанные в частных производных, соответствующими конечными разностями:

- Для границ BC и FE

$$\begin{aligned}
 & j = N_y, 0 < i < N_{inj}; \\
 & \prod_{j=0, N_x - N_i}^i i < i < N_x \\
 & \prod_{i=0}^i P_i \\
 & P_{inj} (i - P(x, y, t)) \\
 & \frac{k}{\mu} \left( \frac{P_{i,j}^k - P_{i-1,j}^k}{h_x} \right) = i
 \end{aligned}$$

$$\begin{aligned}
 & \prod_{i=0}^i P_i \\
 & P_{inj} (i - P(x, y, t)) \\
 & \frac{k}{\mu} \left( \frac{P_{i,j}^k - P_{i-1,j}^k}{h_y} \right) = i
 \end{aligned}$$

- Для границ AB, CD, DE, FA

$$\begin{aligned}
 & x = 0, 0 < y < L_y; \\
 & y = L_y, l_{inj} < x < L_x \\
 & \frac{k}{\mu} \left( \frac{P_{i,j}^k - P_{i-1,j}^k}{h_x} \right) = 0
 \end{aligned}$$

$$\begin{aligned}
 & x = L_x, 0 < y < L_y \\
 & \frac{k}{\mu} \left( \frac{P_{i,j}^k - P_{i-1,j}^k}{h_y} \right) = 0
 \end{aligned}$$

$$\begin{aligned}
 & \prod_{i=0}^i P_i \\
 & y = 0, 0 < x < L_x - l_i
 \end{aligned}$$

Выбранную схему аппроксимации частных производных можно графически представить следующим образом:

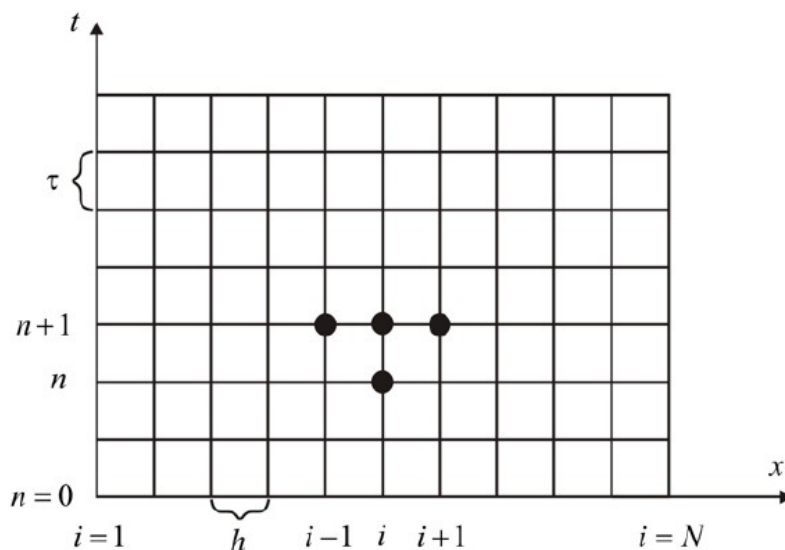


Рис.3 Шаблон неявной четырехточечной разностной системы.

Рис.3 наглядно демонстрирует, что используется четырехточечная разностная схема – три точки берутся на новом временном слое и одна со старого временного слоя.

Сформулированный выше способ аппроксимации производных называется неявным потому, что поле температуры на новом временном слое представлено неявно, т.е. для его определения необходимо решать систему уравнений.

Разностные уравнения (12) и (13) сводятся к стандартному трехдиагональному виду и решаются последовательно методом прогонки. Сначала для всей области решается уравнение (12), после того, как его решение будет найдено, переходят к решению уравнения (13)

#### 1.4 Решение систем линейных алгебраических уравнений методом прогонки

Рассмотрим решение уравнения (12) методом прогонки. Приведем это уравнение к виду [18]:

$$a_i P_{i,j}^{K+1/2} = b_i P_{i+1,j}^{k+1/2} + c_i P_{i+1,j}^{k+1/2} + d_i \quad (14)$$

Преобразуем уравнение (12)

$$\varphi c \frac{P_{i,j}^{k+1/2}}{\tau/2} + \frac{k}{\mu} \frac{2 P_{i,j}^{k+1/2}}{h_x^2} = \varphi c \frac{P_{i,j}^k}{\tau/2} + \frac{k}{\mu} \frac{P_{i+1,j}^{k+1}}{h_x^2} + \frac{k}{\mu} \frac{P_{i-1,j}^{k+1/2}}{h_x^2}$$

Затем

$$\left( \frac{\varphi c}{\tau/2} + \frac{2k}{\mu h_x^2} \right) P_{i,j}^{k+1/2} = \left( \frac{k}{\mu h_x^2} \right) P_{i+1,j}^{k+1/2} + \left( \frac{k}{\mu h_x^2} \right) P_{i-1,j}^{k+1/2} + \frac{\varphi c}{\tau/2} P_{i,j}^k$$

Тогда

$$a_i = \frac{\varphi c}{\tau/2} + \frac{2k}{\mu h_x^2}$$

$$b_i = c_i = \frac{k}{\mu h_x^2}$$

$$d_i = \frac{\varphi c}{\tau/2} P_{i,j}^k \quad (15)$$

Для границ АВ, СН, DE, GF для точек от 0 до Nx запишем выражение:

$$a_0 P_{0,j}^{k+1/2} = b_0 P_{i,j}^{k+1/2} + d_0$$

$$c_0 = 0 \quad \text{т.к. граничная точка имеет только одну соседнюю}$$

Для реализации граничных условий на соответствующих границах мы должны положить коэффициенты  $a_0, b_0, d_0, a_N, c_N, d_N$  следующими:

$$a_0 = a_N = 1; b_0 = b_N = 1; d = d_N = 0 \quad (16)$$

Алгоритм прогонки начинается с записи уравнения в виде:

$$P_{0,j}^{k+1/2} = A_0 P_{1,j}^{k+1/2} + B_0 \quad (17)$$

Где

$$A_0 = \frac{b_0}{a_0}; B_0 = \frac{d_0}{a_0} \quad (18)$$

Соотношение (17) подставляется в (14) для  $i=0$ . В результате получается, что  $P_{0,j}^{k+1/2}$  выражается через  $P_{1,j}^{k+1/2}$ . Продолжая процесс последовательной подстановки (или прямой прогонки), можно выразить  $P_{i,j}^{k+1/2}$  через  $P_{i+1,j}^{k+1/2}$ :

$$P_{i,j}^{k+1/2} = A_i P_{i+1,j}^{k+1/2} + B_i \quad (19)$$

Где  $A$  и  $B$  – новые коэффициенты, появившиеся в процессе подстановки.

Представим, что мы находимся на стадии процесса подстановок, когда только что выразили  $P_{i-1,j}^{k+1/2}$  в виде

$$P_{i-1,j}^{k+1/2} = A_{i-1} P_{i,j}^{k+1/2} + B_{i-1} \quad (20)$$

Если подставить (20) в (14), то получается следующее выражение:

$$a_i P_{i,j}^{k+1/2} = b_i P_{i+1,j}^{k+1/2} + c_i (A_{i-1} P_{i,j}^{k+1/2} + B_{i-1}) P_{i+1,j}^{k+1/2} + d_i$$

которое может быть переписано в форме (19). Таким образом, можно получить формулы для  $A$  и  $B$ :

$$A_i = \frac{b_i}{a_i - c_i A_{i-1}} \quad (21)$$

$$B_i = \frac{d_i - c_i B_{i-1}}{a_i - c_i A_{i-1}} \quad (22)$$

Заметим, что знаменатели в выражениях (21) и (22) одинаковые.

Выражения (21) и (22) рекурсивные, т.е.  $A_i$  и  $B_i$  зависят от значений  $A_{i-1}$  и  $B_{i-1}$ . Такой рекурсивный процесс нуждается в отправной точке. Она обеспечивается выражениями (18), которые не рекурсивны [18].

Перейдя к вычислению  $A_N$  и  $B_N$  можно обнаружить, что, как  $b_N=0$ ,  $A_N$  тоже будет равно нулю ((16) и (21)). В результате согласно (15)  $P_N$  будет равно  $B_N$ . Рассчитав таким образом значение  $P_N$ , можно начать процесс обратной прогонки с использованием формулы (20) для получения значений переменной  $P$  в каждом узле.

Итоговый алгоритм решения методом прогонки может быть разбит на следующие шаги:

1. Вычисляем  $A_0$  и  $B_0$  по выражениям (18)
2. Получаем  $A_i$  и  $B_i$  для  $i$  от 1 до  $N$ , используя рекурсивные выражения (21) и (22)
3. Полагаем  $P_N=B_N$
4. Подставляем найденные значения величин в формулу (20) для  $i$  от  $N-1$  до 0 и определяем  $P$  в узлах от  $N-1$  до 0.

Алгоритм решения уравнения (13) аналогичен.:

#### 1.5 Сравнение полученного результата с приведенным в опубликованном источнике.

Чтобы убедиться в корректности составленной численной схемы и написанного кода, сравним полученный результат с опубликованным в [11]. Для этого зададим начальные и граничные условия соответственно с приведенными в методических указаниях Маслова Е.А. [11].

В качестве примера применения метода конечных разностей рассмотрим задачу на основе двумерного нестационарного уравнения теплопроводности. Анализируется задача теплообмена между высокотемпературной струей и пластиной из конструкционного материала, внешняя поверхность которого подвергается воздействию высокотемпературной

высокоскоростной струи с заданными параметрами. Схема взаимодействия высокотемпературной струи с преградой представлена на рис. 4.

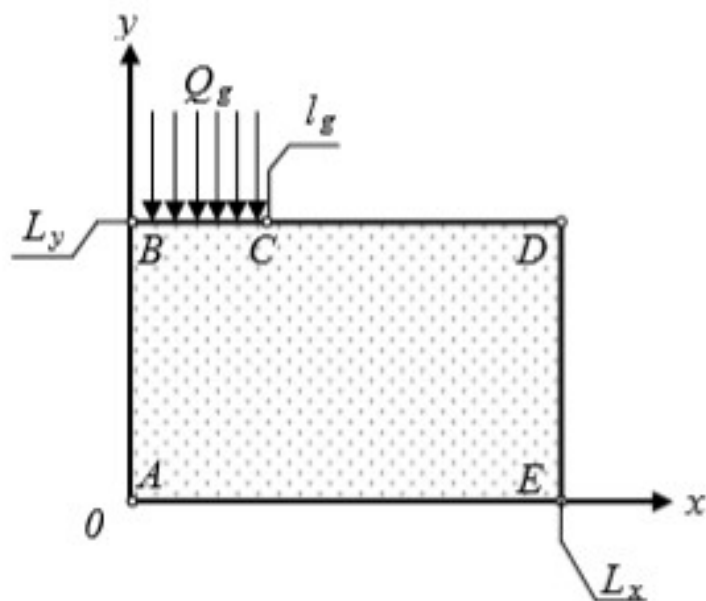


Рис. 4 Область решения задачи:  $x, y$  – декартовы координаты;  $L_x$  – ширина пластины;  $L_y$  – толщина пластины;  $l_g$  – протяженность области воздействия струи;  $Q_g$  – высокотемпературный поток.

При постановке задачи были приняты следующие допущения:

1. Вклад радиационной составляющей в теплообмен на внешней поверхности не учитывается.
2. Возможные процессы плавления и окисления материала преграды активными компонентами газового потока не рассматриваются.
3. Влияние конденсированной фазы в струе на теплообмен учитывается через коэффициент теплообмена  $\alpha_g$  на поверхности.
4. Теплофизические характеристики ( $\lambda, \rho, c$ ) материала пластины постоянны.

Математическая модель, описывающая в рамках сформулированной задачи процесс прогрева пластины, включает нестационарное двумерное уравнение теплопроводности с соответствующими начальными и граничными условиями:

$$c \frac{\partial T(x, y, t)}{\partial t} = \left( \frac{\partial^2 T(x, y, t)}{\partial x^2} + \frac{\partial^2 T(x, y, t)}{\partial y^2} \right) \quad (23)$$

Где  $\rho$  – плотность материала пластины,  $c$  – коэффициент удельной теплоемкости,  
 $\lambda$  – коэффициент теплопроводности,  $T$  – температура пластины

Начальные условия:

$$T(x, y) = T_0 \text{ const} \quad (24)$$

Граничные условия:

- Условие симметрии на оси (AB):

$$x=0, 0 < y < L_y: \left( \frac{\partial T(x, y, t)}{\partial x} \right) = 0 \quad (25)$$

- Условие кондуктивно – конвективного теплообмена газового потока с поверхностью материала пластины (BC):

$$0 \leq x \leq l_g, 0 = L_y: \left( \frac{\partial T(x, y, t)}{\partial y} \right) = \alpha_g (T_g - T(x, z, t)) \quad (26)$$

- Условие кондуктивно-конвективного теплообмена с воздухом на нагреваемой поверхности (CD):

$$l_g \leq x \leq L_x, y = L_y: \left( \frac{\partial T(x, y, t)}{\partial y} \right) = \alpha_e (T_e - T(x, z, t)) \quad (27)$$

- Условие кондуктивно-конвективного теплообмена с воздухом на боковой поверхности (DE):

$$x = L_x, 0 < y < L_y: \left( \frac{\partial T(x, y, t)}{\partial x} \right) = \alpha_e (T_e - T(x, z, t)) \quad (28)$$



- Условие кондуктивно-конвективного теплообмена с воздухом на тыльной стороне пластины (AE):

$$0 \leq x \leq L_x, y=0: -\left(\frac{\partial T(x, y, t)}{\partial x}\right) = \alpha_e (T_e - T(x, z, t)) \quad (28)$$

Где  $T$  – температура пластины,  $T_e$  – температура окружающей среды,  $T_g$  – температура газового потока,  $t$  – время.

Определим константы:

$L_x = 0.1$ ,  $L_y = 0.15$ ,  $l_g = 0.05$ ,

$N_x = 100$ ,  $N_y = 100$ ,  $K = 100$ ,  $\text{Time} = 25$ ,

$\text{Ros} = 1800$ ,  $\text{Cps} = 840$ ,  $\text{Lams} = 1.2$ ,  $\text{Alfae} = 30$ ,  $\text{Alfag} = 2500$ ,

$T_0 = 300$ ,  $T_g = 3000$ ,  $T_e = 300$ .

После реализации данных условий в программном коде сравним полученные результаты:

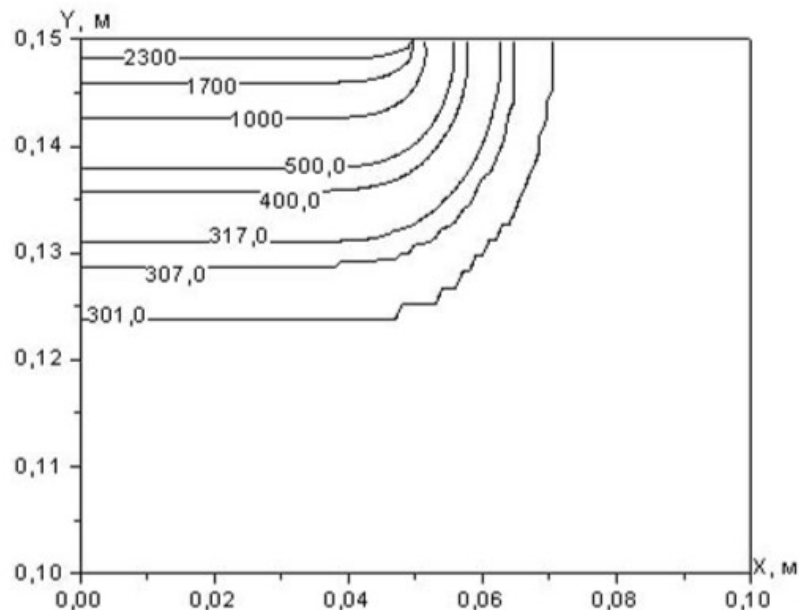


Рис5 Результат, привозящийся в методических указаниях Маслова.

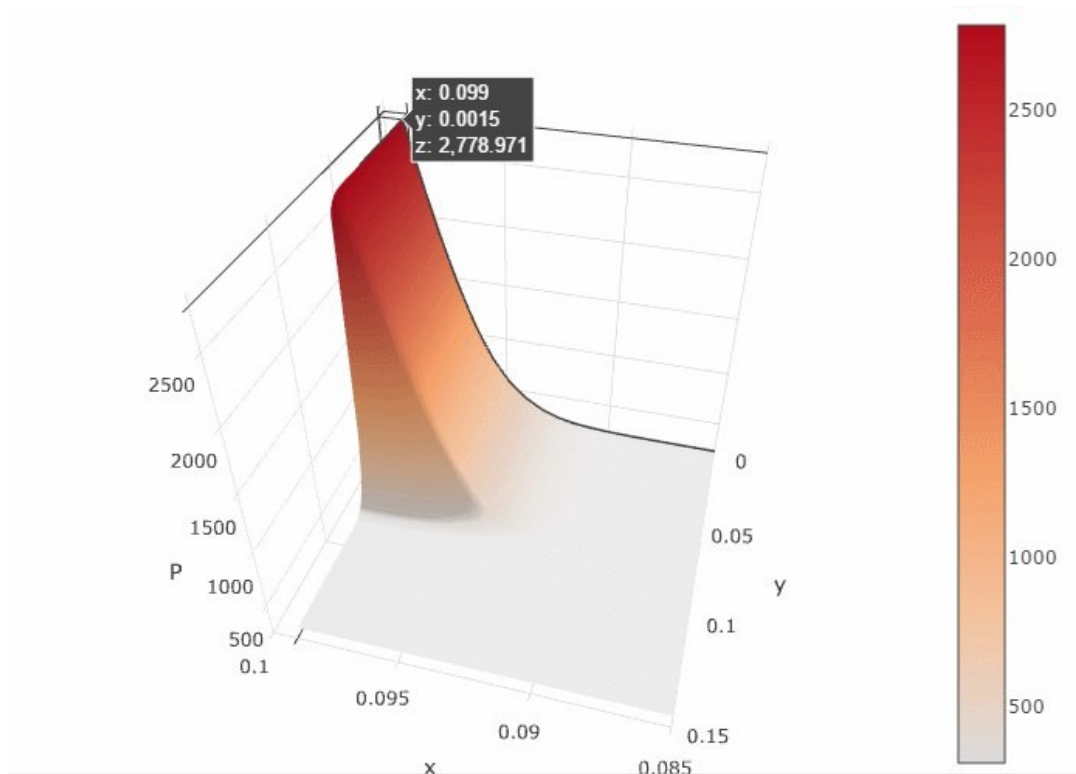


Рис 6 Результат, полученный автором.

Как видно (в презентации видно будет лучше) результаты идентичны, а значит алгоритм работает корректно, можно смело модифицировать граничные условия под необходимую задачу.

## 2 СВОЙСТВА РАЗНОСТНОЙ СХЕМЫ

Составляя алгоритм, следует учитывать его устойчивость, порядок аппроксимации, точность и эффективность. Почти всегда неявные схемы устойчивее явных. Однако, даже на примере уравнения диффузии видно, что, если бы оно было нелинейным (коэффициент диффузии или источник импульса являлся бы функцией), то реализация неявной схемы усложнилось. В этом случае на каждом слое по времени пришлось бы решать систему уже нелинейных уравнений, что сразу многократно усложнило бы задачу.

### 2.1 Сходимость

Исходную краевую задачу

$$LF = g$$

Заменим ее аналогом – разностной задачей:

$$\square_m f = g$$

Решив которую получим вектор  $f$ . Очевидно, что непосредственно сравнивать  $F$  и  $f$  нельзя. Они являются элементами различных векторных пространств (бесконечномерного и конечномерного), так что, оценить норму их разности невозможно.

Для получения требуемых оценок надо иметь два элемента одного и того же векторного пространства. Так, можно с помощью какой-либо интерполяционной процедуры построить из конечного вектора  $f$  функцию непрерывного аргумента – приближенное решение  $\hat{F}$ . Тогда, близость приближения  $\hat{F}$  к точному решению  $f$  может быть оценена величиной

$$\|s\| = \|f - \hat{f}\|$$

Где  $s$  – функция, называемая погрешностью решения.

Если при измельчении сетки величина  $\|s\| \rightarrow 0$ , то говорят, что имеет место сходимость приближенных решений.

Исследуем сходимость рассматриваемой в данной работе численной схемы, найдем решения при Time = 30 и K=100 ( $\tau = \text{Time}/K$ ,  $\tau$  - шаг по времени) на различных сетках.

Таблица 1 Исследование численной схемы на сходимость

Число узлов	P	Сравнение с 300 сеткой, %	Сравнение с соседней сеткой, %
30	335.27	3.76	
50	340.66	2.21	1.58
90	344.83	1.01	1.21
100	345.20	0.91	0.11
150	346.84	0.44	0.47
180	347.35	0.29	0.15
200	347.58	0.22	0.07
250	348.04	0.09	0.13
300	348.36	0.00	0.09

Как видно из Таблицы 1 отклонение решений на соседних сетках достаточно мало, а значит решение сходится.

## 2.2 Устойчивость

Устойчивость разностной схемы означает, что малые возмущения в начальных данных и правой части разностной схемы приводят к равномерно малому по  $\tau$  изменению решения. Свойство устойчивости гарантирует близость при достаточно малых  $\tau$  между точным решением разностной схемы его практической реализацией.

Исследуем сходимость рассматриваемой в данной работе численной схемы, найдем решения на сетке со 100 узлами при Time = 30 при различных шагах по времени

Таблица 1 Исследование численной схемы на устойчивость

k	P	Сравнение с k=600, %	Сравнение соседних, %
10	346.07	0.00	
50	345.24	0.24	0.24
100	345.2	0.25	0.01
150	345.24	0.24	0.01
300	345.39	0.20	0.04
500	345.5	0.16	0.03
600	345.55	0.15	0.01
800	345.62	0.13	0.02

Как видно из Таблицы 2 отклонение решений с разным шагом по времени менее 1%.  
Схему можно считать устойчивой

## ЗАКЛЮЧЕНИЕ

В ходе работы была изучена литература по решению численных схем и моделированию процессов фильтрации., освоен и реализован метод прогонки. Так же автор познакомился с такими языками программирования, как Fortran и C#. Было получено поле давлений между двумя скважинами, нагнетательной и добывающей, путем численного решения уравнения диффузии в частных производных на равномерной сетке методом прогонки.

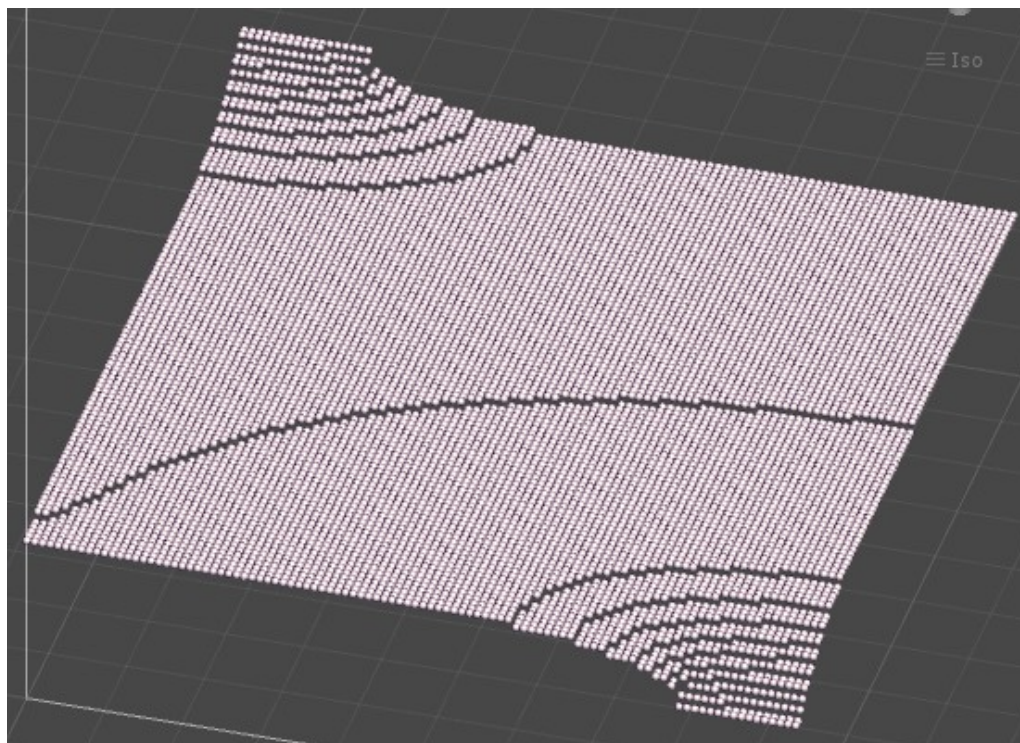
Далее планируется изучить вопрос о переориентации полей давлений вследствие повторного гидроразрыва низкопроницаемого пласта, в котором распространяется нелинейный закон фильтрации Дарси.

# ПРИЛОЖЕНИЕ 1

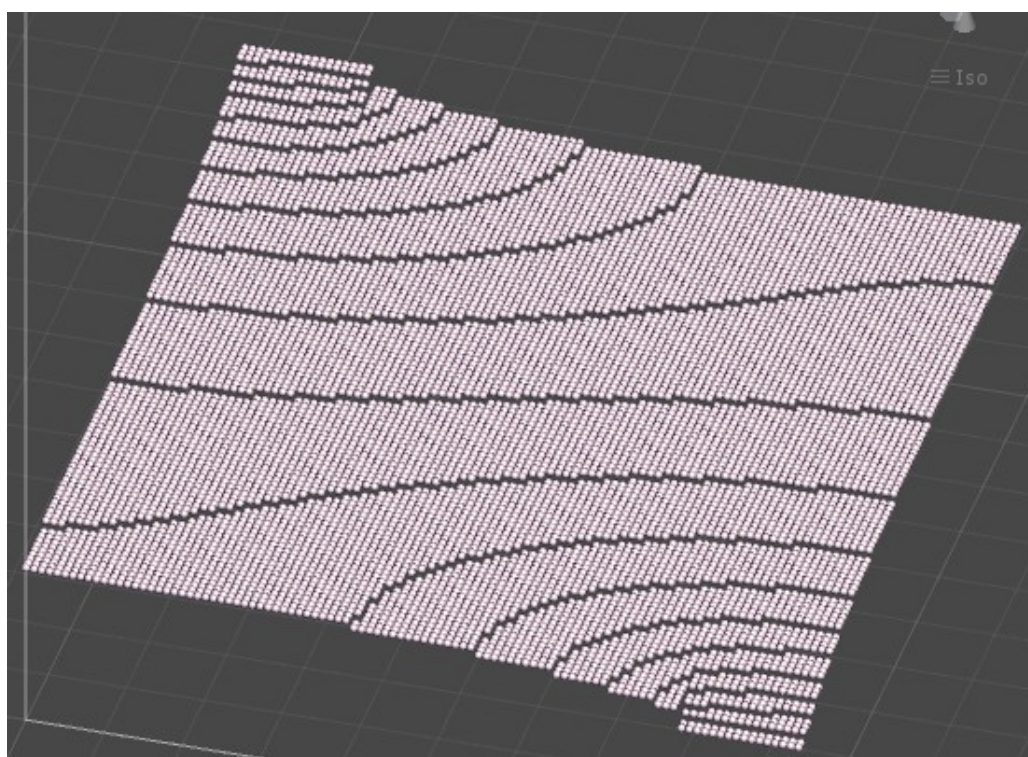
Примеры расчета

Сетка 100 К=100

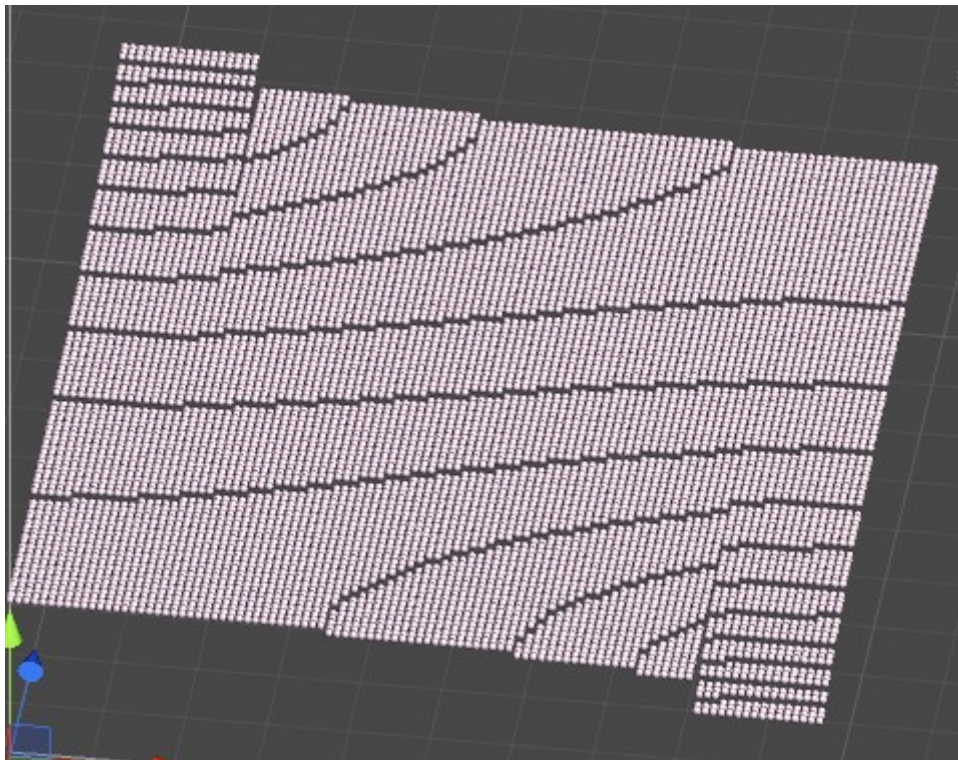
Сутки



10 дней



100 дней





## ПРИЛОЖЕНИЕ 2

### Код программы на C#

```
using System;
using System.IO;

namespace ConductivityEquation.Console
{
    /// <summary>
    /// Типы рядов в матрице.
    /// </summary>
    internal enum RowType
    {
        /// <summary>
        /// Строка.
        /// </summary>
        Line,

        /// <summary>
        /// Столбец.
        /// </summary>
        Column,
    }

    /// <summary>
    /// Класс расширений для работы с матрицами и массивами.
    /// </summary>
    internal static class Extensions
    {
        /// <summary>
        /// Преобразование строки/столбца матрицы в одномерный массив
        данных.
        /// </summary>
        /// <param name="source">Матрица с данными.</param>
        /// <param name="number">Номер столбца/строки.</param>
        /// <param name="length">Длина столбца/строки.</param>
        /// <param name="rowType">Признак типа колонки (строка или
        столбец).</param>
        /// <returns>Одномерный массив данных</returns>
        public static float[] ToArray(
            this float[,] source,
            int number,
            int length,
            RowType rowType)
        {
            var result = new float[length];
            var getCell = rowType == RowType.Line
                ? (i => source[i, number])
                : (Func<int, float>)(i => source[number, i]);

            for (var i = 0; i < length; i++)
            {
                result[i] = getCell(i);
            }

            return result;
        }

        /// <summary>
        /// Заполнение строки/столбца матрицы данными из одномерного массива
        той же длины.
    }
}
```

```

    /// </summary>
    /// <param name="matrix">Матрица с данными.</param>
    /// <param name="data">Массив данных.</param>
    /// <param name="number">Номер строки/столбца.</param>
    /// <param name="length">Длина строки/столбца.</param>
    /// <param name="rowType">Указание на столбец/строку.</param>
    public static void FillRow(this float[,] matrix,
        float[] data,
        int number,
        int length,
        RowType rowType)
    {
        var setCell = rowType == RowType.Line
            ? i => matrix[i, number] = data[i]
            : (Action<int>)(i => matrix[number, i] = data[i]);

        for (var i = 0; i < length; i++)
        {
            setCell(i);
        }
    }
}

/// <summary>
/// Класс для построения матрицы распределения давления.
/// </summary>
internal class MatrixBuilder
{
    #region Constants

    /// <summary>
    /// Плотность вещества.
    /// </summary>
    private const float Ros = 0.15f;

    /// <summary>
    /// Сжимаемость
    /// </summary>
    private const float Cps = 0.000165f;

    /// <summary>
    /// Проницаемость/вязкость
    /// </summary>
    private const float Lams = 0.11f;

    /// <summary>
    /// Коэффициент взаимодействия пласта с нагнетательной скважиной
    /// </summary>
    private const float Alfainj = 1;

    /// <summary>
    /// Коэффициент взаимодействия пласта с добывающей скважиной
    /// </summary>
    private const float Alfaprod = 1;

    #endregion

    /// <summary>
    /// Матрица распределения.
    /// </summary>
    private float[,] TXY;

    /// <summary>
    /// Время (в днях).

```

```

/// </summary>
public float Time = 365;

/// <summary>
/// Давление на границе ВС.
/// </summary>
public float Pinj = 350;

/// <summary>
/// Давление среды.
/// </summary>
public float Pres = 250;

/// <summary>
/// Давление на границе FG.
/// </summary>
public float Pprod = 150;

/// <summary>
/// Ширина рассматриваемого участка.
/// </summary>
public float Lx = 300f;

/// <summary>
/// Длина рассматриваемого участка.
/// </summary>
public float Ly = 500f;

/// <summary>
/// Длина трещины
/// </summary>
public float l = 50f;

/// <summary>
/// Количество точек в матрице распределения по оси OX.
/// </summary>
public int xSize;

/// <summary>
/// Количество точек в матрице распределения по оси OY.
/// </summary>
public int ySize;

/// <summary>
/// Количество итераций (шаг по времени = Time/K)
/// </summary>
public int K = 100;

/// <summary>
/// Конструктор класса.
/// </summary>
public MatrixBuilder(int x, int y)
{
    xSize = x;
    ySize = y;
    TXY = new float[xSize, ySize];
}

#region CalculatingMethods

/// <summary>
/// Метод расчёта прогоночных коэффициентов по оси OX.
/// </summary>

```

```

OX.</param>
    /// <param name="Nx">Количество точек в матрице распределения по оси
    /// <param name="hx">Шаг по оси OX.</param>
    /// <param name="tau">Шаг по времени.</param>
    /// <param name="TY1">Обрабатываемый ряд матрицы.</param>
    /// <returns>Обработанный ряд матрицы.</returns>
    private static float[] ABCD_X(
        int Nx,
        float hx,
        float tau,
        float[] TY1)
    {
        float[]
            Axi = new float[Nx],
            Bxi = new float[Nx],
            Cxi = new float[Nx],
            Dxi = new float[Nx];

        // Граничные условия для левой границы
        Axi[0] = 1;
        Bxi[0] = 1;
        Dxi[0] = 0;

        // Граничные условия для правой границы
        Axi[Nx - 1] = 1;
        Cxi[Nx - 1] = 1;
        Dxi[Nx - 1] = 0;

        for (var i = 1; i < Nx - 1; i++)
        {
            var hx2 = hx * hx;
            Axi[i] = 2 * (Cps * Ros / tau + Lams / hx2);
            Bxi[i] = Lams / hx2;
            Cxi[i] = Lams / hx2;
            Dxi[i] = 2 * Cps * Ros * TY1[i] / tau;
        }

        return TDMA(Nx, Axi, Bxi, Cxi, Dxi);
    }

    /// <summary>
    /// Метод расчёта прогоночных коэффициентов по оси OY.
    /// </summary>
    /// <param name="Ny">Количество точек в матрице распределения по оси
OY.</param>
    /// <param name="hy">Шаг по оси OY.</param>
    /// <param name="tau">Шаг по времени.</param>
    /// <param name="index">Индекс текущего ряда.</param>
    /// <param name="Pinj">Давление на границе BC.</param>
    /// <param name="Pprod">Давление на границе FG.</param>
    /// <param name="Nxlg">Количество точек на границе BC.</param>
    /// <param name="TX1">Обрабатываемый ряд матрицы.</param>
    /// <returns>Обработанный ряд матрицы.</returns>
    private static float[] ABCD_Y(
        int Ny,
        float hy,
        float tau,
        float index,
        float Pinj,
        float Pprod,
        float Nxlg,
        float[] TX1)
    {
        float[] Ayi = new float[Ny],

```

```

        Byi = new float[Ny],
        Cyi = new float[Ny],
        Dyi = new float[Ny];

// Граничные условия для нижней границы
if (index >= Ny - Nxlg)
{
    Ayi[0] = 1 + Alfaprod * hy / Lams;
    Byi[0] = 1;
    Dyi[0] = Pprod * Alfaprod * hy / Lams;
}
// Граница CD.
else
{
    Ayi[0] = 1;
    Byi[0] = 1;
    Dyi[0] = 0;
}

// Граница BC.
if (index <= Nxlg)
{
    Ayi[Ny - 1] = 1 + Alfainj * hy / Lams;
    Cyi[Ny - 1] = 1;
    Dyi[Ny - 1] = Pinj * Alfainj * hy / Lams;
}
// Граница CD.
else
{
    Ayi[Ny - 1] = 1;
    Cyi[Ny - 1] = 1;
    Dyi[Ny - 1] = 0;
}

for (var i = 1; i < Ny - 1; i++)
{
    var hx2 = hy * hy;
    Ayi[i] = 2 * (Cps * Ros / tau + Lams / hx2);
    Byi[i] = Lams / hx2;
    Cyi[i] = Lams / hx2;
    Dyi[i] = 2 * Cps * Ros * TX1[i] / tau;
}

return TDMA(Ny, Ayi, Byi, Cyi, Dyi);
}

/// <summary>
/// Процедура трёхточечной прогонки.
/// </summary>
/// <param name="n">Длина ряда.</param>
/// <param name="a">Ряд А.</param>
/// <param name="b">Ряд В.</param>
/// <param name="c">Ряд С.</param>
/// <param name="d">Ряд D.</param>
/// <returns>Обработанный ряд матрицы.</returns>
private static float[] TDMA(
    int n,
    float[] a,
    float[] b,
    float[] c,
    float[] d)
{
    var T = new float[n];
    var P = new float[n];

```

```

var Q = new float[n];

P[0] = b[0] / a[0];
Q[0] = d[0] / a[0];
for (var i = 1; i < n; i++)
{
    var denominator = a[i] - c[i] * P[i - 1];
    P[i] = b[i] / denominator;
    Q[i] = (d[i] + c[i] * Q[i - 1]) / denominator;
}

T[n - 1] = Q[n - 1];

for (var i = n - 2; i >= 0; i--)
{
    T[i] = P[i] * T[i + 1] + Q[i];
}

return T;
}

#endregion

/// <summary>
/// Процедура расчёта матрицы распределения.
/// </summary>
/// <returns>Матрица распределения.</returns>
public float[,] CreateMatrix()
{
    var Nx = xSize;
    var Ny = ySize;

    // Шаг по оси OX.
    var hx = Lx / Nx;

    // Шаг по оси OY.
    var hy = Ly / Ny;

    // Количество точек на границе BC.
    var Nxlg = (int)(l / hx);

    for (var i = 0; i < Nx; i++)
        for (var j = 0; j < Ny; j++)
        {
            TXY[i, j] = Pres;
        }

    float[]
        TY1 = new float[Nx],
        TX1 = new float[Ny];

    // Шаг по времени.
    float tau = Time / K;

    for (var L = 0; L < K; L++)
    {
        // Направление OX.
        for (var j = 0; j < Ny - 1; j++)
        {
            TY1 = TXY.ToArray(j, Nx, RowType.Line);
            TXY.FillRow(
                ABCD_X(Nx, hx, tau, TY1),
                j,
                Nx,

```

```

        RowType.Line);
    }

    TXY.FillRow(
        TXY.ToArray(1, Nx, RowType.Line),
        0,
        Nx,
        RowType.Line);

    TXY.FillRow(
        TXY.ToArray(Ny - 1, Nx, RowType.Line),
        Ny - 2,
        Nx,
        RowType.Line);

    for (var i = 0; i < Nx - 1; i++)
    {
        TX1 = TXY.ToArray(i, Ny, RowType.Column);
        TXY.FillRow(
            ABCD_Y(Ny, hy, tau, i, Pinj, Pprod, Nxlg,
                TX1),
            i,
            Ny,
            RowType.Column);
    }

    TXY.FillRow(
        TXY.ToArray(1, Ny, RowType.Column),
        0,
        Ny,
        RowType.Column);

    TXY.FillRow(
        TXY.ToArray(Nx - 2, Ny, RowType.Column),
        Nx - 1,
        Nx,
        RowType.Column);
    }

    return TXY;
}

}

/// <summary>
/// Исполняемая программа.
/// </summary>
internal class Program
{
    /// <summary>
    /// Точка входа.
    /// </summary>
    static void Main()
    {
        var xSize = 100;
        var ySize = 100;
        var builder = new MatrixBuilder(xSize, ySize);

        System.Console.WriteLine("Calculating pressure matrix...");
        var matrix = builder.CreateMatrix();

        System.Console.WriteLine("Writing in file...");
        using (var fileStream = new FileStream("matrix.txt",
            FileMode.OpenOrCreate))
            using (TextWriter tw = new StreamWriter(fileStream))

```

```
{
    for (int j = 0; j < xSize; j++)
    {
        for (int i = 0; i < ySize; i++)
        {
            tw.Write($"{matrix[i, j]} ");
        }
        tw.WriteLine();
    }
}
System.Console.WriteLine("Finished. Press any key to quit.");
System.Console.ReadKey();
}
}
```