

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ МОДЕЛИРОВАНИЯ ФЛЮИДОДИНАМИКИ И ТРАНСФАЗНЫХ ПРОЦЕССОВ В КОЛЛЕКТОРАХ, СОДЕРЖАЩИХ ГАЗОГИДРАТЫ

© 2019 г. *П.И. Рагимли¹, О.Р. Рагимли¹, Ю.А. Повещенко^{2,3},
В.О. Подрыга², И.В. Гасилова²*

¹ Московский физико-технический институт (государственный университет)

² Институт прикладной математики им. М.В. Келдыша РАН

³ Национальный исследовательский ядерный университет МИФИ
pvictoria@list.ru

DOI: 10.1134/S0234087919020072

Представлено описание модели, алгоритма и структуры прикладного программного комплекса (кода), предназначенного для моделирования течений в пористой среде, содержащей соединения природных газов (метан и др.) с водой или газовые гидраты. Код «HYDRATID» позволяет решать уравнения флюидодинамики в талой зоне и пьезопроводной среде с учетом фазовых переходов. Код «HYDRATID» реализован на языке JavaScript и работает в рамках компьютерной архитектуры «клиент – сервер».

Ключевые слова: JavaScript, Node.JS, компьютерная модель “клиент – сервер”, газогидрат, флюидодинамика, талая зона, пьезопроводность.

APPLICATION SOFTWARE FOR SIMULATIONS OF FLUID DYNAMICS AND PHASE TRANSFORMATIONS IN COLLECTORS WITH GAS-HYDRATE DEPOSITIONS

P.I. Rahimly¹, O.R. Rahimly¹, Yu.A. Poveshchenko^{2,3}, V.O. Podryga², I.V. Gasilova²

¹ Moscow Institute of Physics and Technology (State University)

² Keldysh Institute of Applied Mathematics of RAS

pvictoria@list.ru

³ National Research Nuclear University MEPhI

The work concerns with model, algorithm and data structures put on the basis of the program complex (code), intended for modeling flows in porous media containing gas hydrates, compounds of natural gas (methane et al.) and water. The code «HYDRATID» solves the fluid dynamics equations in the thawed zone and piezoconductivity medium with gas hydrate inclusions. The code is written in JavaScript and organized according to the principles of computer architecture "client-server".

Key words: JavaScript, Node.JS, client-server technology, gas hydrates, fluid dynamics, thawed zone, piezoconductivity.

1. Введение

Изучение поведения соединений природного газа с водой, или газовых гидратов, в природных и техногенных условиях активно развивается на протяжении многих лет в нефтегазовой отрасли [1-3]. Мотивы интереса к газогидратам разнообразны. Попадание по технологическим причинам водяного пара в проточные части систем добычи и переработки природного газа приводит при температурах ниже точки росы к образованию твердых газогидратов (гидратных пробок). Техногенные гидратные пробки могут формироваться в стволах скважин, в призабойной зоне. Эти явления негативно влияют на функциональные характеристики добывающего и транспортного оборудования. С другой стороны, в настоящее время изучаются перспективные варианты технологий, в которых газогидратам отводится центральная роль – как непосредственного источника газового сырья, так и в качестве средства компактного хранения добытого газа, средства накопления запасов энергии, опреснения морской воды, газоочистки и т.д. Повышение температуры или понижение давления вызывают смену фазового состояния газогидратов с высвобождением весьма большого количества газа (в основном, метана) – около 160 кубометров из одного кубометра твердого газогидрата. Оценки объемов залежей газогидратов у разных авторов имеют сильный разброс, однако считается, что всего лишь десятая часть полного запаса, в случае пригодности для извлечения, вдвое превысит разведанные во всем мире запасы обычного природного газа [3].

Соответственно, к наиболее важным фундаментальным проблемам, которые необходимо решать при проектировании технологий добычи или технического использования газогидратов относятся, прежде всего, воздействия сопутствующих газовых выбросов на экологию, а также снижение рисков разрушений оборудования из-за быстроты фазовых переходов в гидрате при изменении внешних условий [4, 5].

Вычислительные эксперименты с моделями природных и техногенных процессов, сопровождающихся формированием и разложением гидратов, востребованы и в теоретических исследованиях, и при сопровождении экспериментальных работ, в силу многопараметричности и нелинейности моделей динамики гидратосодержащих сред.

Комплекс прикладных программ, представленный в настоящей работе, создан с целью внедрения в практику моделирования результатов работы [6], где предложена термодинамически равновесная дискретная модель совместной двухкомпонентной (H_2O , CH_4) трехфазной (вода, газ, гидрат) флюидодинамики в пьезопроводной зоне коллектора и двухфазной динамики

флюидов в талой зоне (в которой отсутствуют газогидраты), для которой производится расщепление по физическим процессам. Там же построена однородная разностная схема сквозного счета и предложен алгоритм решения разностных уравнений.

В настоящее время средства управления прикладным кодом, такие как, например, JCL – Job Control Language, консольные приложения и т.п. [7], практически вытеснены графическими пользовательскими интерфейсами в силу их очевидных преимуществ в отношении эргономики и повышения производительности труда. Прикладной код «HYDRAT1D» разработан нами на основе вычислительной архитектуры «Клиент – сервер» [8]. Пользовательская часть (клиент) написана на языках JS, HyperText Markup Language (HTML) и на формальном языке Cascading Style Sheets (CSS), описывающем внешний вид интерфейса [9]. Язык для создания программных интерфейсов JavaScript (JS) встроен практически во все веб-браузеры и потому доступен без особых ограничений на тип компьютера [9]. Часть, отвечающая за вычисления (сервер), написана на NodeJS [10]. Программная платформа Node.js, основанная на интерпретаторе (“движке”) V8, превращает этот узкоспециализированный язык в язык общего назначения. У языка появляются новые возможности, такие как взаимодействие с устройствами ввода-вывода через свой модуль, написанный на C++, возможность подключать другие библиотеки, написанные на иных языках, можно разрабатывать оконные приложения для Linux, Windows и macOS.

2. Базовая модель флюидодинамики

Математическая модель, описывающая динамику компонент смеси жидкостей в пористой среде, часть порового пространства которой занята газогидратными включениями, подробно обсуждается в [6]. Здесь мы приведем описание системы уравнений этой модели, решаемых численно кодом HYDRAT1D.

В пространственной области O с границей ∂O рассмотрим термодинамически равновесные двухкомпонентные (вода, метан) трехфазные уравнения фильтрационной флюидодинамики с газогидратными включениями [1, 2]:

$$\frac{\partial}{\partial t} \left\{ m \left[S_v S_w \rho_w + (1 - S_v) \rho_v \beta_w \right] \right\} + \frac{\partial}{\partial z} (\rho_w \mathbf{V}_w) + q_w = 0, \quad (1)$$

$$\frac{\partial}{\partial t} \left\{ m \left[S_v (1 - S_w) \rho_g + (1 - S_v) \rho_v (1 - \beta_w) \right] \right\} + \frac{\partial}{\partial z} (\rho_g \mathbf{V}_g) + q_g = 0, \quad (2)$$

$$\mathbf{V}_w = -\frac{kk_{rw}}{\mu_w} \left(\frac{\partial}{\partial z} P - g\rho_w \mathbf{k} \right), \quad (3)$$

$$\mathbf{V}_g = -\frac{kk_{rg}}{\mu_g} \left(\frac{\partial}{\partial z} P - g\rho_g \mathbf{k} \right), \quad (4)$$

$$\begin{aligned} & \frac{\partial}{\partial t} \left\{ m \left[S_v (S_w \rho_w \varepsilon_w + (1 - S_w) \rho_g \varepsilon_g) + (1 - S_v) \rho_v \varepsilon_v \right] + (1 - m) \rho_s \varepsilon_s \right\} + \\ & + \frac{\partial}{\partial z} \left\{ \rho_w \varepsilon_w \mathbf{V}_w + \rho_g \varepsilon_g \mathbf{V}_g + [P(\mathbf{V}_w + \mathbf{V}_g)] \right\} + \frac{\partial}{\partial z} \mathbf{W} + q_\varepsilon = 0, \end{aligned} \quad (5)$$

$$\mathbf{W} = - \left\{ m \left[S_v (S_w \lambda_w + (1 - S_w) \lambda_g) + (1 - S_v) \lambda_v \right] + (1 - m) \lambda_s \right\} \frac{\partial}{\partial z} (T), \quad (6)$$

$$T_{dis} = f(P) \quad (7)$$

с граничными условиями, которые в общем виде можно записать как

$$(\mathbf{X} = \mathbf{F}(u))_{\partial O}. \quad (8)$$

Принятые обозначения: индексы $l = g, w, v, s$ относятся к газу, воде, гидрату и скелету пористой среды. Здесь (1) – уравнение баланса массы воды, (2) – уравнение баланса массы метана, (3) – закон Дарси, определяющий скорость движения свободной воды \mathbf{V}_w в порах для двухкомпонентной несмешивающейся системы (вода, метан) с абсолютной проницаемостью $k = k(\mathbf{r}, S_v, P)$ и относительной проницаемостью $k_{rw} = k_{rw}(S_w)$. μ_w – вязкость воды. Аналогично понимается уравнение (4) как закон Дарси для скорости свободного метана \mathbf{V}_g с относительной проницаемостью $k_{rg} = k_{rg}(S_w)$ и вязкостью μ_g . Наконец, (5) – уравнение баланса общей внутренней энергии системы, включающей энергии свободной воды, свободного метана, гидрата и скелета. Под ε_l понимаются внутренние энергии единиц массы компонент. Уравнение (6) определяет общий тепловой поток \mathbf{W} в среде с коэффициентами теплопроводности $\lambda_l(P, T)$. В формулах: g – ускорение свободного падения; \mathbf{k} – координатный орт, направленный вертикально вниз; P – давление; S_w – водонасыщенность; v – гидратонасыщенность; $S_v = 1 - v$ – растепленность; $\rho_l(P, T)$ – плотности фаз; β_w – массовая доля воды в гидрате; \mathbf{r} – радиус-вектор; t – время; q_w , q_g и q_ε – плотности соответствующих источников, зависящие от параметров $(t, \mathbf{r}, S_w, S_v, P, T)$. По правилу фаз Гиббса трехфазная двухкомпонентная гидратная система является

многовариантной, т.е. обладает одной степенью свободы (температурой или давлением) [2]. Следовательно, для газового гидрата, находящегося в равновесии с жидкой водой или льдом, зависимость (7) носит однозначный характер. В (8) u, \mathbf{X} – произвольные скаляр (температура, давление, внутренняя энергия и т.п.) и вектор, физически связанный с градиентным потоком этой скалярной величины.

Энтальпии единицы массы $i_l = \varepsilon_l + P/\rho_l$ гидрата, свободных воды и газа термодинамически согласованы в смысле следующего соотношения:

$$\beta_w i_w + (1 - \beta_w) i_g = i_v + h, \quad (9)$$

где h – скрытая теплота фазового перехода единицы массы гидрата. Для удельных (на единицу массы) энтальпий фаз также справедливо [2] уравнение

$$di_l = c_{pl}(-k_{dl}dP + dT) \quad (10)$$

с коэффициентами дросселирования

$$k_{dl} = \frac{1}{c_{pl}} \left[T \left(\frac{\partial V_l}{\partial T} \right)_p - V_l \right]. \quad (11)$$

Здесь индекс « p » у скобок означает, что частная производная по температуре берется при постоянном давлении; c_{pl} и $V_l = 1/\rho_l$ – удельные теплоемкости (при постоянном давлении) и объемы фаз. В частности, для газовой фазы с уравнением состояния:

$$p_g = \frac{P}{z_g RT} \quad (12)$$

справедливо

$$k_{dg} = \frac{RT^2}{c_{pg}P} \frac{\partial z_g}{\partial T}, \quad (13)$$

что означает наличие эффекта Джоуля-Томсона ($k_{dg} \neq 0$) для неидеального газа с коэффициентом сверхсжимаемости z_g в изучаемых процессах изотермической фильтрации.

Из уравнений (1), (2), (5) путем исключения функций S_w и S_v из под знака производной по времени может быть получено уравнение пьезопроводности для флюидодинамики с гидратными включениями в следующем

виде [6]:

$$\begin{aligned}
 & m\delta_\varepsilon \left\{ S_v \left[S_w \frac{1}{\rho_w} \frac{\partial \rho_w}{\partial t} + (1-S_w) \frac{1}{\rho_g} \frac{\partial \rho_g}{\partial t} \right] + (1-S_v) \frac{1}{\rho_v} \frac{\partial \rho_v}{\partial t} + \frac{1}{m} \frac{\partial m}{\partial t} \right\} + \\
 & + \frac{\Psi}{m\rho_v} \left\{ m \left\{ S_v \left[S_w \rho_w \frac{\partial \varepsilon_w}{\partial t} + (1-S_w) \rho_g \frac{\partial \varepsilon_g}{\partial t} \right] + (1-S_v) \rho_v \frac{\partial \varepsilon_v}{\partial t} \right\} + \right. \\
 & \left. + \frac{\partial [(1-m)\rho_s \varepsilon_s]}{\partial t} \right\} + \delta_\varepsilon DIG + \frac{\Psi}{m\rho_v} DIG_\varepsilon = 0.
 \end{aligned} \tag{14}$$

Здесь

$$\begin{aligned}
 DIG &= \frac{1}{\rho_w} \frac{\partial}{\partial z} (\rho_w \mathbf{V}_w) + \frac{1}{\rho_g} \frac{\partial}{\partial z} (\rho_g \mathbf{V}_g) + \frac{q_w}{\rho_w} + \frac{q_g}{\rho_g}, \\
 DIG_\varepsilon &= \left[\frac{\partial}{\partial z} (\rho_w \varepsilon_w \mathbf{V}_w) - \varepsilon_w \frac{\partial}{\partial z} (\rho_w \mathbf{V}_w) \right] + \left[\frac{\partial}{\partial z} (\rho_g \varepsilon_g \mathbf{V}_g) - \varepsilon_g \frac{\partial}{\partial z} (\rho_g \mathbf{V}_g) \right] + \\
 & + \frac{\partial}{\partial z} [P(\mathbf{V}_w + \mathbf{V}_g)] + \frac{\partial}{\partial z} \mathbf{W} + (q_\varepsilon - \varepsilon_w q_w - \varepsilon_g q_g) = \\
 & = \rho_w \mathbf{V}_w \nabla \varepsilon_w + \rho_g \mathbf{V}_g \nabla \varepsilon_g + \frac{\partial}{\partial z} [P(\mathbf{V}_w + \mathbf{V}_g)] + \frac{\partial}{\partial z} \mathbf{W} + (q_\varepsilon - \varepsilon_w q_w - \varepsilon_g q_g).
 \end{aligned}$$

Также

$$\frac{\Psi}{m\rho_v} = \left(\varphi - \frac{1}{\rho_v} \right) \geq 0, \quad \varphi = \frac{\beta_w}{\rho_w} + \frac{(1-\beta_w)}{\rho_g}, \quad \delta_\varepsilon = \beta_w \varepsilon_w + (1-\beta_w) \varepsilon_g - \varepsilon_v \geq 0$$

– удельные скачки (на единицу массы) при фазовом переходе объема и внутренней энергии соответственно.

Далее также используется обозначение: $()_p = \partial/\partial P$.

Уравнение (14) представляет собой основное пьезопроводно-диссипативное термодинамически равновесное уравнение трехфазной двухкомпонентной флюидодинамики с гидратными включениями, расщепленное с блоком переноса процессов насыщения (1)-(4), обладающим на фоне зафиксированных термодинамических параметров среды в основном гиперболическими свойствами [6]. Введем новую величину – бароемкость гидратной системы D_P соотношением:

$$D_P = m\delta_\varepsilon \left\{ S_v \left[S_w \frac{(\rho_w)_p}{\rho_w} + (1-S_w) \frac{(\rho_g)_p}{\rho_g} \right] + (1-S_v) \frac{(\rho_v)_p}{\rho_v} + \frac{(m)_p}{m} \right\} +$$

$$\begin{aligned}
 & + \frac{\Psi}{m\rho_v} \{m\{S_v[S_w\rho_w(\varepsilon_w)_p + (1-S_w)\rho_g(\varepsilon_g)_p] + (1-S_v)\rho_v(\varepsilon_v)_p\} + \\
 & + [(1-m)\rho_s\varepsilon_s]_p,
 \end{aligned} \tag{15}$$

и перепишем уравнение (14) в более компактной форме

$$D_P \frac{\partial P}{\partial t} + \delta_\varepsilon DIG + \frac{\Psi}{m\rho_v} DIG_\varepsilon = 0. \tag{16}$$

В (15) берется полная производная по давлению с учетом зависимости (7). Воспользовавшись также этой зависимостью (7) и выбирая в качестве единственной термодинамической степени свободы в уравнениях (3), (4), (14) соответствующие внутренние энергии ε_w и ε_g , получим уравнение (14) в энергетическом представлении.

Двухфазная талая зона. Полагая растепленность $S_v = 1$, получим двухфазное уравнение в талой зоне:

$$\frac{\partial}{\partial t} \{mS_w\rho_w\} + \frac{\partial}{\partial z} (\rho_w \mathbf{V}_w) + q_w = 0, \tag{17}$$

$$\frac{\partial}{\partial t} \{m(1-S_w)\rho_g\} + \frac{\partial}{\partial z} (\rho_g \mathbf{V}_g) + q_g = 0, \tag{18}$$

$$\begin{aligned}
 & \frac{\partial}{\partial t} \{m(S_w\rho_w\varepsilon_w + (1-S_w)\rho_g\varepsilon_g) + (1-m)\rho_s\varepsilon_s\} + \\
 & + \frac{\partial}{\partial z} \{ \rho_w\varepsilon_w \mathbf{V}_w + \rho_g\varepsilon_g \mathbf{V}_g + [P(\mathbf{V}_w + \mathbf{V}_g)] \} + \frac{\partial}{\partial z} \mathbf{W} + q_\varepsilon = 0,
 \end{aligned} \tag{19}$$

$$\mathbf{W} = - \left\{ m(S_w\lambda_w + (1-S_w)\lambda_g) + (1-m)\lambda_s \right\} \frac{\partial}{\partial z} T. \tag{20}$$

Далее из уравнений (17)-(19), исключая функцию S_w из-под знака производной по времени, получим уравнения, определяющие неизотермический процесс пьезопроводности в талой зоне:

$$\frac{S_w}{\rho_w} \frac{\partial(m\rho_w)}{\partial t} + \frac{1-S_w}{\rho_g} \frac{\partial(m\rho_g)}{\partial t} + DIG = 0, \tag{21}$$

$$m\{S_w\rho_w \frac{\partial\varepsilon_w}{\partial t} + (1-S_w)\rho_g \frac{\partial\varepsilon_g}{\partial t}\} + \frac{\partial[(1-m)\rho_s\varepsilon_s]}{\partial t} + DIG_\varepsilon = 0. \tag{22}$$

Здесь комбинация массовых (DIG) и энергетических (DIG_ε) дивергенций вместе с действием соответствующих источников (q_w, q_g, q_ε) определяются аналогично (14).

3. Основные технологии программирования, использованные при создании кода HYDRAT1D

3.1. Архитектура «Клиент – сервер». В большинстве случаев компьютеры и программы, входящие в состав информационной системы, не являются равноправными. Некоторые из них владеют ресурсами (файловая система, процессор, принтер, база данных и т.д.), другие имеют возможность обращаться к этим ресурсам. Компьютер (или программу), управляющий ресурсом, называют сервером этого ресурса (файл-сервер, сервер базы данных, вычислительный сервер и т.д.). Клиент и сервер какого-либо ресурса могут находиться как на одном компьютере, так и на различных компьютерах, связанных сетью.

Программа-сервер ожидает от клиентской программы запрос и предоставляет ему свои ресурсы в виде данных. В нашем случае клиент обращается к серверу посредством HTTP (HyperText Transfer Protocol) запросов. HTTP это сетевой протокол прикладного уровня для передачи произвольных данных. Преимущества такого подхода состоят в том, что:

- 1) исходный код не дублируется;
- 2) требования к ресурсам клиента существенно снижаются;
- 3) к серверу можно обращаться из любого устройства с любой операционной системой, поддерживающей сетевой протокол HTTP;
- 4) защищенность и целостность данных;
- 5) управление правами доступа к ресурсу.

3.2. Язык программирования Java Script. JavaScript является объектно-ориентированным языком [11]. Используемое в JS прототипирование, или первичная (как правило, малозатратная) реализация базовой функциональности системы, предназначенная для анализа ее работы и дальнейшего совершенствования проекта, обуславливает отличие JS в работе с объектами по сравнению с традиционными класс-ориентированными языками [11, 12]. Кроме того, JS поддерживает функциональное программирование: функции как объекты первого класса, объекты как списки, преобразование функций типа “carrying” (карринг, 13), анонимные функции, замыкания – что придаёт языку дополнительную гибкость [13, 14].

Синтаксис JS очень похож на синтаксис Си, однако JS по сравнению с Си имеет коренные отличия:

- объекты с возможностью интроспекции (аналогично C++);
- функции как объекты первого класса;
- автоматическое приведение типов;
- автоматическое управление памятью 14;
- анонимные функции.

Эти отличия делают JS более удобным для программной реализации в технологии “клиент-сервер”.

4. Комплекс прикладных программ HYDRAT1D

Модульная система Node. Для реализации проекта кода нам понадобилось воспользоваться модульной системой Node. Она построена достаточно просто. Чтобы не встраивать непосредственно в Node все возможные компоненты, разработчики предлагают получать дополнительную функциональность посредством модулей. В основу положен договор, которого придерживаются разработчики, чтобы их модули могли работать вместе с другими модулями. В Node реализованы следующие требования:

1. Доступна глобальная функция `require`, которая принимает идентификатор модуля и возвращает экспортируемый прикладной программный интерфейс.
2. Именем модуля является строка символов, которая может включать в себя начальные разделители – слэши, чтобы идентифицировать путь к файлу.
3. Модуль должен экспортировать только те методы, которые должны быть видимы за пределами его области видимости.
4. Переменные в модуле являются приватными

Система загрузки модулей в языке реализована так, чтобы между файлом и модулем было установлено однозначное соответствие. Для подключения модуля в приложение нужно в команду `require` передавать строку с идентификатором модуля. Есть 3 типа импортируемых модулей:

1. Модули из папки `node_modules`. Например, `var math = require('mathjs');` Есть возможность также подключить конкретный объект: `var subtract = require('mathjs').subtract`. Требуется установка: `npm install mathjs`.
2. Модули, входящие в состав Node. Подключается аналогично первому пункту. Встроены в ядро Node.
3. Пользовательские модули. Для загрузки в программную среду необходимо указать путь к нему. Например, в следующей инструкции `require` Node ожидает найти модуль по имени `sweep.js` в том же каталоге, где находится Node-приложение: `require('./lib/math/sweep.js')`. Или же можно использовать полный путь: `require('/home/username/HYDRAT1D/lib/math/sweep.js')`.

Корневая директория программы состоит из трех вложенных папок и из четырех файлов. Программа запускается с помощью файла `server.js`. Файл одновременно запускает клиентскую и серверную часть приложения. Клиентский код лежит в директории `public`, а серверная (вычислительная) часть в `lib`. Файл `routes.js` в папке `app` отвечает за маршрутизацию клиентских запросов к конкретному адресу. Он же возвращает рассчитанные (и не только) данные пользователю. В нашем случае вычисление происходит по GET запросу `http://www.[хост]/api/calculation?input={объект с входными данными}`. Если проект развернут локально, то вместо хоста будет `localhost:8080`. 8080 – это номер порта, если у вас он занят другим процессом, то его можно изменить в файле `server.js`.

Получив данные от клиента, роутер передает их в файл `lib/calculations/calculation.js`, запуск программы производится командами в соответствии с табл.1.

Таблица 1.

Название	Описание	Доступ
<code>init</code>	Инициализация входных данных. Принимает объект с начальными данными. Возвращает ссылку на сам объект (<code>this</code>).	Открытый
<code>start</code>	Запуск процесса вычисления и методов идущих после вычисления.	Открытый
<code>result</code>	Возвращает объект с параметрами, которые нужны пользователю. Последовательно вызывается маршрутизатором после окончания метода <code>calculation.start()</code> .	Открытый
<code>run_calculation</code>	Запуск вычислений	Закрытый
<code>save_to_file</code>	Сохраняет расчеты в файл. Принимает объект с полем <code>name</code> , где нужно указывать название папки. По умолчанию название папки <code>calc {временной штамп}</code> .	Закрытый
<code>process_data</code>	По умолчанию не включена в программу. Создавалась для обработки локальных сохраненных файлов.	Закрытый

Метод `init` обращается к файлу `lib/init.js`, который заполняет и местами переопределяет параметры задачи (исходные данные), которые хранятся в файле `/lib/params/parameters.js` начальные данные, переданные из клиента (метод `init`), а затем запускает процесс вычисления методом `start`. Файл `parameters.js` содержит в себе все параметры задачи и процедуры ее решения, а также несколько открытых методов, которые оперируют с этими параметрами. Для удобства пользователя код `HYDRATID` дополнен справочной системой, которая помогает ориентироваться в действиях по управлению кодом, и содержит описание вводимой информации – исходных физических данных, параметров расчетной сетки и параметров управления рас-

четом. В частности, там содержится описание файлов `lib/calculations/calculation.js`, `/lib/params/parameters.js` и `parameters.js`.

Решение системы уравнений базовой модели (1)-(7) выполняется в соответствии с блок-схемой, представленной на рис.1.

Ключевым пунктом работы является определение фазового состава всего алгоритма расчета динамики смеси газа с водой при возможном наличии твердой газогидратной компоненты. Соответствующий алгоритм построен таким образом, что вначале определяются фазы задачи относительно значений искомых параметров в n -м слое. Получаем некий контейнер-массив с объектами, характеризующими фазы. Например, [{тип 1-й фазы, левая граница 1-й фазы, правая граница 1-й фазы}, ..., {тип n -й фазы, левая граница n -й фазы, правая граница n -й фазы}].

```
var container=[{type:'triple',from:0, to:0.4},
{type:'melted',from:0.4, to:0.6}, {type:'triple',from:0.6, to:1}]
```

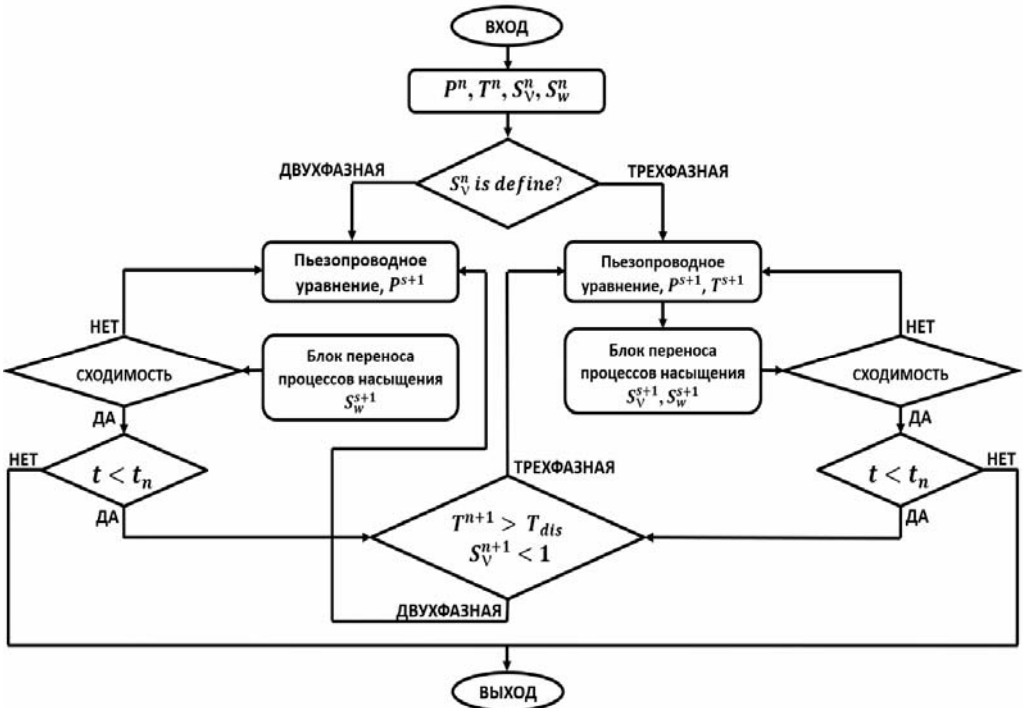


Рис.1. Блок-схема решения связанной задачи.

Затем выполняется цикл по контейнер-массиву, в котором для каждого типа фаз вычисляются граничные точки (`define_edges`), и определяются модули, реализующие для дальнейших расчетов методы соответственно типу фазы.

Основной расчет выполняется в цикле `while`. Здесь в теле оператора вызывается метод `run` модуля `iteration-process.js`, который наследует методы объекта `lib/math/iteration.js` и производит вычисления, завершающиеся методом `save_calculation`. Более низкоуровневые вычисления для блоков пьезопроводности и сатурации лежат в соответствующих папках `./lib/piezocon/` и `./lib/saturation/`. Все функции определены в файле `functions.js`, а при необходимости в этот файл можно добавить новые функции.

5. Работа с кодом HYDRAT1D

Мы отмечали, что для использования комплекса программ (кода) `HYDRAT1D` необходимо установить на компьютере пользователя серверную платформу `Node.js`. По ссылке <https://nodejs.org/en/download/> можно скачать соответствующий установочный файл для трех основных операционных систем `Windows`, `Linux` и `macOS`. Сразу после установки становится доступна новая команда – `node`. Её можно использовать двумя разными способами. Первый способ использования – применить команду без аргументов. При этом откроется интерактивная оболочка, где можно выполнять обычный `JS` код, например:

```
/$ node /> console.log('Hello World');/Hello World/undefined/
```

(здесь разделитель “слэш” означает, что каждая команда пишется на отдельной строке).

В приведенном примере мы написали `console.log('Hello World')` в оболочке. При нажатии на ввод (`Enter`) `Node.js` выполнит этот код, и мы увидим сообщение `Undefined`, поскольку оболочка отображает возвращаемое значение каждой команды, а `console.log` в данном случае ничего не возвращает. В другом варианте использования команды `node` мы можем передать `Node` файл с `JS` для выполнения. Именно таким образом выполняется запуск кода «`HYDRAT1D`». Вместе с командой `node` становится доступной команда пакетного менеджера `npm`. Этот инструмент позволяет локально или глобально устанавливать необходимые для работы пакеты.

Запуск программы выполняется командой `npm install` из корневой директории. По этой команде устанавливаются все нужные модули. Далее выполняется `node server.js`, тем самым запустится локальный веб сервер и клиентская часть, отвечающая по локальному адресу `localhost` на порте `8080` (<http://localhost:8080>). Блок-схема работы кода в технологии “клиент-сервер” представлена на рис.2.

Для оценки производительности кода был выполнен ряд вычислительных экспериментов в соответствии с постановками задач, описанными в [6].

В [6] рассматривается термодинамически равновесная совместная дискретная модель двухкомпонентной (H_2O , CH_4) трехфазной (вода, газ, гидрат) фильтрационной флюидодинамики и двухфазных процессов в талой зоне с отсутствием газогидратов, для которой производится расщепление по физическим процессам. Целью исследования является построение как в талой зоне, так и в среде с газогидратными включениями совместного семейства двухслойных полностью консервативных разностных схем метода опорных операторов с профилированными по пространству временными весами в соответствии с предлагаемым алгоритмом расщепления решения уравнений базовой модели по физическим процессам.



Рис.2. Блок-схема работы кода HYDRAT1D.

Возможность сквозного расчета течения в талой зоне и пьезопроводной среде по однородной разностной схеме согласованной аппроксимации обеспечивает высокое разрешение структуры течения при весьма умеренных затратах вычислительных ресурсов.

Время решения одной задачи в приведенной выше постановке по однородной схеме сквозного счета при наличии талой и пьезопроводной зон и при больших характерных временах фильтрации, на сетке порядка тысячи ячеек в расчетной области составляет около получаса. При необходимости выполнения массовых расчетов возможен запуск кода в параллельном режиме. Для масштабирования пропускной способности на устройстве, на котором установлено разработанное приложение, запускается несколько процессов командой Node.js – по одному процессу на ядро процессора. Node.js

обеспечивает отличную совместимость с процессором и масштабирует пропускную способность практически линейно. В наших экспериментах на сервере с 8-ядерным процессором параллельно выполнялось решение восьми задач за время решения одной задачи в последовательном режиме.

В [8] приведено решение характерной трансфазной задачи. Решение выполнялось на сетке в 100 шагов на персональном компьютере со следующими характеристиками: процессор: 2.7Ghz Intel Core i5, память: 8Gb 1867Mhz DDR3. Данные по времени расчета трех состояний среды в области расчета приведены в табл.2, где (a) – завершен этап формирования талой зоны, (b) – этап интенсивного расширения талой зоны, (c) – этап сглаживания градиента давления и постепенного уменьшения (в конце этапа примерно в 4-5 раз) темпа роста размера талой зоны.

Таблица 2.

Этап эволюции талой зоны	Время расчета (сек)	Время серверной обработки (мс)	Суммарное время отправки и получения данных (мс)	Время обработки данных (визуализация) (мс)
(a)	1	869	64	75
(b)	10	2603	68	83
(c)	100	14360	90	97

Расчеты проводились последовательно. Увеличение времени визуализации результатов расчетов связано с тем, что при добавлении новых графиков старые не уничтожаются (для проведения сравнительного анализа).

Для ускорения прогноза эволюции газогидратных включений по результатам численного анализа задач флюидодинамики пользователям предоставляется возможность кастомизировать исходную модель путем модификаций внешнего интерфейса. Интерфейс пользователя осваивается достаточно быстро, поскольку он интуитивно понятен. Клиентская часть дает возможность произвольно менять распределения начальных величин, граничных условий, различных источников в виде сложных функций с различными условиями и т.д. Такая возможность достигается с помощью многофункциональной инструкции eval() языка JS. Метод eval выполняет JS код, предоставленный строкой. При разработке JS была цель сделать язык доступным для разработки приложений широкому кругу программистов различного уровня подготовки. Владея минимальными знаниями языка JS, пользователь может прямо из интерфейса изменить постановку задачи. Чтобы повторить подобное на хорошо известных языках, например, C++ или Fortran, нужно предварительно отредактировать код и откомпилировать его.

Результаты расчетов обрабатываются на компьютере пользователя. Скорость визуализации данных зависит от вычислительных ресурсов устройства. Подключенная к пользовательской части библиотека с открытым исходным кодом Plotly.js (plot.ly) дает возможность быстро импортировать, анализировать и экспортировать данные – строить качественные линейные графики за миллисекунды (в зависимости от браузера и процессора), строить гистограммы, тепловые карты, 2D, 3D представления расчетных данных.

6. Заключение

Прогресс в развитии JS дает веб-разработчикам все более широкие возможности создания разнообразных технологий и инновационных приложений. В настоящей работе был использован один из наиболее интересных и популярных инструментов для создания легко масштабируемых сетевых приложений Node.js – это серверная реализация языка JS, основанная на разработанном компанией Google JS-движке V8, который был создан Райаном Далем (Ryan Dahl) [10]. Развитием проекта сейчас занимается компания Joyent, крупный провайдер облачных вычислений в США. Серверная среда Node.js состоит из 80% кода C/C++ (ядро).

Язык JS не рассчитан для больших вычислительных задач. Однако он позволяет создавать короткие наглядные коды и показателен в том смысле, что имеет однопоточную модель исполнения. Кроме того, язык JS имеет возможность распараллелить вычисления, переведя их в асинхронную парадигму и распределив между несколькими процессами, но достичь производительности аналогичной программы на C++ с тем условием, что код был написан опытным программистом и представляется не реалистичным. С другой стороны, в случае одномерных постановок в рамках общей модели [3] для вычислений вполне достаточно одного потока, поэтому разработка быстро исполняемого кода не представляет труда. По производительности он будет сравним с аналогичным кодом на языке C++. При оценке скорости исполнения кода нужно учитывать, помимо паттернов и подходов к программированию, еще и потребление памяти, нагрузку на CPU и т.д. На сегодняшний день JS является самым быстрым интерпретируемым языком в мире. Интерпретаторы, написанные под JS от Mozilla – SpiderMonkey, от Google – V8 и от Microsoft – Chakra создавались в конкурентной борьбе лучших разработчиков. Следовательно, код, написанный на JS, может быть оптимизирован. Если в дальнейшем SpiderMonkey будет работать быстрее, чем V8, то при использовании Node.JS переход на новый движок не составит труда. Безусловно, возможны и другие реализации обсуждаемого прикладного кода. Например, функциональные модули пишутся на C++, а далее импортируются в среду JS и далее работа с ними идет в рамках технологии “клиент-сервер”.

Выполненные разработки создают основу для реализации дальнейших проектов создания цифровых технологий, востребованных в нефтегазовой отрасли. Так, в настоящее время разрабатываются проекты больших программных комплексов – "цифровых месторождений", в которых коды математического моделирования интегрируются с кодами обработки данных контрольно–измерительной аппаратуры, установленной на скважинах, в газотранспортных системах и т.д., а также с системами принятия решений. Решения с последующей выдачей рекомендаций персоналу, занятому эксплуатацией, предполагается вырабатывать за характерное время порядка нескольких часов, а при возникновении нештатных ситуаций – гораздо быстрее. В этой связи следует отметить, что во многих случаях применения компьютерного моделирования с целью экспресс-анализа производственной ситуации, целесообразно использовать полную физическую модель при упрощенном описании геометрии потока. Именно для таких случаев и был разработан код HYDRAT1D, показавший себя экономичным и робастным программным средством моделирования связанных флюидодинамических процессов в коллекторе с газогидратными включениями.

СПИСОК ЛИТЕРАТУРЫ

1. *Yu.F. Makogon*. Hydrates of hydrocarbons. – Tulsa: Pennwell Publishing Co., 1997, 484 p.
2. *В.А. Истомин, В.С. Якушев*. Газовые гидраты в природных условиях. – М.: Недра, 1992, 236 с.;
V.A. Istomin, V.S. Yakushev. Gazovye gidraty v prirodnykh usloviikh. – М.: Nedra, 1992, 236 s.
3. *Ф.А. Кузнецов, В.А. Истомин, Т.В. Родионова*. Газовые гидраты: исторический экскурс, современное состояние, перспективы исследований // Российский химический журнал, 2003, т.48, №3, с.5–18;
F.A. Kuznetsov, V.A. Istomin, T.V. Rodionova. Gazovye gidraty: istoricheskii ekskurs, sovremennoe sostoianie, perspektivy issledovaniia // Rossiiskii himicheskii zhurnal, 2003, t.48, №3, s.5–18.
4. *Д. Колесов, И. Ронова, А. Талис, И. Яминский, Г. Терещенко*. Газовые гидраты – объект исследования и применения // Наноиндустрия: журнал, 2010, т.1, с.20–23;
D. Kolesov, I. Ronova, A. Talis, I. Jaminskii, G. Tereshchenko. Gazovye gidraty – obieekt issledovaniia i primeneniia // Nanoindustriia: zhurnal, 2010, t.1, s.20–23.
5. *В.Ш. Шагапов, Н.Г. Мусакаев, М.К. Хасанов*. Нагнетание газа в пористый резервуар, насыщенный газом и водой // Теплофизика и аэромеханика, 2005, т.12, №4, с.645–656;
V.Sh. Shagapov, N.G. Musakaev, M.K. Khasanov. Nagnetanie gaza v poristy rezervuar, nasyshchenny gazom i vodi // Teplofizika i aeromekhanika, 2005, t.12, №4, s.645–656.
6. *П.И. Рагимли, Ю.А. Повецено, В.О. Подрыга, О.Р. Рагимли, С.Б. Попов*. Моделирование процессов совместной фильтрации в талой зоне и пьезопроводной среде с газогидратными включениями / Препринт ИПМ им. М.В. Келдыша, 2018, №40, 32с.;

- P.I. Rahimly, Yu.A. Poveshchenko, V.O. Podryga, O.R. Rahimly, S.B. Popov.* Modelirovanie protsessov sovmestnoi filtratsii v taloi zone i pezoпродводnoi srede s gazogidratnymi vklucheniiami / Preprint IPM im. M.V. Keldysha, 2018, №40, 32s.
7. *C.B. Germain.* Programming the IBM 360. – Englewood Cliffs, New Jersey: Prentice Hall, Inc., 1967.
 8. *П.И. Рагимли.* Комплекс программ для моделирования многофазных течений в коллекторах с газогидратными включениями // Препринт ИПМ им. М.В. Келдыша, 2018, №103, 25с.;
P.I. Rahimly. Kompleks programm dlia modelirovaniia mnogofaznykh techenii v kollektorakh s gazogidratnymi vklucheniiami / Preprint IPM im. M.V. Keldysha, 2018, №103, 25s.
 9. *J.N. Robbins.* Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics. 4th Edition. – Canada: O'Reilly Media, 2012.
 10. *E. Brown.* Web Development with Node and Express. – USA: O'Reilly Media, 2014.
 11. *N. McKenzie.* Is JavaScript is object-oriented language? // Dr. Dobb's Journal: Software tools for the professional Programmer, 2001, v.26, №8, p.115–116.
 12. *K. Simpsons.* You Don't Know JS: this & Object Prototypes. – USA: O'Reilly Media, 2014, 174p.
 13. *B.C. Pierce.* Types and Programming Languages. – Cambridge, Massachusetts, London, England: The MIT Press, 2002, 645p.
 14. *E. Ilyushin, D. Namiot.* On memory management in Javascript applications // International Journal of Open Information Technologies, 2015, v.3, №10, p.11–15.

Поступила в редакцию 13.06.2018

После доработки 13.06.2018

Принята к публикации 10.09.18