ИПМ им.М.В.Келдыша РАН • Электронная библиотека

Препринты ИПМ • Препринт № 103 за 2018 г.

ISSN 2071-2898 (Print)
ISSN 2071-2901 (Online)

Рагимли П.И.

Комплекс программ для
моделирования
многофазных течений в
коллекторах с
газогидратными
включениями

Рекомендуемая форма библиографической ссылки: Рагимли П.И. Комплекс программ для моделирования многофазных течений в коллекторах с газогидратными включениями // Препринты ИПМ им. М.В.Келдыша. 2018. № 103. 20 с. doi:[10.20948/prepr-2018-103](https://doi.org/10.20948/prepr-2018-103)
URL: <http://library.keldysh.ru/preprint.asp?id=2018-103>

О р д е н а Л е н и н а
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
имени М.В.Келдыша
Р о с с и й с к о й а к а д е м и и н а у к

П. И. Рагимли

**Комплекс программ для моделирования
многофазных течений в коллекторах
с газогидратными включениями**

Москва — 2018

Рагимли П. И.

Комплекс программ для моделирования многофазных течений в коллекторах с газогидратными включениями

Работа содержит описание алгоритмов и структуры созданного на их основе кода, предназначенного для моделирования течений в пористой среде, содержащей соединения низкомолекулярных природных газов (метан и др.) с водой, существующие при определенных термобарических условиях в твердой фазе. Программа HYDRAT1D позволяет решать уравнения флюидодинамики в талой зоне и пьезопроводной среде с газогидратными включениями, в плоской и цилиндрической геометриях с учетом различных физических эффектов. Программный код реализован на языке JavaScript (JS) на основе интерпретатора JavaScript («движка» V8).

Работа с кодом HYDRAT1D организована по принципам компьютерной архитектуры «Клиент-сервер», что делает удобным использование кода в режиме удаленного доступа, в облачных вычислениях и т.д. На примере расчетов течений в пористой гидратосодержащей среде обсуждаются достоинства использования данной архитектуры и языка программирования для создания прикладных кодов численного решения начально-краевых задач флюидодинамики.

Ключевые слова: JavaScript, Node.JS, Клиент-сервер, компьютерная модель, газогидрат, флюидодинамика, талая зона, пьезопроводность

Parvin Ilgar gizi Rahimly

On the software development using “client-server” technology for models of flows in porous media containing gas-hydrate depositions

The work deals with algorithms and data structures put on the basis of the code, intended for modeling flows in porous media containing gas hydrates, i.e. low-molecular compounds of natural gas (methane et al.) and water. At low temperature and certain pressure conditions gas hydrates exist in a solid phase occupying part of the pore space. The code HYDRAT1D solves the fluid dynamics equations in the thawed zone and piezoconductivity medium with gas hydrate inclusions. It accounts for a complete set of essential physical effects and implements calculations in 1D planar and cylindrical geometries. The code is written in JavaScript (JS) on the basis of the JS interpreter (the V8 “engine”).

Work with the code HYDRAT1D is organized according to the principles of computer architecture “client-server” which makes the code convenient in remote calculations, cloud technologies etc. A series of numerical experiments on simulating flows in porous hydrate-containing media was carried out. The advantages and problems of computing similar initial value problems by means of JS language and “client-server” architecture are discussed.

Key word: Gas hydrates, fluid dynamics, numerical modeling, JavaScript, Node.JS, client-server technology

Введение

Природные газовые гидраты относятся к тем видам традиционного углеводородного сырья, для добычи которого необходима разработка новых технологий. Этим обусловлены, в частности, фундаментальные исследования, направленные на изучение свойств газогидратов и связанных с ними флюидодинамических процессов [1, 2]. Газогидраты представляют собой соединения природного газа с водой (т.наз. клатратные соединения), которые в условиях достаточно низкой температуры и высокого давления находятся в твердом состоянии, похожем на лед рыхлой консистенции.

Повышение температуры или понижение давления вызывают смену фазового состояния газогидратов, при этом высвобождается весьма большое количество газа (в основном, метан) – 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].

Изучение газовых гидратов в природных и техногенных условиях относится в настоящее время к наиболее активно развивающимся областям исследований в нефтегазовой отрасли. К фундаментальным проблемам, которые необходимо решить при проектировании технологий разработки газогидратных месторождений, относятся проблемы снижения воздействия сопутствующих газовых выбросов на экологию, а также снижение рисков из-за быстроты фазовых переходов в гидрате при изменении внешних условий, что может приводить к резкому и весьма большому возрастанию объема газа, и, соответственно, создает риск разрушений оборудования [4].

Необходимость изучения процессов формирования гидратов обусловлена не только поисками новых источников сырья. Вместе с природным газом в газопровод по техническим причинам может попадать некоторое количество влаги. При температуре ниже точки росы эта влага конденсируется (ситуация, часто возникающая в подводных газопроводах), и взаимодействие ее с газом приводит к возникновению гидратных пробок. При подходящих условиях техногенные гидраты могут также формироваться в стволах скважин, в призабойной зоне. Эти явления в системах добычи и транспортировки природного газа негативно влияют на рабочие характеристики соответствующего оборудования. С другой стороны, в настоящее время изучаются перспективные варианты технологий, в которых газогидратам отводится центральная роль – в качестве средства компактного хранения добытого газа, средства накопления запасов энергии, опреснения морской воды, газоочистки и т.д. [4, 5].

Изучение газовых гидратов в природных и техногенных условиях относится к быстроразвивающимся областям науки и техники, в которых

существенная роль отводится вычислительным экспериментам с компьютерными моделями природных и техногенных процессов, сопровождающихся формированием и разложением гидратов. Компьютерное моделирование востребовано 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.

За последние 20-30 лет работа с компьютером стала привычной практикой едва ли не во всех сферах деятельности человека. До массового распространения графических дисплеев управление работой прикладного кода осуществлялось путем инструкций, вводимых через интерфейсы, построенные на том же языке, на котором был написан основной код, или консольные приложения, и т.п. Соответствующие примеры можно найти в [7], в руководствах по JCL – Job Control Language [8], и другой подобной литературе.

В настоящее время управляющие средства типа JCL вытесняются графическими пользовательскими интерфейсами в силу их очевидных преимуществ в отношении эргономики и повышения производительности труда.

Прикладной код «HYDRAT1D» разработан на основе вычислительной архитектуры «Клиент-сервер» [9]. Пользовательская часть (клиент) написана на языках JS, HyperText Markup Language (HTML) и на формальном языке Cascading Style Sheets (CSS) описывающем внешний вид интерфейса [10]. Язык для создания программных интерфейсов JavaScript (JS) встроен практически во все веб-браузеры, и потому доступен без особых ограничений на тип компьютера [10]. Часть, отвечающая за вычисления (сервер), написана на NodeJS [11]. 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, возможность разрабатывать оконные приложения для Linux, Windows и macOS.[11]

Базовая модель флюидодинамики

Математическая модель, описывающая динамику компонент смеси жидкостей в пористой среде, часть порового пространства которой занята газогидратными включениями, подробно обсуждается в работе [6]. Здесь мы приведем описание системы уравнений этой модели, решаемых численно кодом HYDRAT1.

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

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

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

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

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

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

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

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

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; ν – гидратонасыщенность; $S_v = 1 - \nu$ – 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_e – 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 уравнении (8) величины u и $\mathbf{X}$ – 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]

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

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:

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

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 неизотермической 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]:

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

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

$$\frac{\psi}{m\rho_v} = \left(\varphi - \frac{1}{\rho_v} \right) \geq 0, \quad \varphi = \frac{\beta_w}{\rho_w} + \frac{(1-\beta_w)}{\rho_g},$$

$$\delta_\varepsilon = \beta_w \varepsilon_w + (1-\beta_w) \varepsilon_g - \varepsilon_v \geq 0$$

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

Далее также используется обозначение: $()_p = \frac{\partial}{\partial P}$.

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:

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

$$+ \frac{\psi}{m\rho_v} \{ m \{ S_v [S_w \rho_w (\varepsilon_w)_p + (1 - S_w) \rho_g (\varepsilon_g)_p] + (1 - S_v) \rho_v (\varepsilon_v)_p \} +$$

$$+ [(1 - m) \rho_s \varepsilon_s]_p \}, \quad (15)$$

и перепишем уравнение (14) в более компактной форме

$$D_p \frac{\partial P}{\partial t} + \delta_\varepsilon DIG + \frac{\psi}{m\rho_v} DIG_\varepsilon = 0. \quad (16)$$

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.

Полагая растепленность $S_v=1$, из уравнений для трехфазной зоны получим двухфазное уравнение в талой зоне:

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

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

$$\frac{\partial}{\partial t} \{ m (S_w \rho_w \varepsilon_w + (1 - S_w) \rho_g \varepsilon_g) + (1 - m) \rho_s \varepsilon_s \} +$$

$$+ \frac{\partial}{\partial z} \{ \rho_w \varepsilon_w \mathbf{V}_w + \rho_g \varepsilon_g \mathbf{V}_g + [P(\mathbf{V}_w + \mathbf{V}_g)] \} + \frac{\partial}{\partial z} \mathbf{W} + q_\varepsilon = 0, \quad (19)$$

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

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

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

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

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

Особенности архитектуры «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.

Кратко о JS

JavaScript является объектно-ориентированным языком [12]. 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

[13]. Кроме того, JS поддерживает функциональное программирование: функции как объекты первого класса, объекты как списки, преобразование функций типа “carrying” (карринг, см. [14]), анонимные функции, замыкания — что придаёт языку дополнительную гибкость [14, 15].

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

- объекты с возможностью нтроспекции (C++);
- функции как объекты первого класса;
- автоматическое приведение типов;
- автоматическое управление памятью [15];
- анонимные функции.

Эти отличия делают JS более удобным для программной реализации в технологии «клиент-сервер».

Серверная платформа Node

Эволюция JS с каждым годом дает возможность для веб-разработчиков создавать большое количество новых технологий и инновационных приложений. Один из наиболее интересных и популярных инструментов для создания легко масштабируемых сетевых приложений является Node.js – это серверная реализация языка JS, основанная на разработанном компанией Google JS-движке V8, который был создан Райаном Далем (Ryan Dahl) [11]. Развитием проекта сейчас занимается компания Joyent, крупный провайдер облачных вычислений в США. Серверная среда Node.js состоит из 80% кода C/C++ (ядро).

Следует отметить, что этот язык не рассчитан для больших вычислительных задач, но он позволяет создавать короткие наглядные коды, и показателен в том смысле, что JS имеет однопоточную модель исполнения. Язык 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 нашей задаче [3] нечего распараллеливать, и одного потока предостаточно для достижения целей, и написать код, который будет выполняться быстро, не представляет труда. По производительности он будет сравним с C++ кодом. В чем секрет? Известно, что скорость работы — не главный критерий оценки языка. Нужно учитывать потребление памяти, нагрузку на CPU и т.д. Немаловажным и также являются еще и паттерны и подходы программирования. На сегодняшний день JS является самым быстрым интерпретируемым языком в мире. Конечно, это далеко некомпилируемый язык со статической типизацией, но эти интерпретаторы написанные под JS от Mozilla - SpiderMonkey, от Google - V8 и от Microsoft – Chakra создавались в

конкурентой борьбе лучших разработчиков. Следовательно, код написанный на JS, оптимизируется. Если в дальнейшем SpiderMonkey будет работать быстрее, чем V8, то, при использовании Node.JS, на новый движок не предоставит труда.

В заключение данного раздела отметим, что возможны и другие реализации прикладного кода, разработка которого была целью данной работы. Например, функциональные модули пишутся на C++, а далее импортируются в среду JS и далее работа с ними идет в рамках технологии “клиент-сервер”.

Краткое описание комплекса прикладных программ HYDRAT1D

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

В языке очень простая система загрузки модулей: между файлом и модулем есть однозначное соответствие. Для подключения модуля в приложение нужно в `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 со 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*:

Таблица 1

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

Метод *init* обращается к файлу *lib/init.js*, которая заполняет и местами переопределяет параметры задачи, которые хранятся в файле */lib/params/parameters.js* (

Таблица 2), начальными данными переданные из клиента (метод *init*), а затем запускает процесс вычисления (метод *start*). Файл *parameters.js* содержит в себе все параметры задачи, а также несколько открытых методов (Таблица 3), которые оперируют с параметрами задачи.

Таблица 2

Название параметров	Описание	Значение по умолчанию	Тип
P	Давление		array
P_Left	Давление на левой границе		string
P_Right	Давление на правой границе		string
Pkr	Критическое давление	4689300	number
SP	Давление на s+1 – ой итерации		array
Ps	Давление на s – ой итерации		array
DP	Приращение давления на соседних итерациях		array
P0	Давление в источнике	2E+6	number
Sv	Растепленность		array
SSv	Растепленность на s+1 – ой итерации		array
Svs	Растепленность на s – ой итерации		array
Svm	Величина близкая к единице (Δ_v)		number
Sw	Водонасыщенность		array
SSw	Водонасыщенность на s+1 – ой итерации		array
Sws	Водонасыщенность на s – ой итерации		array
Sw_min	Минимальное значение водонасыщенности для фазовой проницаемости	0.55	number
Sw_max	Максимальное значение водонасыщенности для фазовой проницаемости	0.9	number
T	Температура		array
Tkr	Критическая температура	191.202	number
T_init	Начальная температура		string
ST	Температура на s+1 – ой итерации		array
Ts	Температура на s – ой итерации		array
Q_g	Коэффициент в источнике газа	0	string
Q_w	Коэффициент в источнике воды	0	string
Q_e	Коэффициент при источнике в уравнении энергии	0	string
R	Универсальная газовая постоянная	8.31	number

M	Молярная масса газа	16E-3	number
A	Эмпирический коэффициент в уравнении фазового равновесия	7.28	number
B	Эмпирический коэффициент в уравнении фазового равновесия	169.7	number
Cp_g	Теплоемкость газа при постоянном давлении	2500	number
Cv_g	Теплоемкость газа при постоянном объеме	2081	number
Cv_w	Теплоемкость воды при постоянном объеме	4200	number
Cv_s	Теплоемкость скелета при постоянном объеме	873	number
rho_w	Плотность воды	1000	number
rho_h	Плотность гидрата	910	number
rho_s	Плотность скелета	2800	number
rho_g	Плотность газа	-	array
betta	Массовая доля воды в гидрате	0.9	number
porosity	Пористость	0.35	number
Krw	Относительная проницаемость воды	-	array
Krg	Относительная проницаемость газа	-	array
K_0	Коэффициент для абсолютной проницаемости	1E-14	number
myu_w	Вязкость воды	18E-4	string
myu_g	Вязкость газа	14E-6	string
lambda_w	Теплопроводность воды	561E-3	number
lambda_g	Теплопроводность газа	342E-4	number
lambda_h	Теплопроводность гидрата	2.11	number
lambda_s	Теплопроводность скелета	2	number
H	Теплота фазового перехода	514810	number
z	Коэффициент сверхсжимаемости	1	number
dt	Шаг по времени	0.001	number
ct	Показатель уменьшения или увеличения значения шага по времени. Используется в методах increase_dt и reduce_dt.	1.4	number
max_it	Максимальное число итераций	5	number
min_it	Минимальное число итераций	2	number
max_dt	Максимальное значение шага по времени	1E-20	number
min_dt	Минимальное значение шага по	0.1	number

	времени		
t_stop	Время остановки расчета	1	number
time	Массив времен расчета	0	array
time_steps	Массив шагов по времени		array
current_time	текущее время расчета	0	number
N	Число точек	100	number
from_N	Левая граничная точка фазы	0	number
to_N	Правая граничная точка фазы	N	number
L	Длина области	1	number
h	Шаг по пространству	1E-2	number
is_saturation_on	Статус сатурации	true	boolean
is_piezocon_on	Статус пьезопроводности	true	boolean
kappa_1	Индикатор типа граничного условия на левой границе	1	number
nu_1	Индикатор типа граничного условия на левой границе	0	number
kappa_2	Индикатор типа граничного условия на правой границе	1	number
nu_2	Индикатор типа граничного условия на правой границе	0	number
eT	Относительный скачок температуры при переходе от трехфазного к двухфазному	1E-9	number
G	индекс геометрии задачи	0	number
debug	флаг для отладки	false	boolean
methodname	Название метода	simple	string
deltaE	Доля энергии накопленная после перехода в талое состояние	0.01	number

Таблица 3

Название	Описание	Доступ
is_calculation_completed	Возвращает истину, если расчеты завершены	Открытый
increase_time	Увеличивает время на шаг по времени	Открытый
increase_dt	Увеличивает шаг по времени в ct (Таблица 2) раз.	Открытый
reduce_dt	Уменьшает шаг по времени в ct (Таблица 2) раз.	Открытый
save_time_steps	Сохраняет шаг по времени в каждом временном слое	Открытый

save_calculation	Сохраняет вычисленные параметры (Sv,Sw,T,P). Вызывается после метода <i>run_calculation</i>	Открытый
output	Возвращает объект с выходными параметрами. К этому методу обращается calculation.result(). В этом методе можно добавлять дополнительные переменные в качестве свойства объекта, которые могут понадобиться пользователю.	Открытый
scaling	Масштабирование параметров	Открытый
define_edges	Принимает фазу задачи и вычисляет точки границ фаз.*	Открытый

Метод start вызывает приватную функцию *run_calculation*, он в свою очередь первым делом обращается к файлу lib/saturation/core/phase.js (Таблица 4), чтобы тот смог определить начальные фазы задачи.

Алгоритм расчета фазового состояния флюида

Определение фазового состава является ключевым пунктом работы 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 -ом слое. Получаем некий контейнер-массив с объектами фаз. Например [{тип 1-ой фазы, левая граница 1-ой фазы, правая граница 1-ой фазы}, ..., {тип n -ой фазы, левая граница n -ой фазы, правая граница n -ой фазы}] (23)

```
var container = [{type:'triple', from:0, to:0.4}, {type:'melted', from:0.4, to:0.6}, {type:'triple', from:0.6, to:1}] (23)
```

Далее выполняется цикл по массиву и для каждого типа фаз вычисляем граничные точки (define_edges), а потом по типу фазы выбираем соответствующие методы дальнейших расчетов.

Таблица 4

Название	Описание	Доступ
define	Начальное определение фаз. Вызывается 1 раз в начальный момент времени.	Открытый
define_if_triple	Метод вызывается только в трехфазной зоне для вставки в контейнер новых фаз если они появились	Открытый

define_if_melted	Метод вызывается только в талой зоне для вставки в контейнер новых фаз если они появились	Открытый
add	Добавляет в контейнер новые фазы	Открытый
reset	Вычищает контейнер	Открытый
combine	Вспомогательная функция для конструкции фаз	Закрытый

Далее в методе начинает работать цикл *while*, ожидающий конец расчета. В теле оператора вызывается метод *run* модуля *iteration-process.js*, который наследует методы объекта *lib/math/iteration.js* и выполняет основные вычисления, а затем последовательно выполняется процедура *save_calculation*, описанная в Таблица 3.

Более низкоуровневые вычисления для блоков пьезопроводности и сатурации лежат в соответствующих папках *./lib/piezocon/* и *./lib/saturation/*.

Все функции определены в файле *functions.js*, при необходимости новые функции можно добавить в этот файл.

Таблица 5

Название функций	Описание
init	Давление
K	Абсолютная проницаемость
Krw	Относительная проницаемость воды
Krg	Относительная проницаемость газа
myu_w	Вязкость воды
myu_g	Вязкость газа
T	Температура
TDp	Производная температуры по давлению
psi	Скачок удельного объема (на единицу массы)
E_s	Внутренняя энергия скелета
E_sDp	Производная внутренней энергии скелета по давлению
E_w	Внутренняя энергия воды
E_wDp	Производная внутренней энергии воды по давлению
E_g	Внутренняя энергия газа
E_gDp	Производная внутренней энергии газа по давлению
E_v	Внутренняя энергия гидрата

E_{vDp}	Производная внутренней энергии гидрата по давлению
Dp	Коэффициент бароемкости
ΔE_p	Скачок удельной внутренней энергии (на единицу массы)
m	пористость
mDp	Производная пористости
z	Коэффициент сверхсжимаемости
zDp	Производная коэффициента сверхсжимаемости по давлению
zDT	Производная коэффициента сверхсжимаемости по температуре
ρ_g	Плотность газа
ρDp_g	Производная плотности газа по давлению
ρDp_g	Производная обратной величины плотности газа по давлению
ρ_w	Плотность воды
ρDp_w	Производная плотности воды
ρ_v	Плотность гидрата
ρDp_v	Производная плотности гидрата
ρ_s	Плотность скелета
λ	Эффективная теплопроводность
Q_w	Источник воды
Q_g	Источник газа
Q_e	Источник энергии
$CCLg$	Условие характеристической сшивки для газа
$CCLw$	Условие характеристической сшивки для воды

Работа с кодом HYDRAT1D

Мы отмечали, что для использования комплекса программ 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 вы можете выполнять обычный JS код.

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

(14)

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.

Чтобы запустить программу, нужно в корневой директории программы набрать команду `npm install`, которая установит все нужные модули, а далее `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. 1.

Рис. 1. Архитектура программного комплекса.

Вычислительную часть демонстрирует следующая блок-схема взаимодействия функциональных компонент кода (рис. 2):

Рис. 2. Блок схема для решения связной задачи

Пример расчетов

Рассматривается следующий пространственно-одномерный процесс на отрезке $x \in [0, l]$, l – длина расчетной области. В нем в начальный момент времени давление $P(x, 0) = 2 \times 10^7$ Па, водонасыщенность $S_w(x, 0) = S_w^*$ и растепленность $S_v(x, 0) = S_v^*$ однородны по пространству, $0 < S_w^* < 1$ и $0 < S_v^* < 1$ – постоянные величины. Ускорение свободного падения не учитывается ($g = 0$).

Границы расчетной области предполагаются непроницаемыми твердыми «стенками», т.е. поток через них нулевой:

$$\mathbf{V}_w|_{x=0}=0, \mathbf{V}_g|_{x=0}=0, \mathbf{V}_w|_{x=l}=0, \mathbf{V}_g|_{x=l}=0, t>0.$$

Предполагается, что в некоторой узкой пространственной области работает источник:

$$q_w = \alpha(P - P^*), \quad q_g = \beta(P - P^*), \quad q_\varepsilon = \varepsilon_w q_w + \varepsilon_g q_g,$$

где P^* – постоянная величина, α и β – характеристики источника (стока) в призабойной зоне, обусловленные перепадом давления внутри скважины и в пласте. Всюду в дальнейшем будем считать $\alpha = 0$, $\beta = 0.00001 \frac{c}{M^2}$. При этом

размерность давления $[P] = \text{Па}$. Источник действует только в пространственной области $x \in [0.4, 0.6] \text{ м}$.

Для расчета были выбраны следующие значения параметров, характерные для Мессояхского газогидратного месторождения [16, 17].

$$\rho_w = 10^3 \frac{\text{кг}}{\text{м}^3}, \rho_v = 910 \frac{\text{кг}}{\text{м}^3}, \rho_s = 2800 \frac{\text{кг}}{\text{м}^3}, \beta_w = 0.9, A = 7.28 \text{ К}, B = 169.7 \text{ К},$$

$$\mu_w = 10^{-3} \text{ Па} \cdot \text{с}, \mu_g = 0.014 \cdot 10^{-3} \text{ Па} \cdot \text{с}, c_w = 4165 \frac{\text{Дж}}{\text{кг} \cdot \text{К}}, c_g = 2500 \frac{\text{Дж}}{\text{кг} \cdot \text{К}},$$

$$c_s = 873 \frac{\text{Дж}}{\text{кг} \cdot \text{К}}, M = 0.016 \frac{\text{кг}}{\text{моль}}, h = 514810 \frac{\text{Дж}}{\text{кг}}, R = \frac{8.31}{M} \frac{\text{Дж}}{\text{моль} \cdot \text{К}}, S_w^* = 0.6,$$

$$S_v^* = 0.8, k(S_v) = k_0 \cdot (S_v)^3, k_0 = 10 \text{ мД} = 10^{-14} \text{ м}^2, P^* = 2 \text{ МПа}, m = 0.35,$$

$$k_{rw}(S_w) = 1.477S_w^5 - 1.587S_w^6 + 1.11S_w^7 - 0.0473, k_{rg}(S_w) = 1.044 - 1.7S_w + 0.6S_w^2.$$

Минимальное значение водонасыщенности $S_{w\min} = 0.55$.

$$k_{rw}(S_w) = 0, k_{rg}(S_w) = k_{rg}(S_{w\min}) \text{ при } S_w \leq S_{w\min}.$$

Максимальное значение водонасыщенности $S_{w\max} = 0.9$.

$$k_{rw}(S_w) = k_{rw}(S_{w\max}), k_{rg}(S_w) = 0 \text{ при } S_w \geq S_{w\max}.$$

Длина модельной трубы полагается равной $l = 1 \text{ м}$, шаг по пространственной координате $h = 0.01 \text{ м}$. Расчеты приводятся для моментов времени $t = 1, 10, 100 \text{ с}$. На рис.3 представлены результаты расчетов по схеме со свободно-объемной аппроксимацией $\alpha = 0, \beta = 0.00001$ при $x \in [0.4; 0.6]$. Можно увидеть, что в расчетах реализуется совместный трансфазный пространственно расширяющийся внутри области действия отбора газа $[0.4; 0.6]$ процесс депрессии давления в возникающей талой зоне и в остальной пьезопроводной среде коллектора с газогидратными включениями. Такой трансфазный расчет проводится впервые.

Рис. 3. Распределение растепленности (а), водонасыщенности (б), давления (в) и температуры (г) для моментов времени 1, 10, 100 с. Трехфазная зона – штрих-пунктир, талая зона – сплошная линия.

Заключение

Разработан пакет прикладных программ HYDRAT1D для решения связанных задач динамики флюидов в коллекторе, со сквозным расчетом течения в талой зоне и пьезопроводной среде с газогидратными включениями. Код написан на языке программирования JavaScript, на основе платформы Node.js. В статье кратко описаны математическая модель, численная схема и алгоритм решения связанной задачи. Приведено описание их программной реализации: используемые технологии программирования, программные модули, функции и параметры, а также некоторые сведения о «клиент-серверной» архитектуре применительно к средствам, использованным для организации работы кода HYDRAT1D. Реализованный комплекс программ использован для компьютерного моделирования трансфазной задачи. Выполненные разработки создают основу для реализации дальнейших проектов прикладного программного обеспечения на базе JS. Таким программным обеспечением специалисты в нефтегазовой отрасли могут пользоваться в удаленном доступе при умеренном уровне навыков в информационных технологиях и практике вычислений.

Литература

- [1] Makogon Yu.F. Hydrates of hydrocarbons, Tulsa: Pennwell Publishing C., 1997.
- [2] Истомин В. А., Якушев В. С. Газовые гидраты в природных условиях. — М.: Недра, 1992. — 236 с. — ISBN 5-247-02442-7.

- [3] Кузнецов Ф. А., Истомин В. А., Родионова Т. В. Газовые гидраты: исторический экскурс, современное состояние, перспективы исследований. // Российский химический журнал, т. 48, № 3, 2003, с. 5-18.
- [4] Д. Колесов, И. Ронова, А. Талис, И. Яминский, Г. Терещенко. Газовые гидраты – объект исследования и применения (рус.) // Наноиндустрия : журнал. — 2010. — № 1. — С. 20-23.
- [5] Шагапов В. Ш., Мусакаев Н. Г., Хасанов М. К. Нагнетание газа в пористый резервуар, насыщенный газом и водой. // Теплофизика и аэромеханика, т.12, № 4, 2005, с. 645—656.
- [6] П.И. Рагимли. Моделирование процессов совместной фильтрации в талой зоне и пьезопроводной среде с газогидратными включениями / П.И. Рагимли [и др.] // Препринты ИПМ им. М.В.Келдыша. 2018. № 40. 32 с. doi:10.20948/prepr-2018-40.
URL: <http://library.keldysh.ru/preprint.asp?id=2018-40>
- [7] Джермейн К.Б. Программирование на IBM/360. (Programming the IBM/360, 1967) [Djv-12.2М] Перевод с английского под редакцией В.С. Штаркмана. Издание третье, стереотипное. (Москва: Издательство «Мир». Редакция литературы по математическим наукам, 1978).
- [8] <http://www.intuit.ru/studies/courses/85/85/lecture/28277?page=8>
- [9] <https://www.intuit.ru/studies/courses/508/364/lecture/8643?page=2>
- [10] Дженнифер Н. Роббинс. HTML5, CSS3 и JavaScript. Исчерпывающее руководство = Learning Web Design, 4th Edition. — М.: Эксмо, 2014. — 528 с. — ISBN 978-5-699-67603-3.
- [11] Долгов А. Н., Нуруллин Р. Ю., Программная платформа Node.js, Достижения науки и образования, 2016, № 12, С.17-18
- [12] McKenzie N., Is JavaScript is object-oriented language?, Dr. Dobb's Journal: Software tools for the professional Programmer. 2001, Т. 26, № 8. С. 115-116.
- [13] K. Simpsons, You Don't Know JS: this & Object Prototypes, O'Reilly Media; 1edition,p.174, 2014
- [14] Бенджамин Пирс. Типы в языках программирования / Пер. с англ.: Г. Бронников, А. Отт. — Добросвет, 2011. — С. 76. — 656 с,
- [15] Ilyushin E., Namiot D., On memory management in Javascript applications, International jo urnal of open information technologies. 2015. Т. 3. № 10, С.11-15.
- [16] Р.С. Богатыренко. Особенности разработки и эксплуатации газогидратных месторождений (на примере Мессояхского месторождения). — М.: Диссертация кандидата технических наук, 1979, 159 с.
- [17] А.Л. Сухоносенко. Термогидродинамическое моделирование процессов разработки газогидратных месторождений. — М.: Диссертация кандидата технических наук, 2013, 145 с.

Оглавление

Введение	3
Базовая модель флюидодинамики	5
Особенности архитектуры «Клиент — сервер»	9
Кратко о JS	9
Серверная платформа Node	10
Краткое описание комплекса прикладных программ HYDRAT1D	11
Алгоритм расчета фазового состояния флюида	16
Работа с кодом HYDRAT1D	18
Пример расчетов	20
Заключение	22
Литература	22