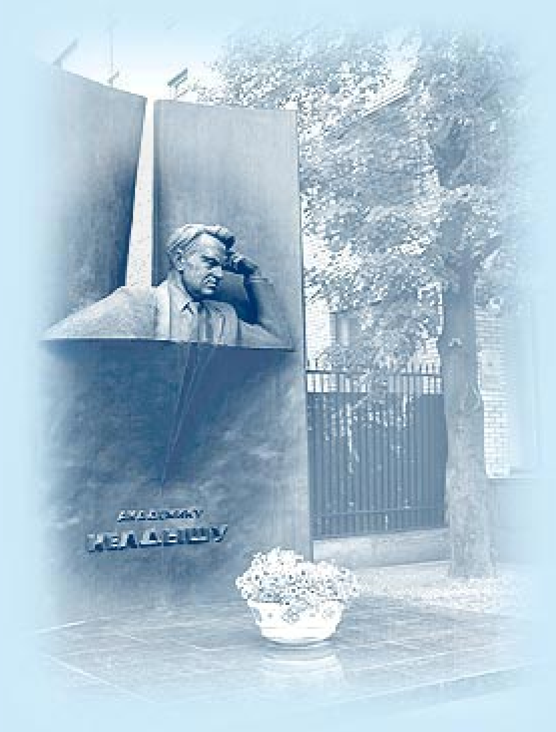
ИПМ им.М.В.Келдыша РАН • Электронная библиотека

Препринты ИПМ • Препринт № 28 за 2023 г.



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

А.В. Иванов

О реализации модели мелкой воды на базе квазигазодинамического подхода в открытом программном комплексе OpenFOAM

Рекомендуемая форма библиографической ссылки: Иванов А.В. О реализации модели мелкой воды на базе квазигазодинамического подхода в открытом программном комплексе OpenFOAM // Препринты ИПМ им. М.В.Келдыша. 2023. № 28. 27 с.
https://doi.org/10.20948/prepr-2023-28
https://library.keldysh.ru/preprint.asp?id=2023-28

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

А.В. Иванов

О реализации модели мелкой воды на базе квазигазодинамического подхода в открытом программном комплексе OpenFOAM

Москва — 2023

Иванов А.В.

О реализации модели мелкой воды на базе квазигазодинамического подхода в открытом программном комплексе OpenFOAM

В данной работе описан новый решатель RSWEFoam для моделирования течений в приближении мелкой воды, который реализован на базе открытой интегрируемой платформы OpenFOAM версии 2012 с использованием метода контрольных объемов. Описаны модель регуляризованных уравнений мелкой воды, лежащая в основе решателя, а также особенности ее реализации. RSWEFoam интегрирован в открытый фреймворк для моделирования задач гидро- и газодинамики в рамках КГД/КГиД подхода QGDSolver. Приведены результаты численного моделирования с применением решателя.

Ключевые слова: уравнения мелкой воды, открытый программный комплекс OpenFOAM, решатель RSWEFoam, квазигазодинамический подход, метод конечных объемов.

Aleksandr Vladimirovich Ivanov

On the implementation of the shallow water model based on the quasi-gasdynamic approach in the open-source software package OpenFOAM

This paper presents RSWEFoam – a new solver for shallow water flow simulation, which is implemented based on the OpenFOAM v2012 open integrable platform using the finite volume method. The model of regularized shallow water equations underlying the solver is described, as well as the features of its implementation. RSWEFoam is integrated into an open-source framework QGDSolver for fluid and gas dynamics problems modeling within the QGD/QGD approach. The results of numerical simulation using a solver are presented.

Key words: shallow water equations, OpenFOAM, RSWEFoam solver, quasi-gas dynamic approach, finite volume method.

Введение

Одной из ключевых проблем проведения современных вычислительных экспериментов является трудность выбора инструмента среди многообразия программ и библиотек цифрового моделирования. Многие научные коллективы разрабатывают самописные программы и индивидуальные вычислительные пакеты для решения конкретных прикладных задач. Такое программное обеспечение (ПО) заточено под определенную проблему и отлично оптимизировано для нее. К сожалению, подобные самописные пакеты и программы чаще всего не развиваются далее, вследствие чего их становится сложно использовать в сторонних задачах или на более новых платформах. И если прежде даже для небольших проектов главной трудностью была ограниченность вычислительных ресурсов, что оправдывало разработку уникальных алгоритмов для каждой задачи и даже для отдельной вычислительной машины, то теперь в этом нет практически никакой необходимости. Более того, открывается возможность не создавать с нуля основу для проведения численного моделирования, т.е. разрабатывать модули для визуализации, построения сеток, модуль параллельных вычислений и т.д., а воспользоваться уже существующими комплексами и библиотеками, полностью сосредоточившись на реализации непосредственно метода моделирования.

Существует множество платформ, позволяющих разрабатывать и применять новые модели для решения широкого спектра задач. Ярким примером является OpenFOAM [1] (англ. Open Source Field Operation And Manipulation) – открытая интегрируемая платформа для численного моделирования задач механики сплошных сред в рамках метода конечного объема. Практика использования открытых программных комплексов широко распространена. Их главными преимуществами являются доступность, открытость, актуальность, удобство, возможность доработки под конкретные задачи и добавления новых методов и моделей без вмешательства создателей комплекса. OpenFOAM как программный комплекс направлен на решение задач механики сплошных сред, предлагает базовые решатели, а также возможность реализации на их основе собственных библиотек и моделей.

В данной работе рассматривается RSWEFoam – реализованный в открытом программном комплексе OpenFOAM решатель для модели мелкой воды, в основе которого лежат регуляризованные уравнения. Решатель построен на базе фреймворка QGDSolver [2], разработанного лабораторией свободного программного обеспечения цифрового моделирования технических систем (СПО ЦМТС) Института системного программирования (ИСП) РАН им. В.П. Иванникова.

Приближение мелкой воды представляет собой переход от трехмерных течений к плоским двумерным, при котором вертикальной компонентой скорости течения можно пренебречь по сравнению с горизонтальными компонентами, значения которых осредняются по всей глубине. Это допустимо при условии, что глубина слоя жидкости достаточно мала по сравнению с характерными размерами задачи, например с горизонтальными масштабами [3].

С практической точки зрения такая модель интересна при исследованиях, в которых можно пренебречь стратификацией, например для моделирования озер и искусственных водоемов [4]; для изучения явлений с преобладанием горизонтальных эффектов над вертикальными, например, таких как приливные колебания или распространение цунами [5]; моделирования колебаний жидкости в резервуарах, например углеводородов в баке топливного танкера [6].

Поскольку уравнения мелкой воды тесно связаны с уравнениями газовой динамики, для численного решения был предложен способ регуляризации уравнений по аналогии с построением квазигазодинамической системы [7]. Такую систему называют системой регуляризованных уравнений мелкой воды (РУМВ).

По РУМВ было опубликовано много работ, например [6–10], в которых продемонстрированы результаты моделирования как тестовых, так и прикладных задач. В частности, в [7] описано исследование алгоритма на общепринятых бенчмарках, таких как задача о распаде разрыва, а в работе [9] приведено моделирование сейшевых колебаний Азовского моря, возникающих под действием перепада атмосферного давления и ветра, и сравнение результатов с реальными данными. Однако все расчеты производились с помощью индивидуальных кодов, и для того, чтобы повторить их, потребуется значительное время. Кроме того, при смене задачи может появиться необходимость в доработке программ ввиду изменения начальных и граничных условий, их формата, типа, конфигурации и других параметров задачи. К сожалению, такое положение вещей значительно усложняет, а порой и делает практически невозможным использование модели РУМВ сторонними исследователями и научными коллективами.

Основная задача данной работы – построить новый эффективный решатель на базе открытого программного комплекса OpenFOAM, добавленный в открытый фреймворк QGDSolver и доступный широкому кругу пользователей, описать его структуру, специфику работы с ним и способ реализации в OpenFOAM, показать примеры его использования на известных модельных задачах.

1. Регуляризованные уравнения мелкой воды

Система уравнений

Рассмотрим некоторую область, в которой располагается вода, ограниченная береговой линией, рис. 1. Разместим систему координат таким образом, чтобы плоскость OXY была параллельна поверхности воды в ее спокойном состоянии, а наиболее глубокая точка водоема лежала на плоскости $z = 0$. В таком случае удобно ввести неотрицательную функцию батиметрии $b(x, y)$, описывающую поверхность дна исследуемого водоема, а также неотрицательную функцию толщины слоя воды $h(x, y, t)$, которая зависит от времени и отсчитывается от дна, рис. 1. При этом можно выделить толщину слоя при спокойной воде $H(x, y)$, а также амплитуду малых возмущений $\eta(x, y, t)$, так что $h(x, y, t) = H(x, y) + \eta(x, y, t)$. Вектор горизонтальной скорости обозначим $\mathbf{u} = \{u_x(x, y, t), u_y(x, y, t)\}$, модуль ускорения силы тяжести $g = |\mathbf{g}|$. Кроме того, для описания свободной поверхности воды удобно ввести функцию $\xi(x, y, t) = h(x, y, t) + b(x, y)$, рис. 1.

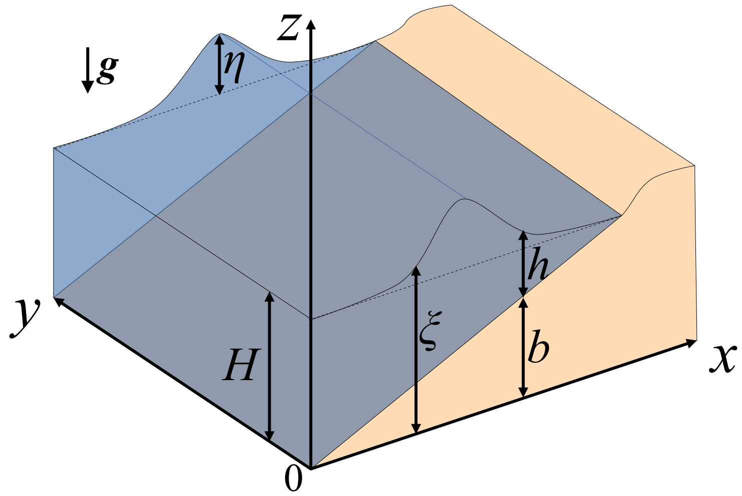


Рис. 1. Обозначения мелкой воды

Для введенных обозначений система РУМВ имеет вид, см. [7, 8]:

$$\frac{\partial h}{\partial t} + \nabla \cdot \mathbf{j_m} = 0, \tag{1}$$

$$\frac{\partial (h\mathbf{u})}{\partial t} + \nabla \cdot (\mathbf{j}_m \otimes \mathbf{u}) + \nabla \frac{gh^2}{2} = h^* (\mathbf{f}^v - g\nabla b) + \mathbf{f}^s + \nabla \cdot \mathbf{\Pi}, \tag{2}$$

где

$$h^* = h - \tau \nabla \cdot (h\mathbf{u}), \tag{3}$$

$$\mathbf{j}_m = h (\mathbf{u} - \mathbf{w}), \tag{4}$$

$$\mathbf{w} = \frac{\tau}{h} \left[\nabla \cdot (h\mathbf{u} \otimes \mathbf{u}) + gh\nabla (b + h) - h\mathbf{f}^v - \mathbf{f}^s\right], \tag{5}$$

$$\mathbf{\Pi} = \mathbf{\Pi}_{NS} + \tau \mathbf{u} \otimes \left[h (\mathbf{u} \cdot \nabla) \mathbf{u} + gh\nabla (b + h) - h\mathbf{f}^v - \mathbf{f}^s\right] + \tau \mathbf{I} \left[gh\nabla \cdot (h\mathbf{u})\right], \tag{6}$$

$\mathbf{f}^v(x, y, t)$ и $\mathbf{f}^s(x, y, t)$ – вектора объемной и поверхностной внешних сил соответственно, $\mathbf{I}$ – диагональная единичная матрица, а $\otimes$ – символ тензорного произведения, который можно расписать как

$$\nabla \cdot (h\mathbf{u} \otimes \mathbf{u}) = h\mathbf{u}(\nabla \cdot \mathbf{u}) + (h\mathbf{u} \cdot \nabla) \mathbf{u} + \mathbf{u} (\mathbf{u} \cdot \nabla h).$$

$\mathbf{\Pi}_{NS}$ – тензор вязких напряжений Навье–Стокса, который при необходимости в ряде задач рассматривается как дополнительный регуляризатор и может быть включен или отброшен, см., например, [7, 8]. Коэффициент кинематической вязкости жидкости μ считается искусственным и вычисляется через параметр τ:

$$\mathbf{\Pi}_{NS} = \mu\frac{h}{2}\left[(\nabla\otimes\mathbf{u}) + (\nabla\otimes\mathbf{u})^{\mathrm{T}}\right], \quad \mu = \tau g h. \tag{7}$$

Свойства системы РУМВ (1)–(6) широко изучены и описаны в научных публикациях, например в работах [8, 11, 12]. Отличие РУМВ от системы классических уравнений мелкой воды заключается в наличии дополнительных слагаемых, определяемых уравнениями (3)–(6), которые, как показано в [11, 12], носят диссипативный характер. Эти слагаемые вносят искусственную вязкость в систему, что значительно упрощает численное решение РУМВ явными методами с аппроксимацией пространственных производных центральными разностями. Степень вязкости определяется параметром τ, который также называется параметром регуляризации и подробнее будет описан в следующем разделе. При $\tau \to 0$ система РУМВ (1)–(6) переходит в систему УМВ.

Метод численного решения

Для численного решения системы (1)–(6) используется явный по времени интегро-интерполяционный метод, который применительно к РУМВ можно назвать методом конечных площадей.

В следующем разделе будет разобран более общий случай, а пока для простоты рассмотрим аппроксимацию уравнений на равномерной прямоугольной пространственной сетке, элемент конечной площади такой сетки представлен на рис. 2, где $\Delta x = \Delta y = d$. Кроме того, положим $\mathbf{f}^v = \mathbf{0}$ и $\mathbf{f}^s = \mathbf{0}$. Основываясь на этом, приведем явную по времени схему решения уравнений (1)–(6).

Для сеточных значений величин введем дополнительные индексы. Верхним индексом $\widehat{\ }$ обозначим величины, взятые на следующем слое по времени, т.е. в момент $t + \Delta t$, а для величин на текущем слое по времени будем использовать прежние обозначения. Нижним индексом при необходимости будем обозначать принадлежность величины ячейке (если используется обозначение заглавной латинской буквой, обозначающей центр ячейки, например P), либо грани/ребру (если используется строчная латинская буква, например f). Например, сеточное значение толщины слоя воды, взятое на следующем слое по времени в точке $M(x, y)$, будет записано как $h(M, t = t + \Delta t) = \widehat{h}_M$.

Рассмотрим элемент конечной площади S_P, центр которого находится в точке P, а одно из ребер e соединяет его с элементом конечной площади S_N с центром в точке N соответственно, рис. 2. Тогда обозначим вектор, соединяющий центры элементов как $\mathbf{d}$, а $\mathbf{L}_e$ – нормаль к ребру e, причем длина нормали совпадает с длиной ребра e: $|\mathbf{L}_e| = e$. В случае прямоугольной

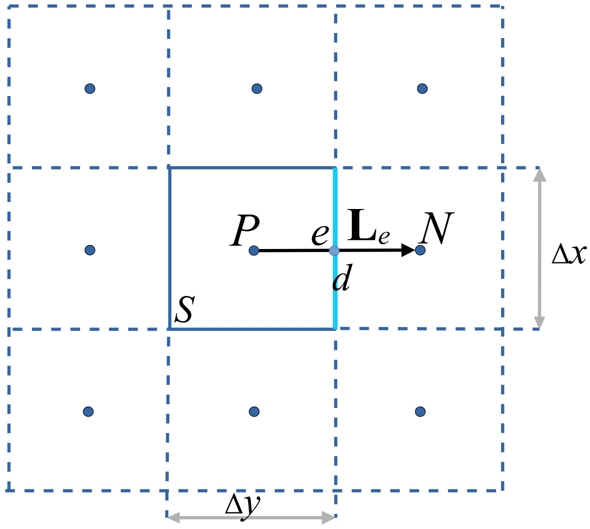


Рис. 2. Схематическое изображение элемента конечной площади на прямоугольной равномерной сетке.

равномерной пространственной сетки $\mathbf{d}_e = \mathbf{L}_e$, поэтому $|\mathbf{L}_e| = |\mathbf{d}_e| = d$, рис. 3.

Выпишем в потоковом представлении систему (1)–(6) :

$$\widehat{h}_P = h_P - \frac{\Delta t}{S_P} \sum_e \phi_e, \tag{8}$$

$$\widehat{(h\mathbf{u})}_P = (h\mathbf{u})_P - \frac{\Delta t}{S_P} \sum_e \mathbf{G}_e, \tag{9}$$

где ϕ_e – поток массы, $\mathbf{G}_e$ – поток импульса через ребро e, а суммирование происходит по потокам через все ребра e элемента S_P.

Запишем выражения для потоков ϕ_e и $\mathbf{G}_e$:

$$\phi_e = (h\mathbf{u})_e \cdot \mathbf{L}_e - \tau_e \left([\nabla \cdot (h\mathbf{u} \otimes \mathbf{u})]_e + gh_e \left[\nabla (b+h)\right]_e \right) \cdot \mathbf{L}_e, \tag{10}$$

$$\mathbf{G}_e = \Phi_e \mathbf{u}_e + \frac{g\,(h_e)^2}{2} \mathbf{L}_e - \mathbf{L}_e \cdot \mathbf{\Pi}_e + g\,(h_e - \tau_e [\nabla \cdot (h\mathbf{u})\,]_e)\, b_e \mathbf{L}_e, \tag{11}$$

$$\begin{aligned} \mathbf{\Pi}_e = \mathbf{\Pi}_{NSe} + \tau_e \mathbf{u}_e \otimes \left((h\mathbf{u})_e \cdot [\nabla \mathbf{u}]_e + gh_e \left[\nabla (b+h)\right]_e \right) + \\ + \tau_e \mathbf{I} \left(gh_e \left[\nabla \cdot (h\mathbf{u})\right]_e \right). \end{aligned} \tag{12}$$

$$\mathbf{\Pi}_{NSe} = \tau_e g \frac{h_e^2}{2} \left[(\nabla \otimes \mathbf{u}) + (\nabla \otimes \mathbf{u})^{\mathrm{T}} \right]_e. \tag{13}$$

Нижний индекс e обозначает величины, значения которых взяты на ребре e, а скобки $[\cdot]_e$ говорят о том, что оператор, заключенный в скобках, берется в центре ребра e. При этом аппроксимация производной по направлению для некоторой величины $\mathbf{U}$ записывается в виде

$$\mathbf{U}_e = \mathbf{U}_P + \frac{\mathbf{U}_N - \mathbf{U}_P}{d}, \quad \frac{\partial \mathbf{U}}{\partial \mathbf{L}_e} = \frac{\mathbf{U}_N - \mathbf{U}_P}{d}. \tag{14}$$

Устойчивость описываемого метода связана со слагаемыми с коэффициентом τ, который имеет вид

$$\tau = \alpha \frac{l}{c}, \tag{15}$$

где l – характерный размер пространственной ячейки, который используется в численном алгоритме, например, в виде $l = \sqrt{s} = d$, где s – площадь ячейки, $c = \sqrt{gh}$ – скорость распространения длинной волны, α – численный коэффициент, выбираемый из условий точности и устойчивости счета. Как правило, $0 < \alpha < 1$, и в качестве базового значения можно выбирать $\alpha = 0.5$. При правильном подборе α параметр τ соотносится со временем, необходимым малому возмущению для преодоления площади пространственной ячейки. Однако параметр τ можно модифицировать. Поскольку $c = \sqrt{gh}$ выражает скорость распространения длинных волн, то можно расширить область применения алгоритма на случаи, когда характерная скорость будет значительно больше величины c. В таком случае в качестве характерной скорости распространения возьмем

$$c = \sqrt{gh} + |\mathbf{u}|,$$

тогда

$$\tau = \alpha \frac{l}{\sqrt{gh} + |\mathbf{u}|}. \tag{16}$$

Условие устойчивости имеет вид условия Куранта, подробнее см. в [8], где шаг по времени выбирается по формуле

$$\Delta t = \beta \left(\frac{\Delta x + \Delta y}{2c} \right)_{min}, \tag{17}$$

число Куранта $0 < \beta < 1$ зависит от величины параметра регуляризации τ в виде $\beta = \beta(\alpha)$ и подбирается в процессе вычислений для обеспечения монотонности численного решения.

2. Реализация алгоритма в OpenFOAM

Аппроксимация регуляризованных уравнений

В данной работе используется OpenFOAM v2012, [1]. Платформа OpenFOAM предоставляет возможность использования двух численных методов моделирования:

- класс методов fvm (англ. finite volume method) – метод конечного объема, наиболее развит и распространен в рамках платформы;
- класс методов fam (англ. finite area method) – метод конечных площадей, также используется во многих решателях, но менее популярен.

Поскольку большая часть решателей разработана с использованием fvm-методов и стандартизирована для них, решатель RSWEFoam также реализован с использованием метода конечных объемов, поэтому опишем процесс перехода от плоской задачи к пространственной постановке.

Предположим, что исследуемая область расчета покрыта некоторой пространственной неструктурированной сеткой, рис. 3. В качестве примера рассмотрим сетку, состоящую из пятиугольников. Для перехода от конечно-площадного к конечно-объемному подходу необходимо растянуть двумерную пространственную сетку, расположенную в плоскости XOY рис. 3 в перпендикулярном плоскости XOY направлении, т.е. вдоль оси z, рис. 4. При таком переходе толщина сетки постоянна, обозначим ее как Δz, площади элементов S_P заменяются на их объемы $V_P = S_P \Delta z$, центры ячеек смещаются на $\Delta z/2$ вдоль оси z, а нормаль $\mathbf{L}_e$ к ребру e переходит в нормаль $\mathbf{S}_f$ к грани f, длина которой совпадает с площадью грани, рис. 4.

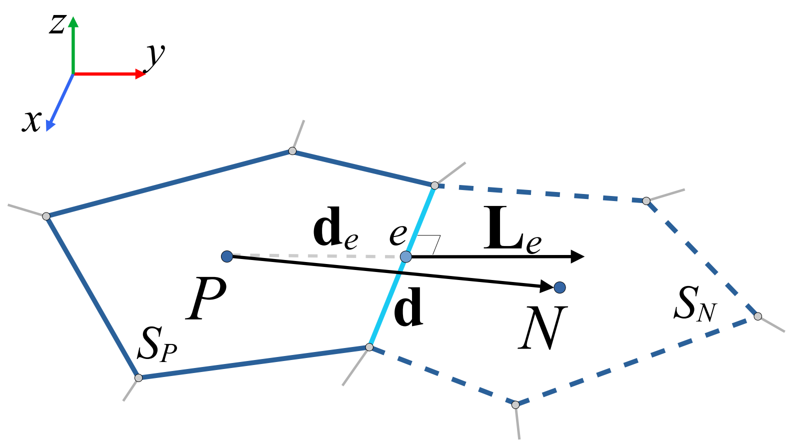


Рис. 3. Схематическое изображение элемента конечной площади.

Рассмотрим элемент конечного объема V_P, центр которого находится в точке P, а одна из граней f соединяет его с элементом конечного объема V_N с центром в точке N соответственно, рис. 4. Тогда обозначим вектор, соединяющий центры элементов как $\mathbf{d}$, а нормаль к грани f – $\mathbf{S}_f$, причем длина нормали совпадает с площадью грани f. Запишем для элемента V_P аппроксимацию уравнений (1)–(2) в потоковом представлении с использованием метода конечных объемов, где аналогично положим $\mathbf{f}^v = \mathbf{0}$ и $\mathbf{f}^s = \mathbf{0}$:

$$\widehat{h}_P = h_P - \frac{\Delta t}{V_P} \sum_f \Phi_f, \tag{18}$$

$$(\widehat{h\mathbf{u}})_P = (h\mathbf{u})_P - \frac{\Delta t}{V_P} \sum_f \mathbf{F}_f, \tag{19}$$

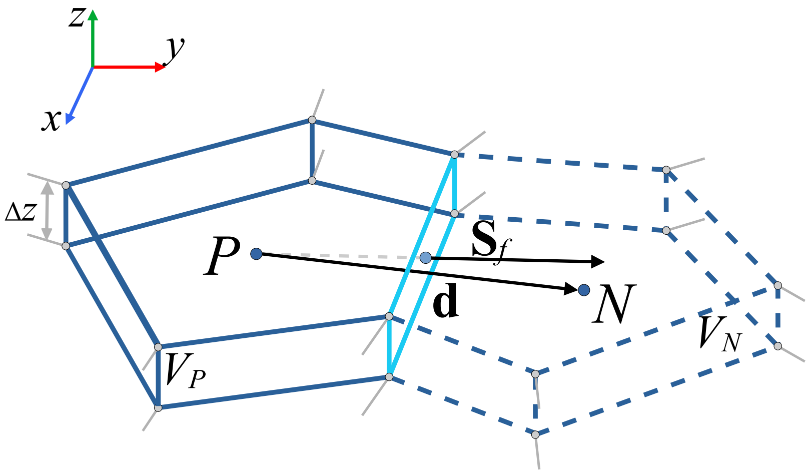


Рис. 4. Схематическое изображение элемента конечного объема.

где Φ_f – поток массы, а $\mathbf{F}_f$ – поток импульса через грань f, а суммирование происходит по потокам через все грани конечного объема V_P.

Запишем выражения для потоков Φ_f и $\mathbf{F}_f$:

$$\Phi_f = (h\mathbf{u})_f \cdot \mathbf{S}_f - \tau_f \left([\nabla \cdot (h\mathbf{u} \otimes \mathbf{u})]_f + gh_f [\nabla (b+h)]_f \right) \cdot \mathbf{S}_f, \tag{20}$$

$$\mathbf{F}_f = \Phi_f \mathbf{u}_f + \frac{g(h_f)^2}{2} \mathbf{S}_f - \mathbf{S}_f \cdot \mathbf{\Pi}_f + g\left(\tilde{h}_f - \tau_f \Big[\nabla \cdot (h\mathbf{u}) \,]_f\right) b_f \mathbf{S}_f, \tag{21}$$

$$\begin{aligned} \mathbf{\Pi}_f = \mathbf{\Pi}_{NSf} + \tau_f \mathbf{u}_f \otimes \left((h\mathbf{u})_f \cdot [\nabla \mathbf{u}]_f + gh_f [\nabla (b+h)]_f \right) + \\ + \tau_f \mathbf{I} \left(gh_f [\nabla \cdot (h\mathbf{u})]_f \right). \end{aligned} \tag{22}$$

$$\mathbf{\Pi}_{NSf} = \tau_f g \frac{h_f^2}{2} \left[(\nabla \otimes \mathbf{u}) + (\nabla \otimes \mathbf{u})^{\mathrm{T}} \right]_f. \tag{23}$$

Величины, значения которых взяты на грани f ячейки, обозначены нижним индексом $_f$, а скобки $[\cdot]_f$ говорят о том, что оператор, заключенный в скобках, берется в центре ребра f.

Определение $\tilde{h}_f$ будет дано в следующем разделе.

Величину τ, используя формулы (15) или (16), в центре ячейки можно определить по-разному. Рассмотрим ячейку с центром в точке P, рис 4. Выпишем расстояние между ячейками с поправкой на ортогональность сетки

$$\delta_f = \delta_{PN} = \frac{|\mathbf{d}|^2}{(\mathbf{d} \cdot \mathbf{n}_f)}.$$

Тогда величину τ_P можно определить, используя средневзвешенное расстояние между центром ячейки P и всех соседних ячеек C_δ

$$\tau_P = \frac{\alpha C_\delta}{\sqrt{gh_P + |\mathbf{u}_P|}}, \quad C_\delta = \frac{\sum_f \delta_f |S_f|}{\sum_f |S_f|}. \tag{24}$$

Заметим, что для аппроксимации потоков требуется вычисление частных производных на гранях f вычислительной ячейки. В то время как нормальный к поверхности грани компонент дифференциальных операторов может быть аппроксимирован с помощью линейной интерполяции значений в центрах ячеек со смежными гранями, компоненты тангенциального направления требуют дополнительного рассмотрения. Для этого воспользуемся подходами, предложенными для аппроксимации τ-слагаемых в случае регуляризованных уравнений для моделирования несжимаемой жидкости:

а) расчет значений в центрах ячеек с линейной интерполяцией их на грани;
б) редуцированный метод, при котором используется только нормальная к поверхности компонента производной, а тангенциальные компоненты не учитываются [13];
в) метод наименьших квадратов, который использует разложение Тейлора для получения дискретных выражений для частных производных [13];
г) применение метода Гаусса к фиктивному контрольному объему, определенному вокруг рассматриваемой грани f [14]. В рамках этого метода расчетный шаблон включает вершины, грани и точки в ячейках, прилегающих к грани (см. рис. 5). Например, выражение для производной по x скалярного поля C на четырехугольной грани выглядит следующим образом:

$$\frac{\partial C}{\partial x} \approx \frac{1}{V_f} \sum_{m=1}^{8} n_{m,x} C_m,$$

где V_f – объем фиктивной восьмигранной ячейки, построенной вокруг грани f, m – индекс грани фиктивной ячейки, C_m – среднее значение скалярного поля C на грани m, $n_{m,x}$ – x-компонента нормали к грани m.

Как показали численные эксперименты, последний подход представляется наиболее эффективным с учетом предыдущей практики и успешного применения для моделирования несжимаемой жидкости [15], поэтому именно данный подход используется для аппроксимации производных на гранях расчетных ячеек.

К сожалению, в OpenFOAM отсутствует готовая библиотека для аппроксимации производных на гранях ячеек, однако она была реализована в рамках QGDSolver – фрэймворка на основе OpenFOAM, предназначенного для моделирования течений жидкости с использованием регуляризованных уравнений (квазигазодинамического и квазигидродинамического подходов). Фрэймворк

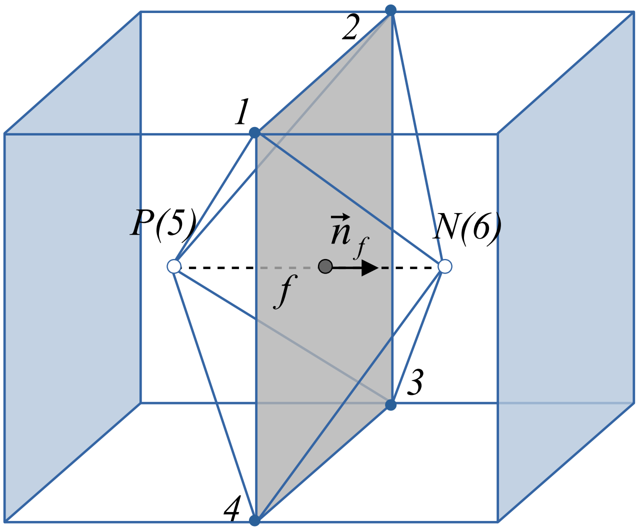


Рис. 5. Схема шаблона для вычисления частных производных на грани f конечного объема: P – центр ячейки с нормалью от f, направленной наружу, N обозначает центр ячейки, к которой нормаль от f направлена внутрь

содержит библиотеку для аппроксимации частных производных в центрах граней ячеек и набор решателей для широкого спектра задач. Он является открытым и доступен для скачивания [2]. В работах [16, 17] описаны особенности библиотеки, а также основные решатели фреймворка QGDSolver – QHDFoam и QGDFoam.

Условия хорошей балансировки и сухого дна

Условия “хорошей балансировки” (с англ. – well balanced) и сухого дна (с англ. – dry zone conditions) чрезвычайно важны при решении задач в приближении мелкой воды, поскольку они обеспечивают устойчивость численного алгоритма и позволяют корректно решать практические задачи, в особенности в прибрежных зонах. Описываемый здесь метод удовлетворяет этим условиям, см., например [7], однако важно убедиться, что это справедливо и для реализации алгоритма в рамках OpenFOAM решателя RSWEFoam.

Условия “well-balanced” описаны в различных статьях, например в [18], и требуют от вычислительного алгоритма, чтобы решение задачи с начальными условиями покоящейся жидкости не зависело от времени, т.е. что в изначально покоящейся жидкости не должны возникать возмущения, обусловленные неровностями дна. Для того чтобы выполнить это, запишем условия для покоящейся жидкости:

$$\begin{cases} h(x,y,t) = h(x,y,t=0) \implies \xi(x,y,t) = h(x,y) + b(x,y) = const; \\ \mathbf{u}(x,y,t) = \mathbf{u}(x,y,t=0) = \mathbf{0}. \end{cases} \tag{25}$$

С учетом (25) для того, чтобы алгоритм удовлетворял условиям хорошей балансировки, необходимо, чтобы $\sum_f \Phi_f = 0$ и $\sum_f \mathbf{F}_f = \mathbf{0}$. Для проверки этого выпишем величины потоков (20), (21), подставив необходимые выражения и убрав слагаемые, тождественно равные 0:

$$\sum_f \Phi_f = \sum_f \left(g\tau_f h_f \left[\nabla \left(b+h\right)\right]_f \right) \cdot \mathbf{S}_f = \sum_f \left(g\tau_f h_f \left[\nabla \left(\xi\right)\right]_f \right) \cdot \mathbf{S}_f \equiv 0. \quad (26)$$

$$\sum_f \mathbf{F}_f = \sum_f \left(\frac{g\left(h_f\right)^2}{2}\mathbf{S}_f + g\tilde{h}_f b_f \mathbf{S}_f \right). \quad (27)$$

Исходя из построения потока Φ_f и условия покоящейся жидкости (25) условие well-balanced выполняется автоматически, однако для потока импульса это не так. Чтобы это исправить, определим $\tilde{h}_f$ как

$$\tilde{h}_f = \frac{\sum_f \left(h_f\right)^2}{2\sum_f h_f}. \quad (28)$$

Нетрудно заметить, что, подставив выражение (28) в (27), получим $\sum_f \mathbf{F}_f \equiv \mathbf{0}$. Отметим, что для упрощения вычислений можно использовать $\tilde{h}_f = h_f$, например, для задач с плоским дном или задач, в которых эффект неровности дна не так значителен.

Для того чтобы корректно рассчитывать течения вблизи берега, необходимо учитывать условия сухого дна [18, 19]. В рамках этих условий вводится параметр отсечения ε, который определяет минимально возможное значение толщины слоя воды. Подробное описание работы алгоритма в рамках этих условий приведено в [20]. Краткая формулировка условий сухого дна применительно к РУМВ выглядит следующим образом:

$$\text{если } h < \varepsilon: \quad \mathbf{u} = \mathbf{0}, \quad \tau = 0, \quad (29)$$

кроме того, для всех граней, прилегающих к сухой ячейке, проверяется отсутствие исходящих потоков, чтобы строго соблюдался закон сохранения массы.

Отметим, что для ε существует ограничение снизу, накладываемое исходя из геометрических соображений, см. [10], выражение для уровня отсечения в ячейке P:

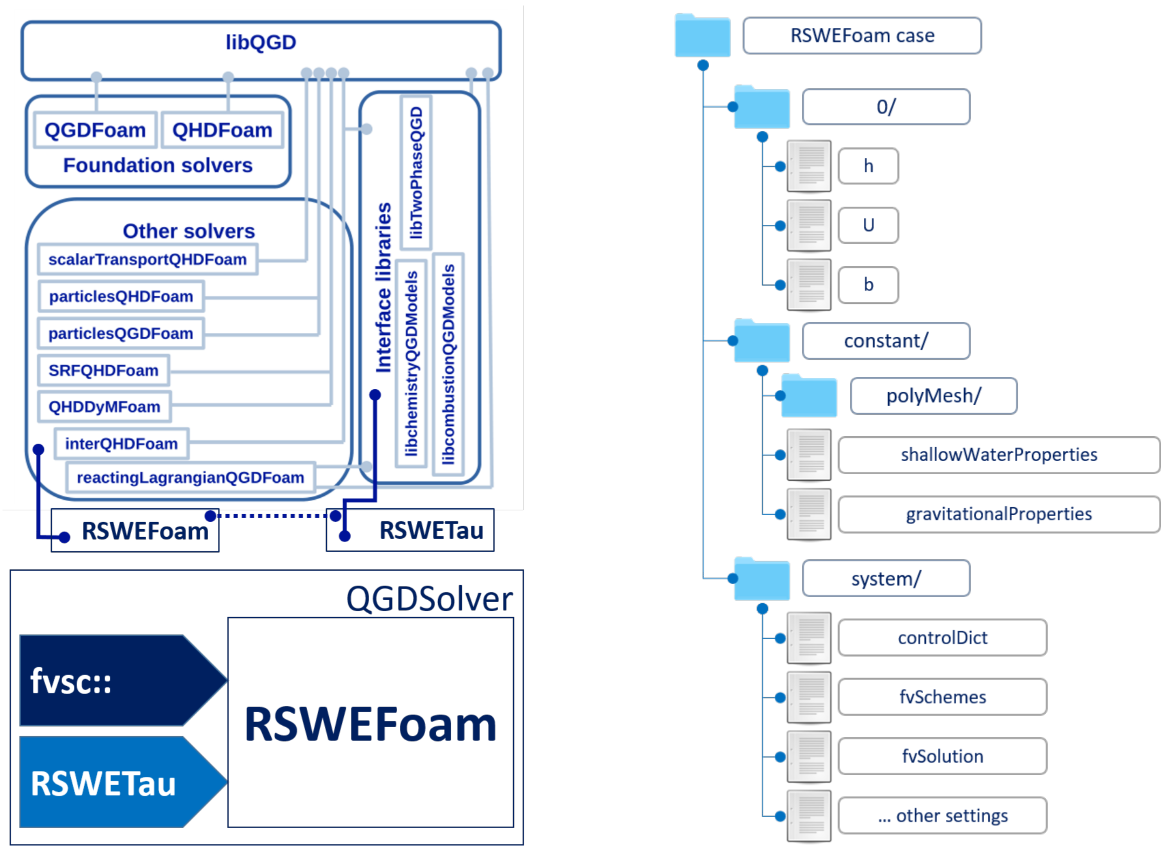
$$\varepsilon_P \leqslant \max_{N\in\{N_i\}} \left(|b_P - b_N|, \varepsilon_0\right),$$

где $\{N_i\}$ – центры ячеек, смежных с ячейкой P, ε_0 – минимальное значение параметра.

Особенности реализации решателя RSWEFoam и основы работы с ним

На данный момент решатель доступен для скачивания в репозитории лаборатории ЛСПО ЦМТС ИСП РАН [2]. В ветке https://github.com/unicfdlab/QGDsolver/tree/digitef-dev-v2012 в папке app, в которой располагаются все решатели библиотеки, находится одноименная папка RSWEFoam с исходным кодом решателя. А в папке `tutorials/RSWEFoam` располагаются описанные в данной работе вычислительные кейсы: `dryDamBreak` – задача распада разрыва над сухим дном, `asymDam` – разрушение несимметричной дамбы, `3Cones` – задача о затоплении поверхности с тремя конусами.

Благодаря структуре фреймворка QGDSolver [2] решатель RSWEFoam реализуется и интегрируется в него естественным образом, рис. 6а. Основные библиотеки, которые необходимо использовать для реализации решателя – библиотека для аппроксимации пространственных производных на гранях ячеек fvsc, которая реализована в рамках фреймворка, и библиотека для расчета параметра τ.



а б

Рис. 6. (а) Схема реализации решателя RSWEFoam; (б) структура вычислительного кейса.

Для запуска расчета решателя RSWEFoam необходимо подготовить вычислительный кейс рис. 6б, т.е. построить сетку и определить начальные и граничные условия, как и для любого другого решателя OpenFOAM, а кроме того задать параметры задачи. Параметры, относящиеся к методу РУМВ, располагаются в файле shallowWaterProperties:

```
1  QGD
2  {
3      QGDCoeffs   RSWETau;
4  }
5
6  alpha        0.2;
7  wellBalancedScheme  true;
8  dryZoneCondition  true;
9  eps0    1e-6;
10 tauU    1.0;
11 NS    1.0;
```

Листинг 1 Файл shallowWaterProperties

Здесь строки 1 – 4 – обязательные для решателя, в них указывается, что для расчета параметра τ используется конкретная библиотека RSWETau.

Все остальные параметры являются необязательными; в случае если они не указаны, будут заданы их стандартные значения (укажем их в скобках):

- `alpha` – коэффициент α (стандартное значение 0.5);
- `wellBalancedScheme` – булевый параметр, определяющий, использовать ли схему well-balanced для $\tilde{h}_f$ (28) или использовать $\tilde{h}_f = h_f$ (стандартное значение false, т.е. $\tilde{h}_f = h_f$);
- `dryZoneCondition` – булевый параметр, определяющий, использовать ли условия сухого дна (стандартное значение false);
- `eps0` – параметр отсечения ε_0 (стандартное значение 10^{-6}м);
- `tauU` – вещественный коэффициент, указывающий, в каком соотношении использовать величину скорости $|\mathbf{u}|$ в выражении для параметра τ, т.е. $\tau_P = \alpha C_\delta / \sqrt{g h_P + \texttt{tauU} |\mathbf{u}_P|}$, при $\texttt{tauU} = 0$ это формула (15), а при $\texttt{tauU} = 1$ – формула (16) (стандартное значение 0);
- `NS` – вещественный коэффициент, указывающий, в каком соотношении использовать тензор вязких напряжений $\mathbf{\Pi}_{\mathrm{NS}}$ в выражении для $\mathbf{\Pi}$, т.е. при $\texttt{NS} = 0$ $\mathbf{\Pi}_{\mathrm{NS}}$ не используется, а при $\texttt{NS} = 1$ – формула (6) (стандартное значение 1);

Для запуска решателя используются стандартные OpenFOAM команды:

```
> blockMesh
> RSWEFoam
```

Листинг 2 Запуск решателя

3. Расчетные задачи

Отметим для краткости, что во всех задачах, описанных ниже, используется схема для выполнения условия well-balanced, а также в $\mathbf{\Pi}$ входит тензор вязких напряжений $\mathbf{\Pi}_{NS}$.

Одномерная задача распада разрыва над сухим дном

Одномерная задача распада разрыва – задача Римана – является классическим тестом для проверки численных методов решения уравнений мелкой воды. Постановка задачи состоит в том, что в начальный момент времени одномерная область с плоским дном $b = const$ разделена пополам разрывом. По обе стороны от разрыва разные уровни жидкости h_L – слева, h_R – справа. Для задачи можно построить аналитическое решение, которое зависит от соотношения уровней h_R, h_L.

Для апробации решателя RSWEFoam интересен частный случай задачи, при котором $h_R = 0$, т.е. когда справа от разрыва располагается сухая зона. Это позволит проверить сразу несколько параметров алгоритма.

Рассмотрим область длиной $L = 50$м с плоским дном $b = const$, где в начальный момент времени в центре области располагается разрыв уровня воды, жидкость при этом покоится $u(x, t = 0) = 0$. Задача была разобрана в [21]. Величина ускорения свободного падения $g = 10$ м/с^2. Уровень воды слева от разрыва $h_L = 1$м, справа вода отсутствует $h_R = 0$м, рис. 7а. Аналитическое решение задачи в общем виде, [21]:

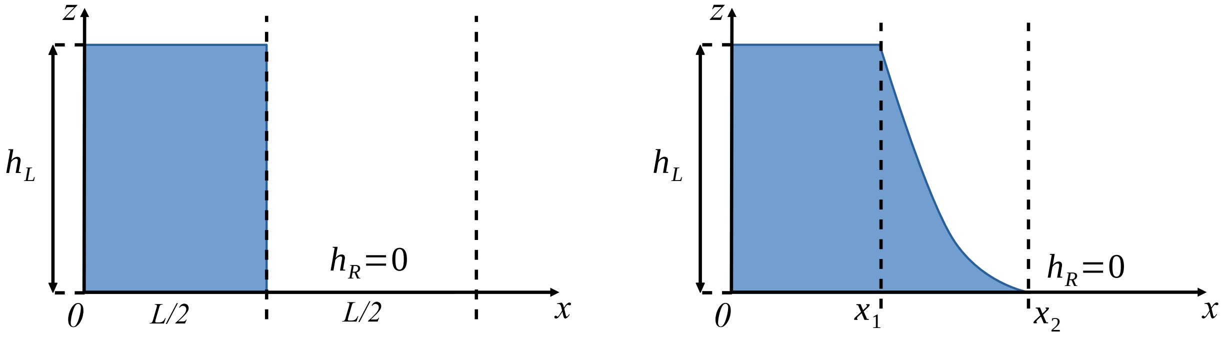
$$h(x,t), u(x,t) = \begin{cases} h = h_L, \quad u = 0, \quad \text{если } x \leqslant x_1 = -t\sqrt{gh_L}; \\ h = \frac{1}{9g}\left(2\sqrt{gh_L} - \frac{x}{t}\right)^2 h_L, \quad u = \frac{2}{3}\left(\sqrt{gh_L} + \frac{x}{t}\right), \\ \text{если } x_1 < x \leqslant x_2 = -t\sqrt{gh_L}; \\ h = 0, \quad u = 0, \quad \text{если } x_2 < x. \end{cases} \tag{30}$$

Расчет ведется до момента времени $t = 3$с.

При распаде образуются две точки движения разрыва x_1 и x_2, при этом отметим, что положение точки x_2 в расчете определяется условиями сухого дна.

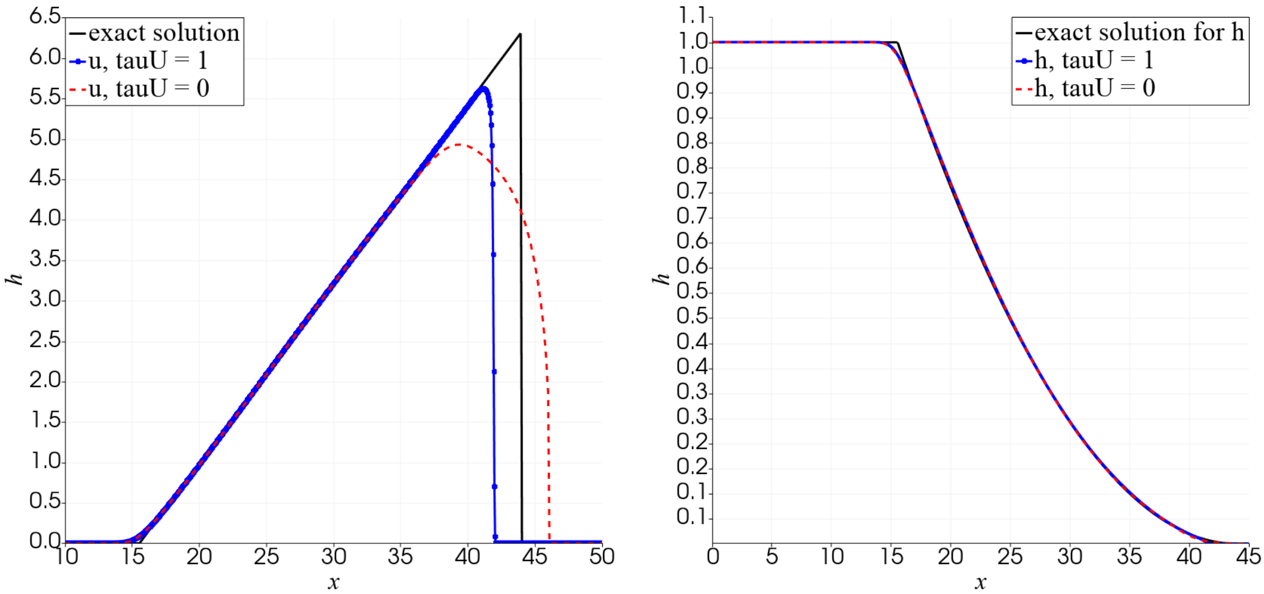
На рис. 8 приведено сравнение двух подходов для определения параметра τ с аналитическим решением на сетке $\Delta x = 0.05$м, с параметрами $\alpha = 0.2$, $\beta = 0.1$, $\varepsilon = 10^{-4}$. Можно заметить, что решение для скорости при использовании формулы (16) ближе к аналитическому, а для (15) решение сильно сглаживается, что объясняется быстрым ростом τ вблизи сухого дна, который компенсируется скоростью распространяющегося разрыва в первом случае. Решение для толщины слоя воды, рис. 8б, практически не отличается.

Для исследования сходимости решения по сетке рассматривается более близкий к аналитическому подход определения τ по формуле (16), рис. 9. Численное решение совпадает с аналитическим и отличается только на границе с сухим дном. Из рис. 9 можно заметить, что уменьшение шага сетки сдвигает эту границу. Кроме того, значительное влияние на расположение сухой точки для численного решения оказывает величина параметра отсечения ε. Отметим,



а б

Рис. 7. Схема решения. (а) Распределение жидкости в начальный момент времени; (б) распределение жидкости в момент времени $t > 0$, точки x_1, x_2 движутся в противоположных направлениях от начального положения разрыва.



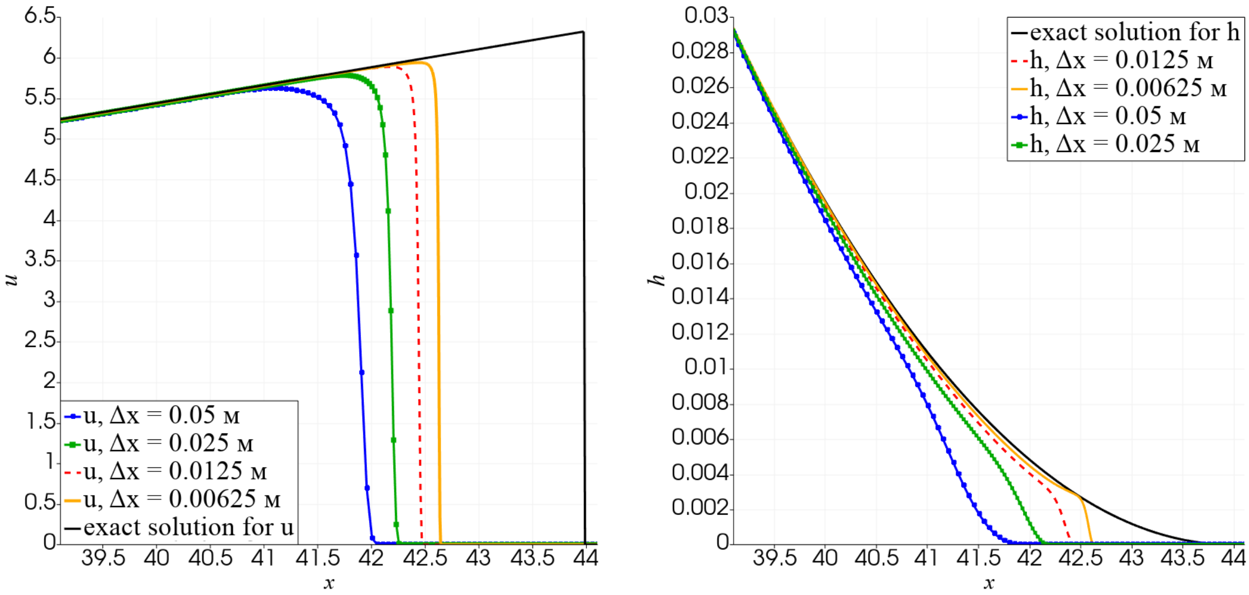
а б

Рис. 8. Сравнение подходов для вычисления τ для момента времени $t = 3$с, $\varepsilon = 10^{-4}$м, $\alpha = 0.2$, $\beta = 0.1$: (а) скорость течения u; (б) толщина слоя воды h.

что при значении $\varepsilon < 10^{-6}$м для обеспечения устойчивости численного алгоритма необходимо уменьшать шаг по времени.

Разрушение несимметричной дамбы

Возможности работы алгоритма для двумерных задач иллюстрируются примером расчета нестационарного течения, возникающего при разрушении несимметричной дамбы. Эта задача является известным тестом, см., например, [19]. С помощью индивидуального кода РУМВ эта задача моделировалась в [7].



а б

Рис. 9. Сходимость численного решения по сетке в момент времени $t = 3$с. Фрагменты численного решения, параметр τ рассчитан по формуле (16), $\varepsilon = 10^{-6}$м, $\alpha = 0.2$, $\beta = 0.1$: (а) скорость течения u; (б) толщина слоя воды h.

В соответствии с [19] рассматривается задача о течении, возникающем при мгновенном разрушении дамбы, разделяющей два бассейна с водой в квадратной области с плоским дном $b(x, y) = 0$. Высота уровня воды в левом бассейне составляет 10м, в правом 5м. Длина разрыва равна 75м, начало разрыва расположено в точке с координатой $y = 95$м, рис. 10. Толщина стенки дамбы равна 10м, и ее левая сторона расположена в точке с координатой $x = 95$м. Величина ускорения свободного падения $g = 9.81$ м/с^2. На всех границах дамбы ставятся граничные условия отражения

$$\frac{\partial h}{\partial \mathbf{n}} = 0, \quad u_n = 0, \quad \frac{\partial u_\tau}{\partial \mathbf{n}} = 0.$$

Линии уровня для распределения толщины слоя воды на момент времени $t = 7.2$с представлены на рис. 10. Расчеты проведены на равномерной прямоугольной сетке с шагами $\Delta x = \Delta y = 1$м и с заданными значениями параметров $\alpha = 0.2$, $\beta = 0.2$. На рис. 10 можно заметить, что в решении проявляются характерные особенности возникающего к указанному моменту времени течения, а именно: сглаженный немонотонный профиль слева от разрыва и резкий, но монотонный профиль в правой части области, а также отражение волны от верхней стенки бассейна. В статье [19] задача о разрушении несимметричной дамбы решается с использованием неструктурированной пространственной треугольной сетки с помощью двух численных алгоритмов высокого порядка точности, результаты которых оказываются близкими. Результаты расчетов на основе алгоритма с регуляризацией для $\Delta x = \Delta y = 1$м хорошо совпадают с решением [19], полученным на сетке с шагами порядка 2м.

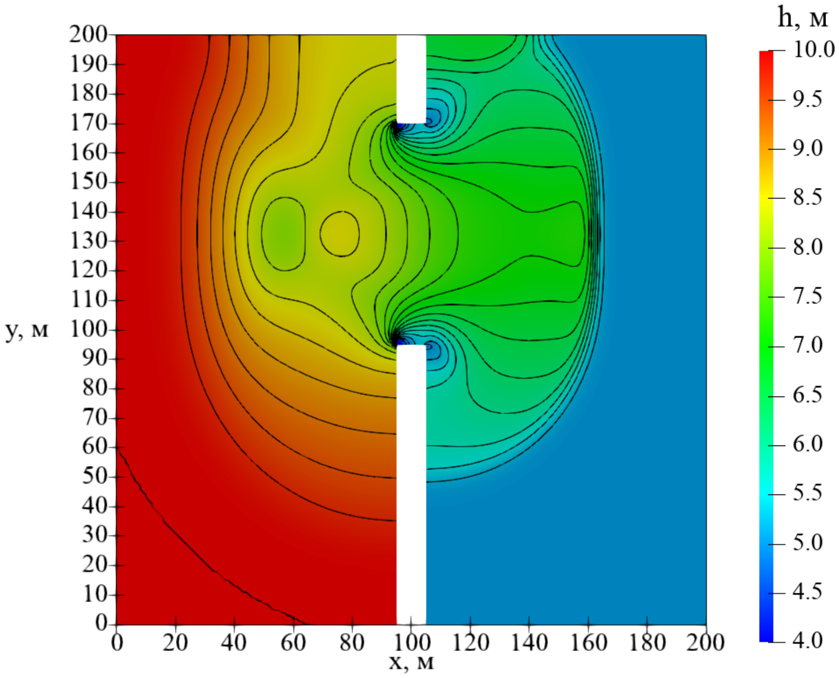


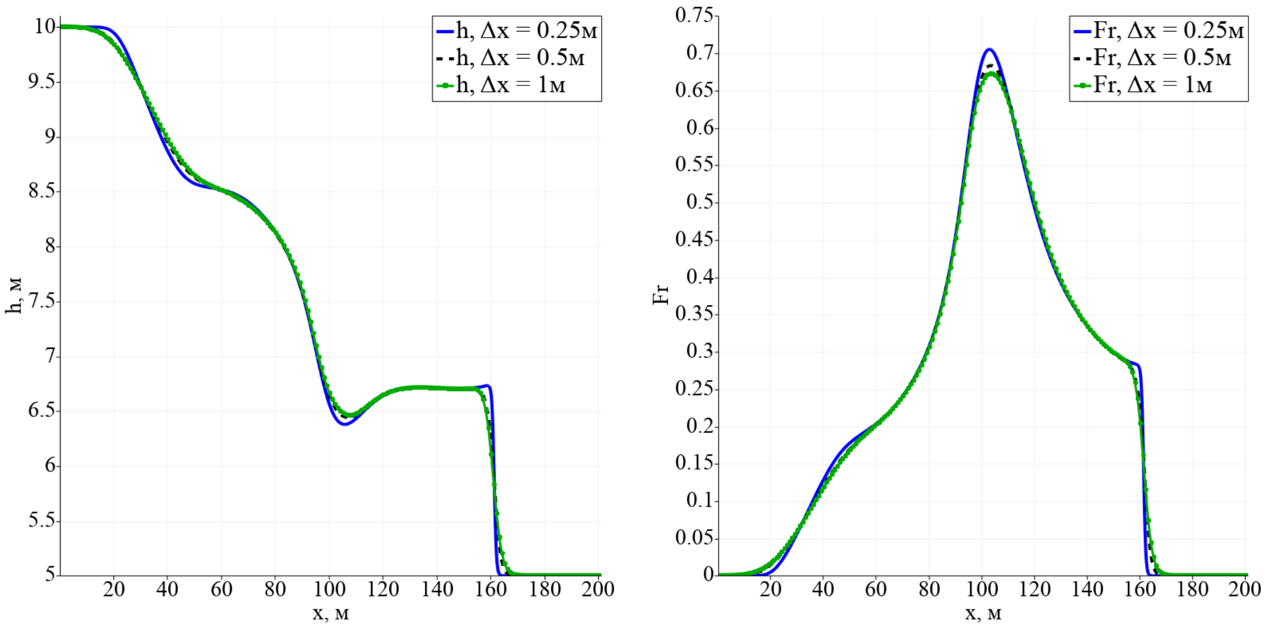
Рис. 10. Разрушение несимметричной плотины. Толщина слоя жидкости и линии уровня, $t = 7.2$с, $\Delta x = \Delta y = 1$м, $\alpha = 0.2$, $\beta = 0.2$.

В модели мелкой воды аналогом числа Маха в газовой динамике Ma $= |u|/c_S$, где $c_S = \sqrt{\gamma RT}$ скорость звука в газе, является число Фруда $Fr = |u|/c$. При этом скорость распространения малых возмущений вычисляется как $c = \sqrt{gh}$.

На рис. 11 представлены распределения уровня воды и числа Фруда вдоль линии $y = 160$ м, параллельной оси OX в момент времени $t = 7.2$с и с параметрами $\alpha = 0.2$, $\beta = 0.2$ для различных пространственных шагов $\Delta x = \Delta y = 1$м, $\Delta x = \Delta y = 0.5$м и $\Delta x = \Delta y = 0.25$м. Видна сходимость решения при сгущении сетки как для h, так и для Fr. Профили численного решения для уровня воды и числа Фруда соответствуют результатам [19] и приближаются к нему при уменьшении шага сетки.

Задача о затоплении поверхности с тремя конусами

Задача о разрушении плотины и затоплении поверхности с тремя конусами, освещенная в различных работах, например [18, 22], приведена здесь для демонстрации применения условий сухого дна, записанных в рамках OpenFOAM, к пространственным задачам. Впервые в рамках РУМВ эта задача была изучена с помощью индивидуального кода на структурированных треугольных сетках в [21].



а б

Рис. 11. Разрушение несимметричной плотины, срез вдоль линии $y = 160$м. Сходимость численного решения по сетке в момент времени $t = 7.2$с, $\alpha = 0.2$, $\beta = 0.2$: (а) уровень воды $h(x) = h(x, y = 160)$; (б) Число Фруда $\mathrm{Fr}(x) = \mathrm{Fr}(x, y = 160)$.

Рассматривается канал прямоугольной формы длиной 75м и шириной 30м. Начальный разрыв расположен на линии $x = 16$м, уровень воды слева от разрыва составляет $h = 1.875$м, а остальная область полагается сухой ($h = 0$). Канал со всех сторон огражден непроницаемыми стенками, на которых ставятся условия отражения

$$\frac{\partial h}{\partial \mathbf{n}} = 0, \quad u_n = 0, \quad \frac{\partial u_\tau}{\partial \mathbf{n}} = 0.$$

Обтекаемые препятствия представляют собой три конуса разных размеров. Основание первого конуса высотой $h_1 = 3$м расположено в центре с координатами $x_1 = 47.5$м, $y_1 = 15$м, радиус основания $r_1 = 10$м. Два других конуса имеют одинаковую высоту $h_2 = 1$м и радиус $r_2 = 8$м. Центр одного из малых конусов расположен в точке $x_2 = 30$м, $y_2 = 6$м, а центр другого малого конуса в точке $x_3 = 30$м, $y_3 = 24$м. Таким образом, батиметрию в задаче можно задать, используя функцию

$$b(x, y) = \max\left[0, 3 - \frac{3}{10}\sqrt{(x - x_1)^2 + (y - y_1)^2}, 1 - \frac{1}{8}\sqrt{(x - x_2)^2 + (y - y_2)^2},\right.$$
$$\left. 1 - \frac{1}{8}\sqrt{(x - x_3)^2 + (y - y_3)^2}\right].$$

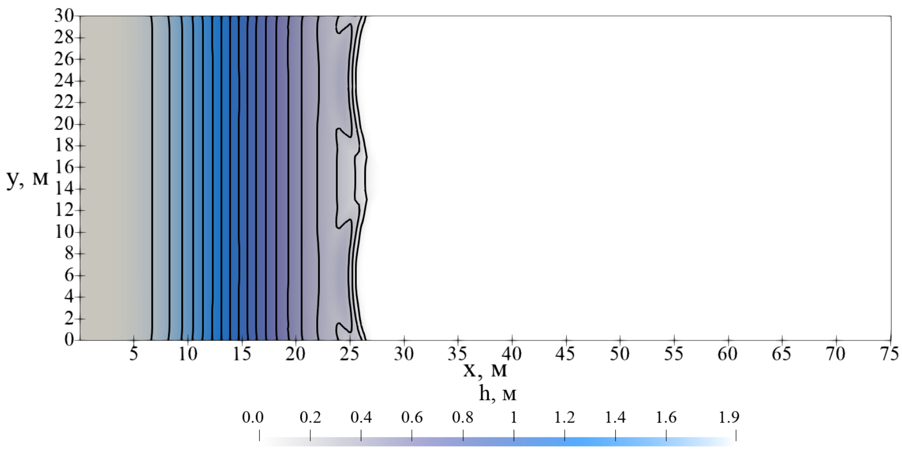
Для приближения результатов численного моделирования к реальным вводится сила трения о дно, которая имеет вид

$$\mathbf{f} = -\frac{gn^2}{h^{4/3}}\mathbf{u}|\mathbf{u}|, \tag{31}$$

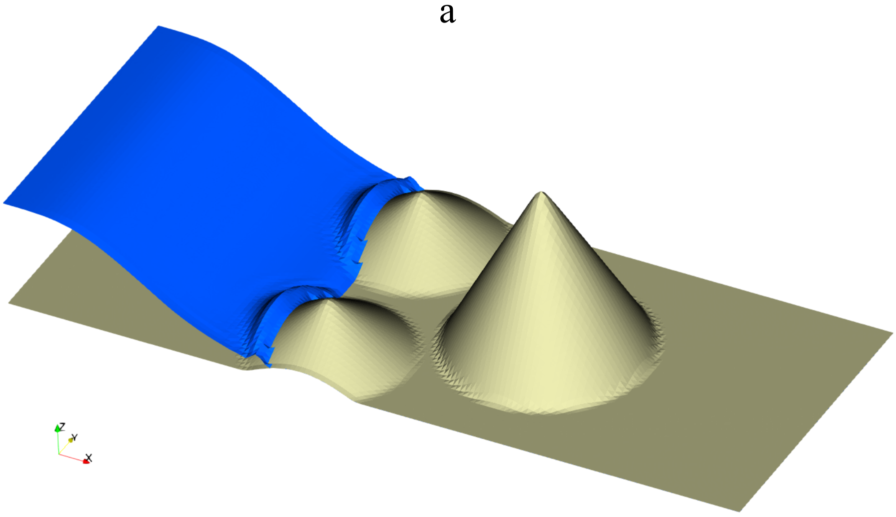
где n – коэффициент Маннинга, который зависит от материала подстилающей поверхности. В задаче полагается $n = 0.018$ с/м$^{1/3}$.

Численное решение проводилось на равномерной прямоугольной сетке с шагами по пространству $\Delta x = \Delta y = 0.5$м. Параметр регуляризации $\alpha = 0.2$, число Куранта $\beta = 0.1$, а минимальное значение параметра отсечения $\varepsilon_0 = 10^{-3}$м. Расчет проводился до момента затухания колебаний воды в канале $t = 300$с.

На рис. 12–15 представлены результаты расчета распределения уровня воды в моменты времени $t = 2$с, $t = 6$с, $t = 12$с и $t = 30$с соответственно.



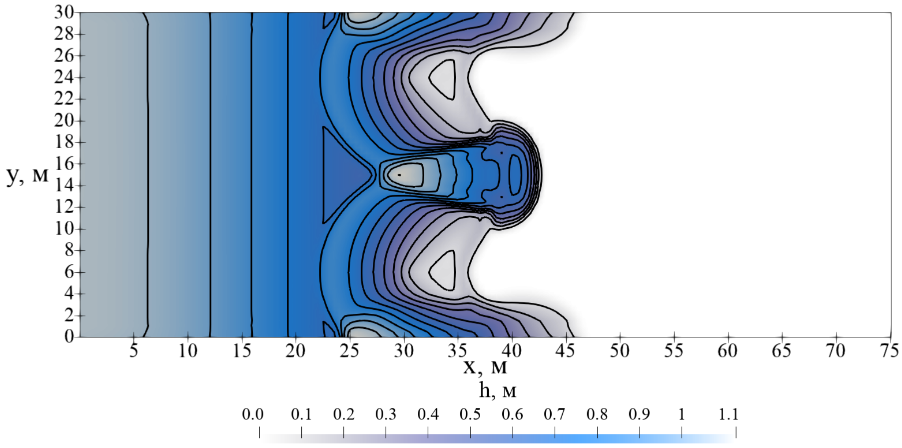
а



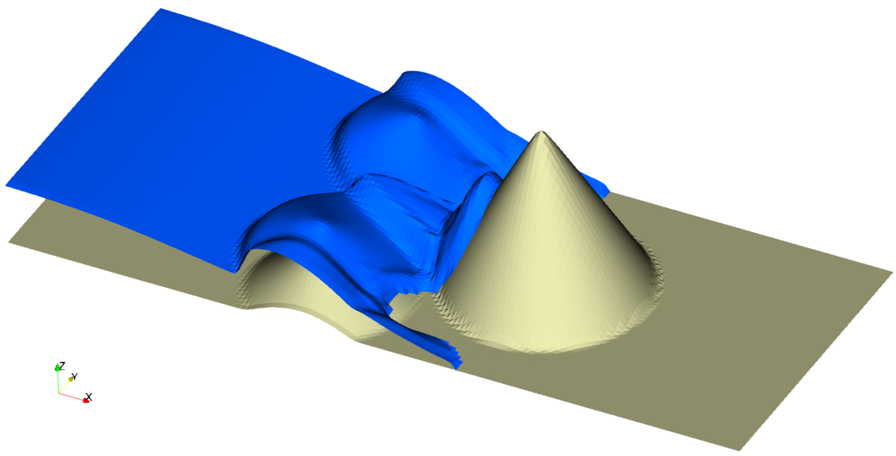
б

Рис. 12. Уровень воды для момента $t = 2$с: (а) линии уровня; (б) 3D вид.

Каждый рисунок состоит из распределения уровня воды и изолиний (а), а также из трехмерного вида задачи (б). Отметим, что любой сколь угодно малый слой воды прорисовывается на трехмерном виде задачи, поэтому на нем не всегда корректно отображается область сухого дна.



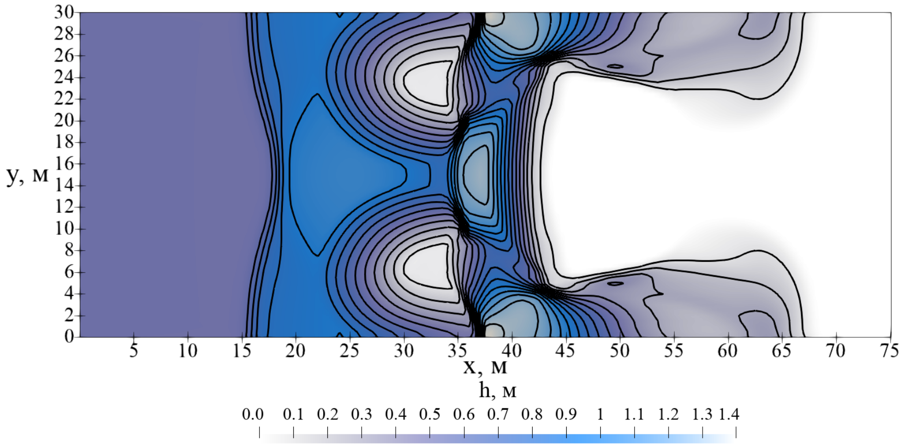
а



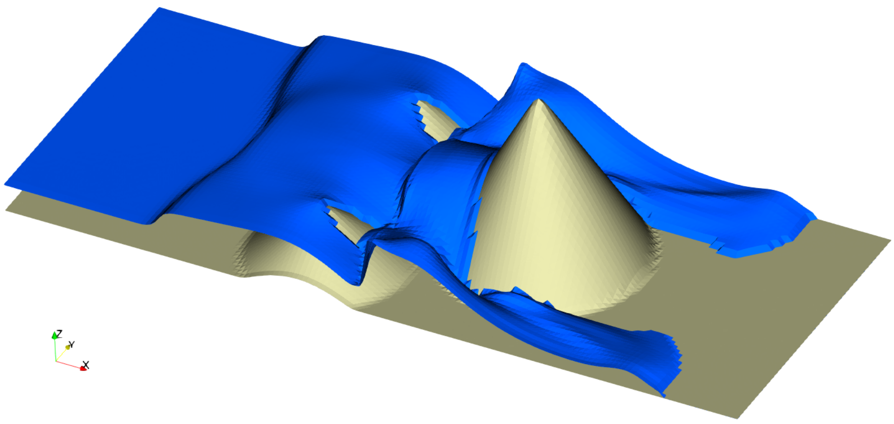
б

Рис. 13. Уровень воды для момента $t = 6$с: (а) линии уровня; (б) 3D вид.

На рисунках можно заметить, как жидкость обтекает два маленьких холмика, рис. 12, т.е. как происходит процесс набегания волны на наклонную поверхность. На рис. 13 также виден процесс столкновения набегающей волны с самым высоким конусом, вершина которого всегда остается сухой. Затем вокруг маленьких холмиков образуется область с сухим дном, рис. 14а, при этом в расчете в этой области наблюдается тонкий слой воды, который отображается в трехмерном виде задачи, рис. 14б. Распространяющиеся волны по обе стороны от большого конуса приобретают округлую форму благодаря наличию силы трения (31). На финальном рис. 15 можно заметить, как отраженная волна натекает на большой конус. Приведенные результаты (рис. 12–15) хорошо соответствуют численным результатам, полученным с применением адаптивных сеток [22] и неструктурированных сеток [18], где для расчета использовался метод Годунова, модифицированный специальным образом для решения задач с сухим дном.



а



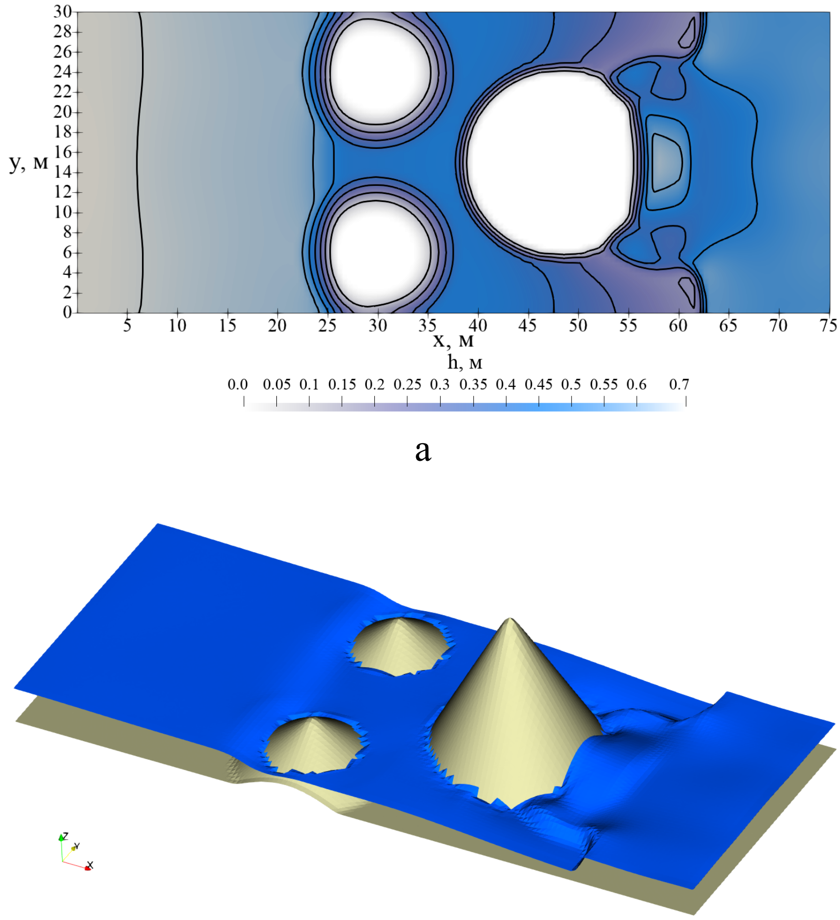
б

Рис. 14. Уровень воды для момента $t = 12$с: (а) линии уровня; (б) 3D вид.

Заключение

На текущий момент разработан и протестирован решатель RSWEFoam, который был описан в данной работе. Основным результатом является включение решателя в библиотеку QGDSolver, благодаря чему он стал доступен всем желающим.

Основная задача RSWEFoam – решение прикладных задач, поэтому в дальнейшем планируется расширить функционал решателя и дополнить его работой с внешними силами (силой Кориолиса, донного трения и силы трения ветра), граничными условиями, описывающими реальные водоемы (это необходимо, например, для моделирования приливов или сложных течений на границе), а также рассмотреть водоемы с батиметрией более сложной формы, для которых появится необходимость в использовании неструктурированных сеток. Кроме того, аналогично вышеописанному подходу может быть реализован



а

б

Рис. 15. Уровень воды для момента $t = 30$с: (а) линии уровня; (б) 3D вид.

и решатель для двухслойной мелкой воды, а также подключен модуль, описывающий перенос примеси.

Библиографический список

[1] Weller H. G., Tabor G., Jasak H., Fureby C. A tensorial approach to computational continuum mechanics using object-oriented techniques. 1998. Access mode: https://www.openfoam.com.

[2] Крапошин М. В., Рязанов Д. А., Ватутин К. А., Епихин А.С. Фреймворк QGDsolver.
Режим доступа: https://github.com/unicfdlab/QGDsolver.

[3] Ландау Л. Д., Лифшиц Е. М. Гидродинамика. М.: Наука, 1986. 736 с.

[4] Елизарова Т. Г., Иванов А. В. Численное моделирование переноса пассивного скаляра в мелкой воде с использованием квазигазодинамического подхода // ЖВМиМФ. 2020. Т. 60, № 7. С. 1248–1267. Режим доступа: http://doi.org/10.31857/s0044466920070066.

[5] Елизарова Т. Г., Иванов А. В. Об однородном алгоритме численного

моделирования волны цунами // Ученые записки физического факультета Московского Университета. 2018. № 3. С. 1830103–1–1830103–6. Режим доступа: http://uzmu.phys.msu.ru/abstract/2018/3/1830103.

[6] Елизарова Т. Г., Сабурин Д. С. Численное моделирование колебаний жидкости в топливных баках // Математическое моделирование. 2013. Т. 25, № 3. С. 75–88.

[7] Елизарова Т. Г., Булатов О. В. Регуляризованные уравнения мелкой воды и эффективный метод численного моделирования течений в неглубоких водоемах // ЖВМиМФ. 2011. Т. 51, № 1. С. 170–184. Режим доступа: http://mi.mathnet.ru/zvmmf8054.

[8] Шеретов Ю. В. Регуляризованные уравнения гидродинамики. Тверь: Тверской государственный университет, 2016. 222 с. Режим доступа: https://www.researchgate.net/publication/311562501_Regularized_Hydrodynamic_Equations.

[9] Елизарова Т. Г., Сабурин Д. С. Применение регуляризованных уравнений мелкой воды к моделированию сейшевых колебаний уровня Азовского моря // Математическое моделирование. 2017. Т. 29, № 1. С. 45–62.

[10] Елизарова Т. Г., Булатов О. В. Регуляризованные уравнения мелкой воды для численного моделирования течений с подвижной береговой линией // Журнал вычислительной математики и математической физики. 2016. Т. 56, № 4. С. 158–177.

[11] Злотник А. А. Энергетические равенства и оценки для баротропных квазигазо- и квазигидродинамических систем уравнений // ЖВМиМФ. 2010. Т. 50, № 2. С. 325–337. Режим доступа: http://mi.mathnet.ru/zvmmf4831.

[12] Злотник А. А. О построении квазигазодинамических систем уравнений и баротропной системы с потенциальной массовой силой // Матем. моделирование. 2012. Т. 24, № 4. С. 65–79. Режим доступа: http://mi.mathnet.ru/mm3259.

[13] Development of OpenFOAM Solver for Compressible Viscous Flows Simulation Using Quasi-Gas Dynamic Equations / Kraposhin M. V., Daniil R. A., Smirnova E. V. et al. / 2017 Ivannikov ISPRAS Open Conference (ISPRAS). IEEE, 2017. Access mode: http://doi.org/10.1109/ISPRAS.2017.00026.

[14] Истомина М. А., Шильников Е. В. Об аппроксимации потоковых величин на пространственных сетках нерегулярной структуры // Препринты ИПМ им. М.В.Келдыша. 2019. № 86. С. 1–22. Режим доступа: https://library.keldysh.ru/preprint.asp?id=2019-86.

[15] Shilnikov E. V., Elizarova T. G. Simulation of hypersonic flows using the QGD-based parallel program complex “EXPRESS-3D” // High Temp. Mater. Processes: Int. Q. High-Technol. Plasma Processes. 2018. Vol. 22, no. 2–3. P. 99–113. Access mode: http://doi.org/10.1615/HighTempMatProc.

2018024713.

[16] Development of a new OpenFOAM solver using regularized gas dynamic equations / M. V. Kraposhin, E. V. Smirnova, T. G. Elizarova, M. A. Istomina / Computers & Fluids. 2018. Vol. 166. P. 163–175. Access mode: http://doi.org/10.1016/j.compfluid.2018.02.010.

[17] Kraposhin M. V., Ryazanov D. A., Elizarova T. G. Numerical algorithm based on regularized equations for incompressible flow modeling and its implementation in OpenFOAM // Computer Physics Communications. 2022. Vol. 271. P. 108216. Access mode: http://doi.org/10.1016/j.cpc.2021.108216.

[18] Huang Yuxin, Zhang Ningchuan, Pei Yuguo. Well-Balanced Finite Volume Scheme for Shallow Water Flooding and Drying Over Arbitrary Topography // Engineering Applications of Computational Fluid Mechanics. 2013. Vol. 7, no. 1. P. 40–54. Access mode: http://doi.org/10.1080/19942060.2013.11015452.

[19] Ricchiuto M., Abgrall R., Deconinck H. Application of conservative residual distribution schemes to the solution of the shallow water equations on unstructured meshes // Journal of Computational Physics. 2007. Vol. 222, no. 1. P. 287–331. Access mode: http://doi.org/10.1016/j.jcp.2006.06.024.

[20] Иванов А. В. Вычислительный комплекс для моделирования морских течений с применением регуляризованных уравнений мелкой воды // Математическое моделирование. 2021. Т. 33, № 10. С. 109–128. Режим доступа: http://doi.org/10.20948/mm-2021-10-08.

[21] Булатов О. В. Численное моделирование течений в приближении мелкой воды на основе регуляризованных уравнений: дис. ... канд. физ. мат. наук: 05.13.18. М., 2014.

[22] Liang Qiuhua, Borthwick Alistair G. L. Adaptive quadtree simulation of shallow flows with wet–dry fronts over complex topography // Computers & Fluids. 2009. Vol. 38, no. 2. P. 221–234. Access mode: http://doi.org/10.1016/j.compfluid.2008.02.008.

Оглавление