

**Тема работы:** моделирование процессов соляного диапиризма ползущим течением, происходящим под действием архимедовых сил, на основе аналитического решения квазистационарной краевой задачи для однородно-вязкой ньютоновской жидкости, занимающей полупространство со свободной границей.

**Состав коллектива:**

1. Абрамов Тимофей Владимирович, м.н.с. ИНГГ, аспирант НГУ, программист;
2. Конторович Владимир Алексеевич, д.г.-м.н., чл.-корр. РАН, руководитель;
3. Лаврентьев Михаил Михайлович, д.ф.-м.н., проф., руководитель;
4. Лунев Борис Валентинович, к.ф.-м.н., с.н.с., консультант.

**Научное содержание работы:**

а) **Постановка задачи:** создание программы для быстрого численного моделирования соляного тектогенеза (соляного диапиризма) 3D ползущим течением высоковязкой ньютоновской жидкости, которая может быть использована для воспроизведения эволюции реальных структур соляной тектоники.

б) **Современное состояние проблемы:** В последнее время довольно часто в качестве математической модели для различных процессов в недрах Земли рассматривается медленное течение неоднородной (в первую очередь, с различной плотностью) вязкой несжимаемой ньютоновской жидкости в поле силы тяжести. В частности, это относится к моделированию развития осадочных бассейнов, к исследованию тепловой конвекции в мантии, изучению влияния континентов на структуру тепловых потоков в мантии и многим другим задачам.

Одной из наиболее простых, и в то же время интересных и практически значимых является задача моделирования процессов соляной тектоники (соляного диапиризма). Под этим термином подразумевается всплывание твердой каменной соли (в геологическом масштабе времени, т. е. десятки и сотни миллионов лет), вызванный внутренней гравитационной неустойчивостью неоднородной среды — соль имеет невысокую плотность и практически не уплотняется по мере захоронения под другими осадками (в отличие от этих осадков). Таким образом, согласно современным представлениям, это не что иное как частный случай неустойчивости Рэлея-Тейлора и во многих работах (Исмаил-заде, Короткий, Цепелев, Наймарк, Коуи и др.) моделируется ползущим течением несжимаемой ньютоновской жидкости.

Соляной тектогенез привлекает пристальное внимание геологов и геофизиков, в особенности специалистов по деформациям осадочных пород, горных инженеров, разведчиков недр и инженеров по подземным сооружениям. Академический интерес вызван уже тем, что всплывающая соль создает контрастные, высокоамплитудные складки в осадочном слое земной коры. С практической точки зрения, с соляной тектоникой так или иначе связано большинство крупных залежей углеводородов практически во всех крупных нефтегазоносных провинциях: всплывающая соль является существенным структурообразующим фактором в осадочных бассейнах, а также может образовывать ловушки для нефти и газа.

К сожалению, традиционные геофизические методы разведки, такие как сейсмо- или гравиразведка, не всегда результативны в их изучении, они не всегда дают полную и достоверную информацию. Дополнить и исправить ее может помочь численное моделирование. Под изучением реальной геологической структуры методом численного моделирования подразумевается воспроизведение ее эволюции по известному (как правило, частично) современному состоянию структуры (а также другой геологической информации) путем решения множества задач прямого моделирования с различными исходными параметрами модели. Подобрать их так, чтобы конечное состояние соответствовало реальному там, где оно хорошо известно, можно с высокой степенью уверенности говорить о соответствии модели реальной истории эволюции, и восстановить таким образом недостающие современные данные. Поскольку есть необходимость «насчитывать» большое число различных вариантов эволюции, программный инструмент для численного

моделирования соляного тектогенеза должен быть очень эффективным, чтобы справляться с каждым за относительно небольшое время (часы или несколько десятков часов).

Однако трехмерное численное моделирование геофизических процессов, адекватное реальным процессам сопряжено с большими объемами вычислений, посильными разве что для высокопроизводительных вычислительных кластеров. Поэтому, во-первых, необходимо, чтобы математическая модель допускала численное решение с распараллеливанием на большое множество узлов, а во-вторых, нужна эффективная программная реализация с учетом конкретной архитектуры вычислительной машины. К сожалению, в наиболее известных работах по 3D моделированию соляного тектогенеза (например, Исмаил-заде и др., 2000) численное решение трудно назвать эффективным. Решение с помощью разностных схем сводилось к решению систем ЛАУ (линейных алгебраических уравнений) с количеством переменных, пропорционально количеству узлов сетки, таким образом, вычислительная сложность алгоритма составила  $O(N^3)$ . Из-за этого за разумное время удавалось сделать расчет только на очень небольших сетках порядка  $25 \times 25 \times 25$  узлов, и даже параллельные вычисления не сильно помогали из-за высокой вычислительной сложности алгоритма и малой эффективности его распараллеливания. Это, конечно же, очень мало для моделирования эволюции реальной солянокупольной структуры.

По этим причинам трехмерное численное моделирование соляного тектогенеза не было рабочим инструментом для анализа этого процесса. Даже в достаточно новых работах продолжает использоваться [Ашраф, 2012] используется физическое и численное 2D моделирование.

в) **Подробное описание работы.** В качестве эффективной модели для численного моделирования соляного тектогенеза ползущим течением высоковязкой ньютоновской жидкости была использована предложенная Лунёвым (Лунёв, Лапковский 2009), которая основана на аналитическом решении квазистационарного уравнения движения жидкости в форме Стокса в полупространстве со свободной поверхностью, в предположении однородной вязкости среды. Для соответствующей краевой задачи было найдено аналитическое выражение функции Грина (Лунёв, 1986), таким образом, решение квазистационарной задачи ищется в виде свертки этой функции с распределением плотности в полупространстве. Прямое вычисление свертки занимает  $O(N^2)$  времени, и может быть вычислено за  $O(N \log N)$  с помощью быстрых алгоритмов. В обоих случаях, это кардинально быстрее решения систем ЛАУ, которые занимают  $O(N^3)$  времени.

Формально решается задача о происходящем под действием силы тяжести ползущем течении множества несмешивающихся несжимаемых ньютоновских жидкостей с различной плотностью и однородной, но очень высокой вязкостью ( $\sim 10^{20}$  Па · с), занимающих ограниченное свободной поверхностью полупространство. Такая постановка задачи для описания процесса соляного диапиризма подробно рассмотрена в (Лунёв, Лапковский 2009), там же обоснована возможность использования однородной вязкости.

На рис. 1 схематично показано полупространство в декартовой системе координат  $x_1, x_2, x_3$  заполненное жидкостями  $W_1, \dots, W_4$ , со своими плотностями  $\rho_1, \dots, \rho_4$  и единой вязкостью  $\mu$ . Они разделены границами  $S_1, \dots, S_3$ ,  $F$  – свободная поверхность.

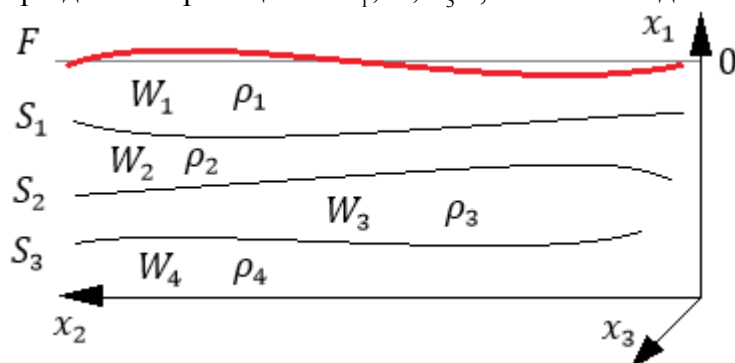


Рис. 1 — Полупространство, занятое жидкостями

Распределение плотности в полупространстве в момент времени  $t$  однозначно задается актуальной конфигурацией границ  $S_i: \rho(\mathbf{x}, t) = \rho_i$ , для  $\mathbf{x} \in W_i$ . Плотность  $\rho$ , тензор напряжений  $\mathbf{T}$  и давление  $P$  представляются в виде:

$$\begin{aligned}\rho(\mathbf{x}, t) &= \rho^0(x_3, t) + \sigma(\mathbf{x}, t) \\ \mathbf{T}(\mathbf{x}, t) &= \mathbf{T}^0(x_3, t) + \boldsymbol{\tau}(\mathbf{x}, t), \\ P(\mathbf{x}, t) &= P^0(x_3, t) + p(\mathbf{x}, t)\end{aligned}\quad (1)$$

где  $\rho^0(x_3, t)$ ,  $\mathbf{T}^0(x_3, t)$  и  $P^0(x_3, t)$  характеризуют гидростатическое состояние, а  $\sigma(\mathbf{x}, t)$ ,  $\boldsymbol{\tau}(\mathbf{x}, t)$  и  $p(\mathbf{x}, t)$  – отклонения от него. Для гидростатического состояния скорость течения  $\mathbf{v}^0 \equiv 0$ , так что течение  $\mathbf{v}$  связано только с возмущениями. Показано (Лунёв, Лапковский 2009), что течение задача вычисления ползущего течения в весоном полупространстве со свободной границей может быть сформулирована относительно возмущений с разделением на квазистационарную и эволюционную части в виде, линейризованном по граничным условиям:

Квазистационарная часть:

$$\begin{aligned}\mu \nabla^2 \mathbf{v} - \nabla p &= -\sigma \mathbf{g}; \\ \nabla \cdot \mathbf{v} &= 0; \\ (v_3 = \tau_{31} = \tau_{32} = 0)_{x_3=0},\end{aligned}\quad (2)$$

С дополнительным условием определения возмущения свободной поверхности

$$F = x_3 - h(x_1, x_2) :$$

$$(\tau_{33})_{x_3=0} = -\rho^0 g h. \quad (3)$$

Эволюционная часть:

$$\frac{\partial S_i}{\partial t} + \mathbf{v} \cdot \nabla S_i = 0. \quad (4)$$

Расчет ползущего течения в каждый момент времени  $t^{(k)}$  происходит в два этапа: сначала отыскивается поле течения  $\mathbf{v}^{(k)}$ , которое задается распределением  $\rho(\mathbf{x}, t^{(k)})$ , определяемым текущей конфигурацией границ  $S_i^{(k)}$ . На втором этапе решается эволюционное уравнение, в результате чего границы перемещаются полем  $\mathbf{v}^{(k)}$  в новое положение  $S_i^{(k+1)}$ . Новая конфигурация границ задает новое распределение плотности  $\rho(\mathbf{x}, t^{(k+1)})$ , из которого рассчитывается новое поле  $\mathbf{v}^{(k+1)}$ . Таким образом, эволюция описывается набором квазистационарных состояний, связанных эволюционным уравнением. Для задачи (2) в (Лунев, 1986) было найдено аналитическое выражение функции Грина, что позволяет отыскивать поле течения как ее свертку с распределением плотности:

$$\mathbf{v}(\mathbf{x}) = \iiint V(\mathbf{x}, \boldsymbol{\xi}) d\xi_1 d\xi_2 d\xi_3 \quad (5)$$

где  $V$  – функция Грина, а  $\mathbf{x}$ ,  $\boldsymbol{\xi}$  – декартовы координаты точек пространства.

Такой подход имеет множество преимуществ по сравнению с использованием разностных схем. Во-первых, поскольку это точное решение задачи, повышается точность расчета. Во-вторых, по формуле (5) можно рассчитывать  $\mathbf{v}$  в любой точке полупространства, независимо от других. Это понижает размерность задачи, так как для решения эволюционного уравнения (4) необходимо вычислять скорость только на границах  $S_i$ . Но основным преимуществом является скорость расчета из-за меньшей алгоритмической сложности: прямое вычисление свертки занимает  $O(N^2)$  (вместо  $O(N^3)$  для разностных методов) времени, и может быть эффективно распараллелено по аналогии с задачей N-тел. Параллельная реализация позволяет достигать производительности, близкой к пиковой, в том числе на GPU. Также есть способы расчета свертки за  $O(N \log N)$  операций с помощью быстрых алгоритмов.

г) **Полученные результаты.** Было реализовано два основных подхода к численному решению задачи — прямое вычисление свертки (5) на большом количестве графических процессоров и применение быстрых алгоритмов расчета свертки с расчетам на различных вычислительных машинах.

По первому подходу были получены результаты:

1. Промоделирован вычислительный эксперимент из (Исмаил-заде и др., 2000), где моделировался рост одиночного соляного диапира. На рис. 2 показаны расчеты из настоящей работы

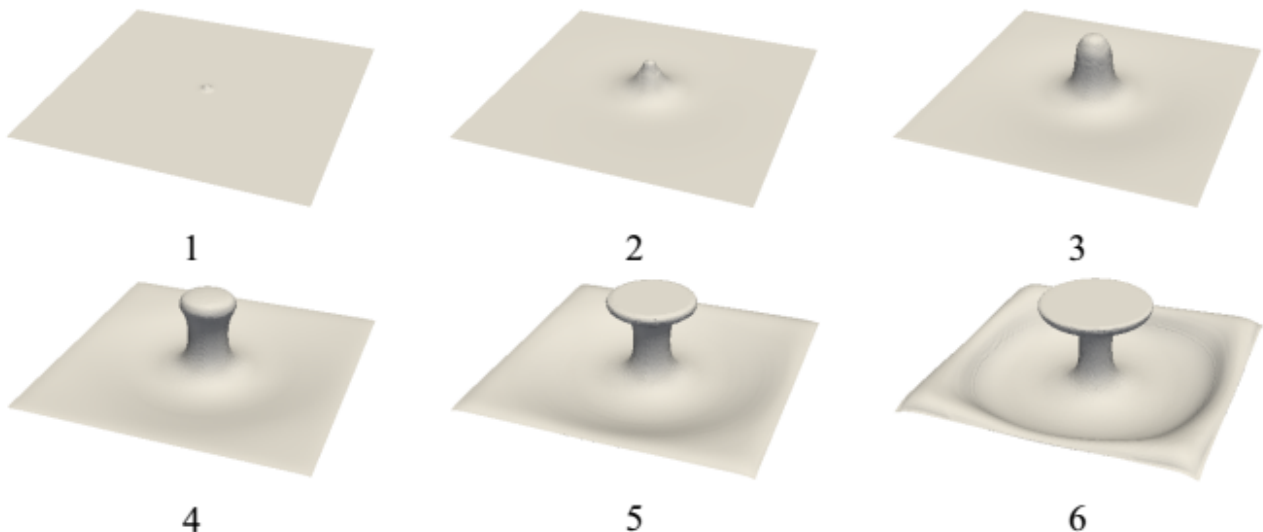


Рис. 2 — Моделирование эволюции одиночного возмущения, сделанное в данной работе, показаны шесть стадий

На рис. 3 показаны в сравнении конечные стадии эволюции из настоящей работы и из (Исмаил-заде и др., 2000):

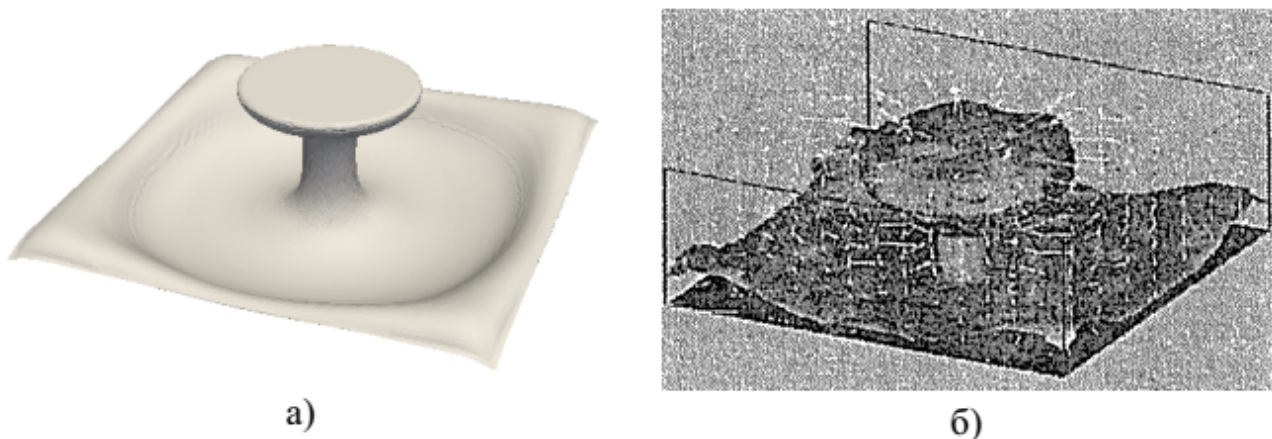


Рис. 3 — Моделирование эволюции одиночного возмущения. Показаны результаты из данной работы (а) в сравнении с результатами из (Исмаил-заде и др., 2000) (б). Модельное время 15.6 млн лет.

Поскольку исходные данные были заданы идентично, результаты совпадают не только качественно, но и количественно.

Расчет всей эволюции в настоящей работе занял приблизительно 42 минуты на одной Tesla M2090 (170 шагов по времени, примерно 15 секунд на шаг). Размер сетки 120 x 120 x 60, среднее количество точек описания границ в процессе счета 255349. Для сравнения, в (Исмаил-заде и др., 2000) расчет одного шага по времени занимал 12 минут на 16-и процессо рах IBM SP2. Для расчета поля течения использовалась сетка 25 x 25 x 25

узлов.

2. Промоделирован физический эксперимент эксперимент из (Талбот, Джексон, 1987), где моделировался эволюция произвольно возмущенного слоя неустойчивой жидкости.

Пример физического моделирования показан на рис. 4, а. Однако у автора не было ее подробного описания, поэтому численно воспроизвести эволюцию получилось только качественно. Результат показан на рис. 4, б.

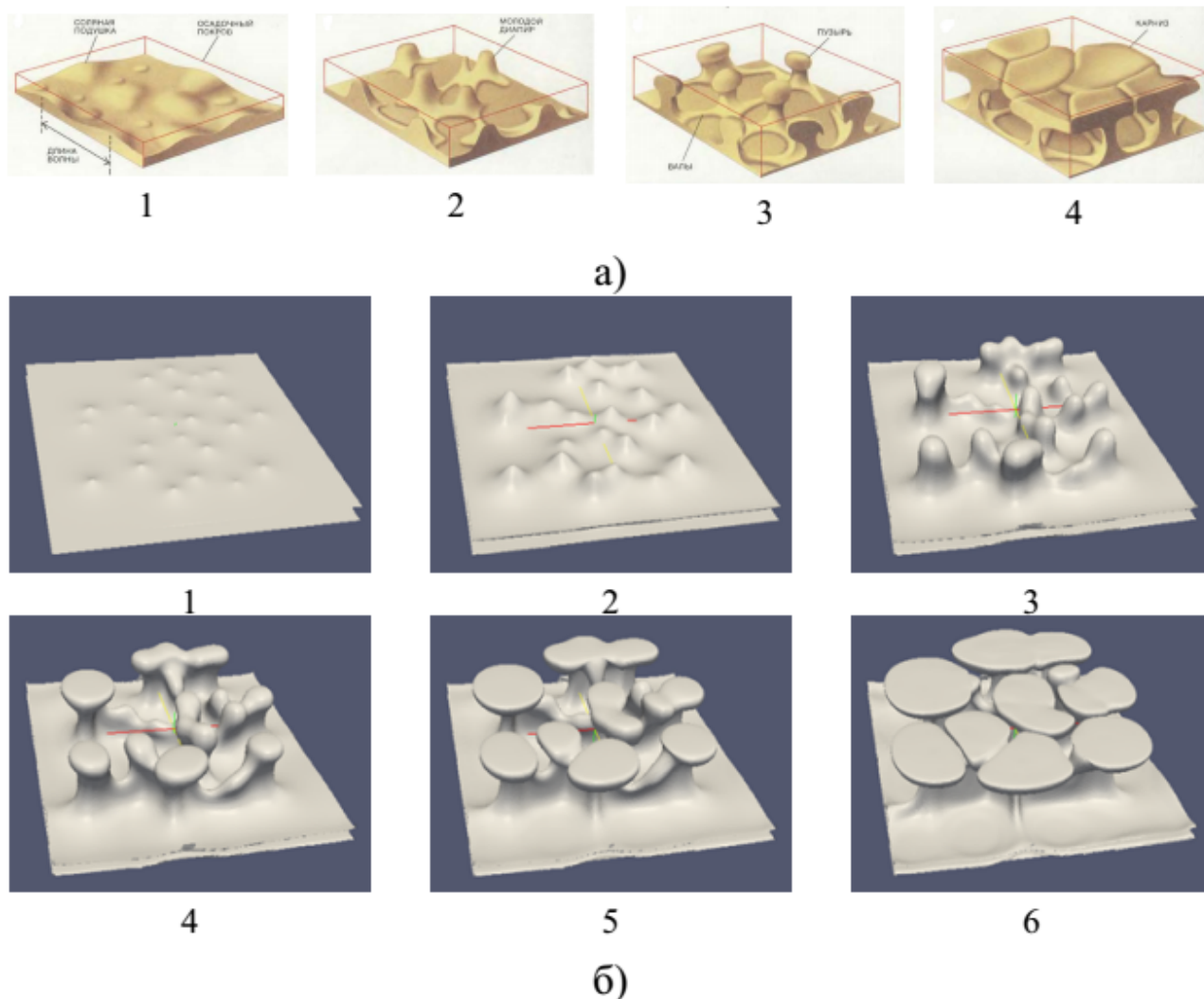


Рис. 4 — Моделирование всплытия легкой соли, возмущенной случайным образом во множестве мест. а — физическое моделирование из (Талбот, Джексон, 1987), б — численное из данной работы

Параметры численной модели были заданы те же, что и для модели на рис. 2 и 3, кроме характера возмущения неустойчивого слоя. Время расчета также составило порядка 40 минут на одной Tesla M2090. Качественное сходство результатов очевидно.

3. Протестирована масштабируемость программной реализации на множестве графических процессоров Tesla M2090 Информационно-вычислительного центра (ИВЦ) Новосибирского государственного университета.

Очевидно, что тестирование масштабируемости заключается в запуске программы на различном количестве графических процессоров и проверке скорости ее работы. В идеале скорость должна увеличиваться пропорционально количеству задействованных GPU, но в действительности производительность растет нелинейно и с некоторого момента может и падать при увеличении числа вычислительных узлов.

Однако программа содержит множество операций, из которых вычислительно наиболее

сложная — это расчет свертки, но есть также работа с полигональной сеткой, создание регулярной сетки «источников», и множество других вспомогательных операций, например запись результатов на жесткий диск. Поэтому описывать производительность программы лишь одной кривой (т. е. учитывая все операции) не совсем корректно. Поэтому для каждого теста используется три графика:

- «kernel»: скорость работы «вычислительного ядра» программы, т. е. расчета свертки на GPU. Никаких других операций не учитывается, только непосредственно расчет на графическом процессоре;
- «mpi»: также скорость работы вычислительного ядра, но с учетом времени пересылки данных по MPI. Берется промежуток времени от начала расчета поля течения (рассылки данных по вычислительным узлам) до того, как главный узел соберет насчитанные данные от всех остальных (т. е. до того как на головном узле окажется рассчитанное поле течения целиком);
- «all»: скорость работы программы в целом, включая увеличение дета лизации поверхностей, получение из них регулярной сетки аномальной плотности, расчет поля течения, решение эволюционного уравнения, сохранение результатов в долговременную память и т. д.

Первая серия тестов была проведена для модели с сеткой задания распределения плотности  $64 \times 64 \times 64$  узлов, изначально 4 поверхности, которые будут погружаться и добавляться еще поверхности. Эволюция была рассчитана до примерно 330 млн. лет. Это очень небольшая задача, не загружающая полностью даже одну Tesla M2090. Графики показаны на рис. 5.

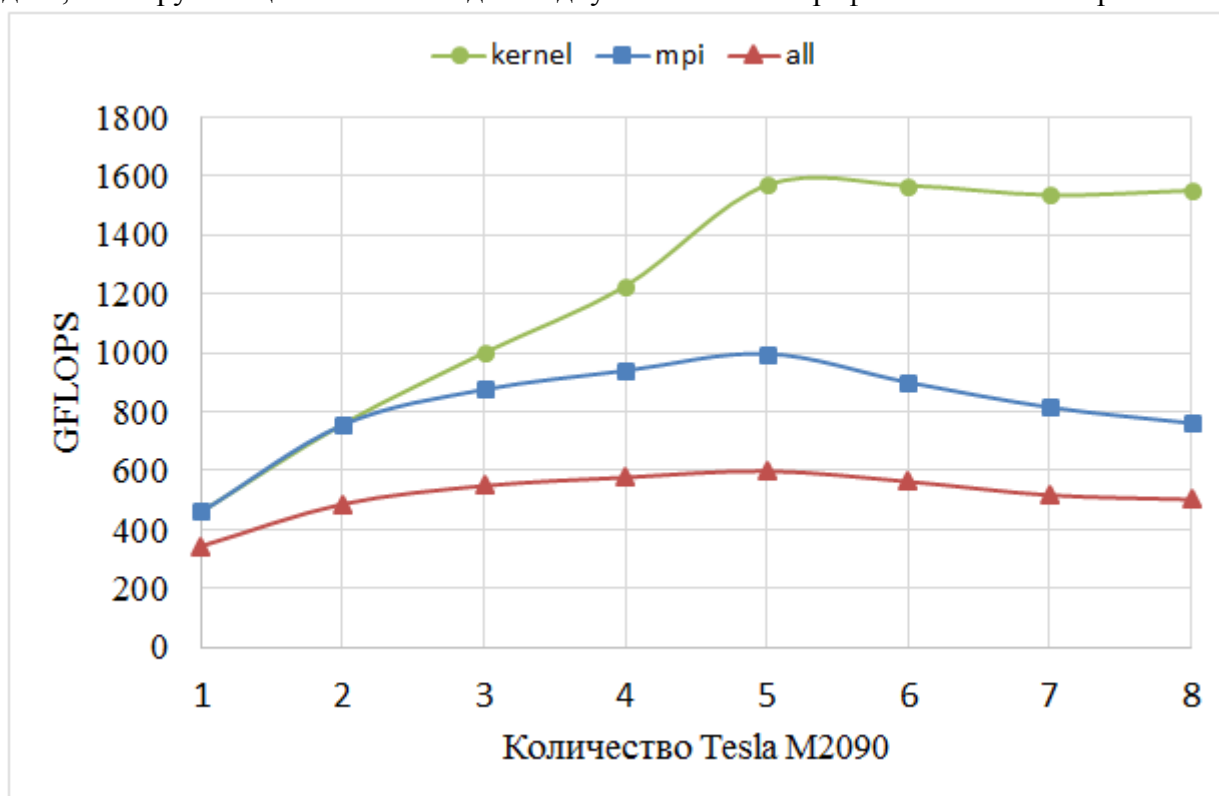


Рис. 5 — Проверка работы программы при небольшом объеме вычислений. «kernel» — средняя скорость выполнения непосредственно GPU-ядра программы, «mpi» — средняя скорость расчета полного поля течения с помощью нескольких вычислительных узлов по протоколу MPI (включает время на обмены данными между узлами), «all» — среднее время выполнения шага эволюции (включает работы у сетками и поверхностями)

Данный график показывает проблему масштабируемости приложения, работающих на множестве вычислительных узлов. В данном случае использование более пяти Tesla дает отрицательный прирост скорости. Это происходит из-за небольшого размера вычислительной задачи — весь расчет эволюции (162 итерации) вместе с сохранением промежуточных

состояний на жесткий диск потребовал около 248 секунд. График для более крупных задач будет подобен, но его «выполаживание» и отрицательный прирост производительности будет происходить на большем количестве Tesla.

Вторая серия экспериментов решает ту же задачу, но более точно, на сетке  $128 \times 128 \times 128$  узлов. Графики показаны на рис. 6.

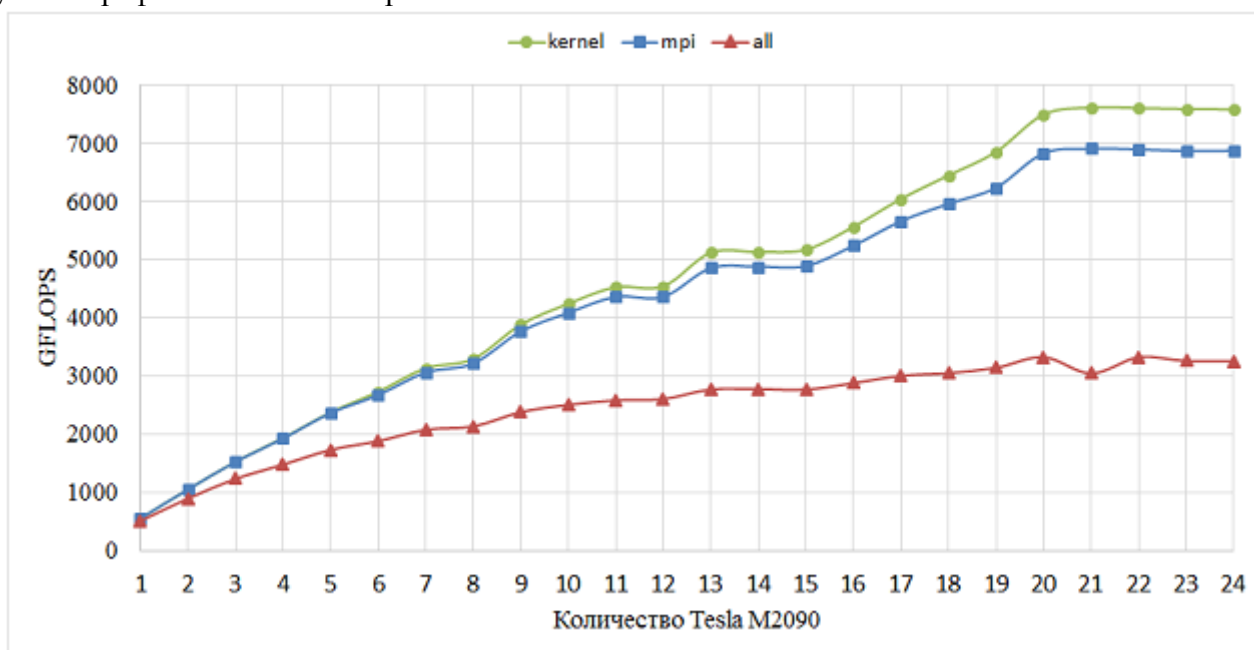


Рис. 6 — Проверка работы программы при значительном объеме вычислений

Здесь выхолаживание графика происходит значительно позже и удается добиться значительной реальной производительности. На рис. 7 показаны в сравнении два расчета — на грубой и точной сетке

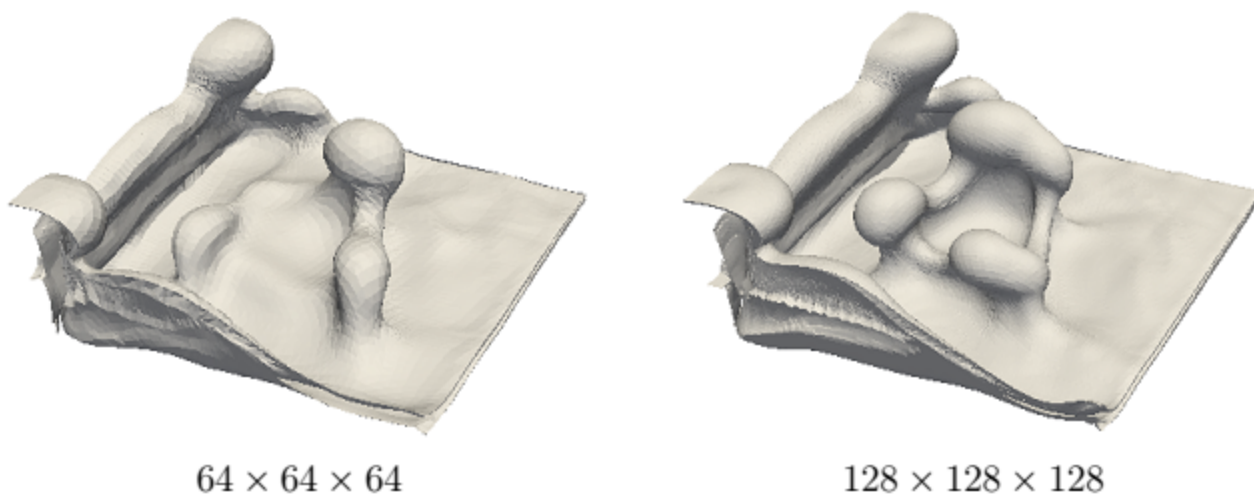


Рис. 7 — Расчеты на грубой ( $64 \times 64 \times 64$ ) и точной сетке ( $128 \times 128 \times 128$ )

При качественном сходстве есть заметные различия, что показывает необходимость расчета на густых сетках для большей точности.

4. Был реализован быстрый алгоритм для расчета свертки, который работает намного быстрее непосредственного, даже распараллеленного на множестве GPU. В настоящее время используется библиотека `fftw` для быстрых Фурье-преобразований, которые необходимо делать для быстрого расчета свертки. На рис. 8 показано, насколько быстрее работает быстрый алгоритм.

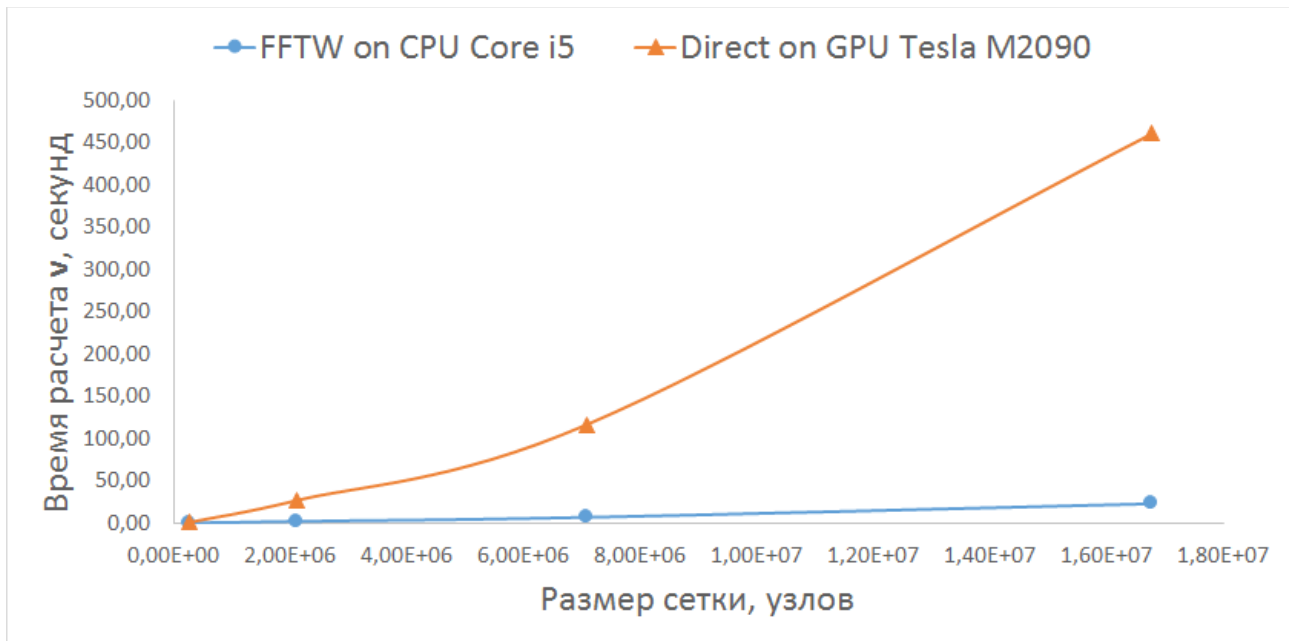


Рис. 8 — Сравнение времени работы прямого расчета свертки на одном GPU (Tesla M2090) и одном центральном процессоре Intel Core i5.

Рис. 8 наглядно показывает, что быстрый алгоритм со сложностью  $O(N \log N)$  даже на центральном процессоре работает значительно быстрее чем прямой расчет свертки  $O(N^2)$  на GPU, при том что реализацию на GPU можно считать оптимальной.

На настоящий момент использование графических ускорителей для данной задачи уже не является актуальным, вместо этого применяется быстрый метод с расчетом на центральных процессорах. Однако он требует значительных объемов оперативной памяти для работы, мало зависит от количества процессоров и их ядер и более эффективен при работе на вычислительной машине с общей памятью. Поэтому для его работы больше подходят сервера с большими объемами оперативной памяти. На кластере для тестирования метода было сделано несколько расчетов:

Количество узлов сетки	Количество шагов по времени	Время расчета, часов	Очередь кластера и количество оперативной памяти
$192 \times 256 \times 48 \approx 2.3 \cdot 10^6$	268	0.75	x1230g9q, 2 Гбайт
$384 \times 512 \times 96 \approx 1.8 \cdot 10^7$	516	11.1	x1230g9q, 8 Гбайт
$768 \times 1024 \times 192 \approx 1.5 \cdot 10^8$	1017	70.96	x1230g9q, 192 Гбайт
$4096 \times 4096 \times 64 \approx 1.0 \cdot 10^9$	196	97.30	vkop2q, 1.5 Тбайт

Узлы с большим количеством оперативной памяти позволили протестировать метод и получить решение на сетках очень большого размера, до миллиарда узлов. Конечная стадия эволюции модели при расчете на сетке с миллиардом узлов ( $4096 \times 4096 \times 64 \approx 1.0 \cdot 10^9$ ) показана на рис. 8.

**Эффект от использования кластера в достижении целей работы.** На вычислительном кластере удалось проанализировать эффективность различных подходов к численному решению задачи. Метод, основанный на непосредственном расчете самой ресурсоемкой операции — вычислении свертки, был распараллелен на множество графических процессоров (до 24 Tesla M2090), где были достигнуты пределы масштабируемости на некоторых сетках.



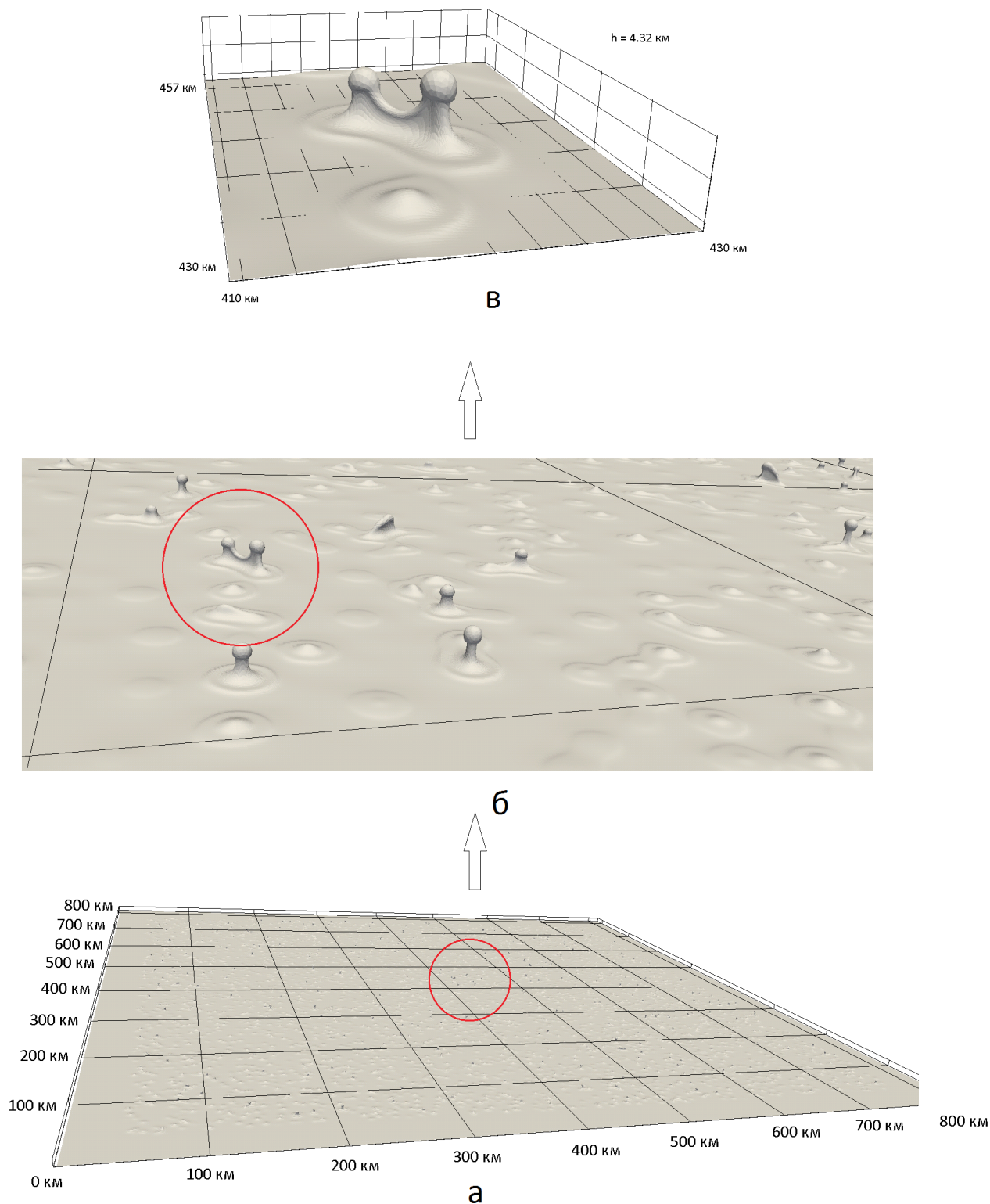


Рис. 8 — Расчет на сетке с миллиардом ( $4096 \times 4096 \times 64 \approx 1.0 \cdot 10^9$ ) узлов. Под а — вся поверхность всплывающего слоя ( $800 \times 800$  км.), б — участок  $100 \times 100$  км., в — отдельно стоящий «сдвоенный» диапир.

Быстрый метод ориентирован на выполнение на вычислительной машине с большим объемом собственной оперативной памяти. Расчеты на очень больших сетках нельзя было бы

сделать на настольном персональном компьютере. С помощью быстрого метода удалось сделать расчеты эволюции на сетках до 100 миллионов узлов на сервере с 192 Гб оперативной памяти и до 1 миллиарда узлов на сервере с 2 Тб памяти (было использовано 1,5 Тб).

#### **Перечень публикаций, содержащих результаты работы.**

1. Лунёв Б.В., Абрамов Т.В. Высокоэффективные вычисления 3-D ползущих течений для оперативного моделирования соляного тектогенеза // Недропользование. Горное дело. Направления и технологии поиска, разведки и разработки месторождений полезных ископаемых. Геоэкология. — Т. 2. — Новосибирск: 2014. (сборник материалов конференции);
2. Лунёв Б.В., Лапковский В.В., Абрамов Т.В. Численное моделирование развития инверсионных складок в подсолевых слоях // Недропользование. Горное дело. Направления и технологии поиска, разведки и разработки месторождений полезных ископаемых. Геоэкология. — Т. 2. — Новосибирск: 2014. (сборник материалов конференции);
3. Абрамов Т.В., Лунёв Б.В. Моделирование соляного диапиризма расчетом трехмерных ползущих течений с использованием технологии параллельных вычислений CUDA на GPU // Всероссийская конференция молодых ученых по математическому моделированию и информационных технологиям. — Тюмень: 2014. (электронный ресурс);
4. B.V. Lunev, T.V. Abramov. Fast parallel method for computing of 3D creeping flows on GPU with CUDA, modeling the salt diapirism // International conference advanced mathematics, computations and applications. — 2014. (сборник материалов конференции);
5. Абрамов Т.В., Лунёв Б.В. Моделирование соляного диапиризма расчетом трехмерных ползущих течений с использованием технологии параллельных вычислений CUDA на GPU // CUDA альманах. — 2014. — С. 10.
6. Абрамов Т.В. Использование технологии параллельных вычислений NVIDIA CUDA на GPU для численного моделирования неустойчивости Релея-Тейлора в высоковязкой ньютоновской жидкости // Параллельные вычислительные технологии 2015. (сборник материалов конференции);
7. Абрамов Т. В. Массивно-параллельный расчет неустойчивости Релея-Тейлора с помощью аналитического выражения функции Грина соответствующей краевой задачи // Вычислительные технологии. — 2015. — Т. 20, № 4. — С. 3 – 16. (zbMATH, AMS MathSciNet, Ulrich's Periodicals Directory, Web of Science: RSCI (Russian Scientific Citation Index at Web of Science), РИНЦ, Перечень ВАК РФ);
8. Абрамов Т.В. Использование теоремы о свертке в задаче моделирования неустойчивости Релея-Тейлора // УМ-2015. — Красноярск: 2015. (электронный ресурс)
9. Абрамов Т.В., Лунёв Б.В. Использование теоремы о свертке для ускорения численного моделирования процессов соляного диапиризма // АМСА-2015. (сборник материалов конференции);
10. Абрамов Т.В. Оптимизация численного решения краевой задачи с известной функцией Грина для быстрого моделирования процессов соляной тектоники // Трофимуковские чтения 2015. — 2015. (сборник материалов конференции);
11. Абрамов Т. В. Быстрое численное решение краевых задач с известной функцией Грина через циклическую свертку // Вычислительные технологии. — 2016. — Т. 21, № 2. — С. 3 – 11. (zbMATH, AMS MathSciNet, Ulrich's Periodicals Directory, Web of Science: RSCI (Russian Scientific Citation Index at Web of Science), РИНЦ, Перечень ВАК РФ).

**Ваши впечатления от работы вычислительной системы и деятельности ИВЦ НГУ, а также Ваши предложения по их совершенствованию.** Впечатления от работы на ИВЦ НГУ положительные, не в последнюю очередь благодаря хорошему, понятному описанию принципов работы на вычислительном кластере, продуманной и удобной организации работы кластера в целом, и, в частности, отзывчивости и оперативности Владислава Анатольевича Калужного, ведущего инженера НГУ. Предложений по совершенствованию работы кластера у меня нет.