

УДК: 519.6

## Алгоритмическое и программное обеспечение решения задач взаимодействия конструкции с жидкостью/газом на гибридных вычислительных системах

И. М. Кузьмин<sup>а</sup>, Л. Е. Тонков<sup>б</sup>, С. П. Копысов<sup>с</sup>

Институт механики УрО РАН,  
Россия, 426067, г. Ижевск, ул. Т. Барамзиной, д. 34

E-mail: <sup>а</sup> imkuzmin@gmail.com, <sup>б</sup> tnk@udman.ru, <sup>с</sup> s.kopysov@gmail.com

Получено 3 апреля 2013 г.

Рассматривается создание прикладного программного интерфейса с выделением самостоятельного приложения для синхронизации и обмена данными, в котором реализуются отдельные подзадачи связывания для решения сопряженных задач взаимодействия конструкции с жидкостью или газом. Обсуждаются алгоритмы связывания подзадач и деформирования расчетных сеток. На численных примерах показывается возможность решения ряда задач на кластерах с графическими процессорами.

Ключевые слова: моделирование взаимодействия газа и деформируемого тела, параллельные вычисления, гибридные вычислительные системы, деформация сеток, радиальные базисные функции

## Algorithms and Software for Solving Coupled Fluid-Structure Interaction Problems on Hybrid HPC Platform

I. M. Kuzmin, L. E. Tonkov, S. P. Kopysov

*Institute of Mechanics UB RAS, 34 T. Baramzinoy st., Izhevsk, 426067, Russia*

**Abstract.** — In this paper, we propose a new software for simulation of fluid-structure interaction. The software is designed for solving coupled problems and provides an interface for synchronization and data exchange between existing fluid and structural solvers. Algorithms of coupling solvers and mesh deformation are discussed. The software can be used on hybrid CPU/GPU platforms.

**Keywords:** fluid-structure interaction, parallel computing, hybrid HPC platforms, computational mesh deformation, radial basic function

**Citation:** *Computer Research and Modeling*, 2013, vol. 5, no. 2, pp. 153–164 (Russian).

Работа выполнена при финансовой поддержке РФФИ (грант 11-01-00275-а, 12-07-00080-а) и программы Президиума РАН №18 при поддержке УрО РАН (проект 12-П-1-1005).

## Введение

При решении сопряженных задач взаимодействия потока газа и деформируемой конструкции (FSI — Fluid-Structure Interaction) в рамках разделенного подхода [Farhat, Lesoinne, 2000], согласование получаемых решений на границах подобластей является неотъемлемым и весьма важным этапом, существенно влияющим на адекватность получаемых результатов. Под согласованием понимается прежде всего совокупность операций по обмену результатами вычислений между соответствующими решателями физических подзадач, интерполяции значений между узлами расчетных сеток, деформирование сетки газодинамической области и, возможно, реализация семейства итерационных процедур при жестком связывании.

Актуальным является построение параллельных алгоритмов как решения отдельных физических подзадач динамики деформирования и газодинамики, так и непосредственно решающих всю совокупность задач сопряжения: интерполяция, деформирование расчетных сеток, итерационное связывание и создание прикладного программного интерфейса с выделением самостоятельного приложения для синхронизации и обмена данными, в котором реализуются отдельные подзадачи связывания. Таким образом, можно считать, что мультидисциплинарный программный комплекс должен строиться по архитектуре клиент-сервер, где каждое приложение-сервер эффективным образом распараллелено, решает свою узкоспециальную прикладную задачу и обменивается расчетными данными через клиента с другими серверами. Подход обеспечит наибольшую универсальность и позволит объединять независимые приложения, от которых требуется лишь использование соответствующего прикладного программного интерфейса. В данной работе представлены различные алгоритмы связывания при решении задач FSI, схемы взаимодействия различных приложений и определяются затраты, необходимые для обмена данными. Приводится анализ существующих программных решений, структура и функции разрабатываемого программного обеспечения для FSI, а также рассматриваются параллельные алгоритмы, реализованные в связующем программном обеспечении (ПО).

## Вычислительные алгоритмы связывания в задачах FSI

Существуют два основных подхода для решения таких задач: совместный (монолитный) подход и разделенный (последовательный) подход. Монолитный подход лучше отражает физику явления, поскольку на каждом шаге по времени он позволяет получить согласованное решение для газа и конструкции. Однако этот подход обладает рядом существенных недостатков. Во-первых, одновременное решение всех уравнений достаточно громоздко и требует большого количества времени и ресурсов кластера. Во-вторых, такой метод требует общий математический аппарат для решения всех уравнений. Однако, на сегодняшний день большинство решателей для моделирования газовой динамики используют метод конечного объема, а решатели для моделирования деформирования конструкции — метод конечных элементов. Использование данных решателей для моделирования задач, для которых они не предназначены, может привести к снижению точности получаемого решения. Поэтому применимость монолитного подхода для решения задач взаимодействия потока с деформируемой конструкцией крайне ограничена.

Последовательный метод позволяет использовать для решения газодинамического и прочностного блоков различные готовые решатели, разработанные именно для данных задач. Наиболее общеупотребительный вариант разделенного подхода последовательный. При этом каждое приложение решается последовательно до определенного времени, после чего полученные данные пересылаются в другое приложение. Однако как все явные методы разделения, они могут проявлять неустойчивость, поскольку на каждом временном шаге обмена информацией оба решения не согласованы между собой. Для большинства типов схем разделения согласование

происходит на уровне расположения границ областей и расчетных нагрузок на несогласованных сетках.

Алгоритмы, обеспечивающие двустороннюю связь между газодинамической задачей (CFD) и задачей деформирования конструкции (CSD), подразделяют по степени сопряжения на «слабо» и «жестко» связанные. В первом случае передача данных между приложениями выполняется явно, т. е. нагрузка для задачи деформирования на шаге  $t^n$  зависит от давления газа на шаге  $t^{n+1}$ . В слабосвязанном случае, в пределах одного временного шага, граничные условия пересылаются только однажды между телом  $\Omega_s$  и областью, занятой газом  $\Omega_f$ , и нет никаких приближений, выполняемых отдельными задачами. В этом случае реализуется так называемое шахматное связывание, при котором программы обоих приложений не обязательно должны одновременно выполняться на вычислительном кластере.

В жесткосвязанном подходе используется несколько обменов на каждом временном шаге и тем самым реализуется неявное сопряжение, обеспечивающее более сильную связь между программными приложениями рассматриваемых задач. Здесь приложения одновременно выполняются на кластере и обмениваются между собой данными на каждой итерации до выполнения необходимых условий сходимости. Такие алгоритмы связывания включают решение нелинейных систем уравнений.

Не все условия связывания удовлетворяются в конце временного шага  $t = t^{n+1}$ , одна из задач не имеет верного граничного условия в момент времени  $t = t^{n+1}$ . Для простоты представления схемы связывания, разделенные задачи используют связанные степени свободы на интерфейсе, а именно, нагрузки  $\mathbf{f}_\Gamma$  и перемещение интерфейса  $\mathbf{u}_\Gamma$ . Тогда задачу деформирования можно представить через нелинейную функцию  $S$ , которая на временном шаге  $t^{n+1}$  связывает интерфейсные нагрузки, полученные от газа  $\mathbf{f}_\Gamma^{n+1}$ , с перемещениями на интерфейсе  $\mathbf{u}_\Gamma^{n+1}$ :

$$\mathbf{u}_\Gamma^{n+1} = S(\mathbf{f}_\Gamma^{n+1}).$$

Соответственно, газодинамическая задача может быть представлена в виде нелинейной функции  $F$  перемещений границы  $\mathbf{u}_\Gamma^{n+1}$  для вычисления соответствующих нагрузок

$$\mathbf{f}_\Gamma^{n+1} = F(\mathbf{u}_\Gamma^{n+1}).$$

Используя введенные обозначения, сопряженную задачу можно представить как систему нелинейных уравнений

$$\mathbf{u}_\Gamma^{n+1} = S[F(\mathbf{u}_\Gamma^{n+1})] = S \circ F(\mathbf{u}_\Gamma^{n+1}),$$

или

$$\mathbf{f}_\Gamma^{n+1} = F \circ S(\mathbf{f}_\Gamma^{n+1}).$$

Таким образом, для решения жесткосвязанной задачи необходимо обеспечить эффективное параллельное решение задачи газовой динамики и динамики деформирования, выполнить обмен данными минимального объема между задачами, провести интерполяцию давления и перемещений между несогласованными сетками и деформирование расчетной газодинамической сетки. Рисунок 1 показывает схему связывания.

Для решения задачи FSI, в случае схемы жесткого связывания, необходимо решить следующие задачи:

- использование эффективных решателей для задач CFD и CSD,
- обеспечение связи между модулями решения обеих задач,
- интерполирование давления и перемещений между сетками,
- деформирование сетки (CMD).

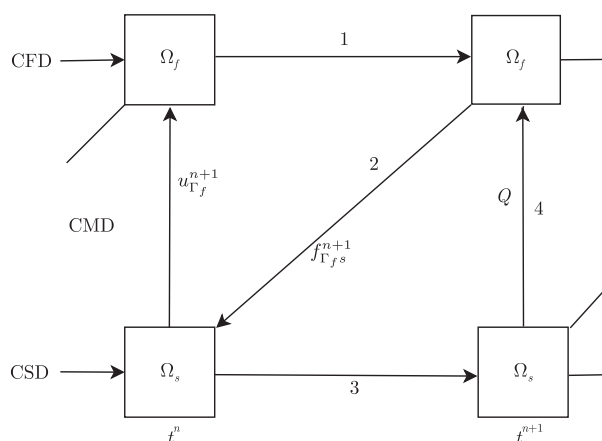


Рис. 1. Алгоритм решения задачи FSI

Решение этих задач представляется эффективным в рамках отдельного связывающего программного обеспечения и реализацией в нем некоторых вычислительных алгоритмов решения связанной задачи FSI.

## Существующие программные реализации

Анализ существующих решений прикладного программного обеспечения (ППО) для построения мультидисциплинарных приложений позволяет выделить несколько устоявшихся подходов. MpCCI (<http://www.mpcci.de/multiphysics-engineering.html>) — наиболее распространенный и функционально-наполненный коммерческий пакет для сопряжения мультидисциплинарных приложений. Система построена по архитектуре «клиент-сервер» и состоит из нескольких компонент. К сожалению, система является коммерческой и закрытой, кроме того поддерживает лишь ограниченный набор распространенных вычислительных приложений, значительную часть из которых составляют также коммерческие решения (ANSYS, Abaqus, Fluent, STAR-CD и т.п.).

CoMA (<http://carat.st.bv.tum.de>) — промежуточное ПО, обеспечивающее контроль и обмены между отдельными связываемыми приложениями. Наиболее важными функциями системы являются: распараллеливание и коммуникации, которые осуществляются под MPI с помощью выделения групп процессоров между отдельными приложениями и CoMA; обмен данными между распределенными сетками основан на поиске ближайших соседей и линейной интерполяции полей и потоков с выполнением условий консервативности; модуль контроля связывания обеспечивает различные варианты явного/неявного связывания и условий сходимости.

MCT (<http://www.mcs.anl.gov/research/projects/mct/>) — свободно распространяемое ППО, реализованное на объектно-ориентированном фортране, предназначено для узкого круга задач моделирования климата. MCT содержит классы для поддержки связывания параллельных приложений: классы поддержки описания данных; классы поддержки передачи данных; классы поддержки преобразования данных. Здесь можно особо выделить, что модуль преобразования данных состоит главным образом из алгоритмов интерполяции между различными сетками с разным разрешением, алгоритмы усреднения, вычисления интегралов по поверхностям и др.

OpenPALM [http://www.cerfacs.fr/globc/PALM\\_WEB/](http://www.cerfacs.fr/globc/PALM_WEB/) — недавно ставшее доступным ППО OpenPALM (CERFACS-ONERA) имеет следующую отличительную функциональность: динамическое связывание нескольких приложений; управление параллельными коммуни-

кациями (коммуникационная парадигма "конечной точки": передающий объект ничего не знает о получателях); двухуровневый параллелизм; достаточно сложный графический интерфейс для связывания приложений; явные схемы и утилиты интерполяции; утилиты линейной алгебры; блоки отладки и анализа ускорения.

preCICE [http://www5.in.tum.de/wiki/index.php/PreCICE\\_Webpage](http://www5.in.tum.de/wiki/index.php/PreCICE_Webpage) — ППО для решения задач FSI, использующее только структурированные сетки. ППО реализовано в технологии двухсторонних коммуникаций и обеспечивает полную поддержку разделенного подхода с учетом полной независимости от решателей в отдельных подобластях. Отображение данных основывается на быстром поиске соседей алгоритмами кривых заполняющих пространство.

FlowVision <http://www.flowvision.ru/> — коммерческое ППО для решения задач FSI выполнена компанией ТЕСИС для анализа задач взаимодействия жидкость-конструкция и основана на методе подсеточного разрешения геометрии, реализованном в FlowVision. К сожалению, ориентирована только на использование программных продуктов Abaqus и FlowVision.

Следует отметить, что единого универсального подхода на данный момент не существует и таковой вряд ли возможен, так как каждый научный коллектив имеет свои наработки, обладающие определенной спецификой, определяемой, в частности, кругом решаемых задач. В отличие от этих существующих инструментов, рассматриваемый подход стремится объединять компонентную технологию с эффективной технологией отображения данных, обеспечить применимость для широкого круга приложений, сеток и достичь прозрачных механизмов программного расширения, позволяющий в дальнейшем перейти к рассмотрению не только связывания сопряженных задач, но и связыванию многомасштабных задач.

## Структура связывающего приложения для решения задач FSI

Для отработки разработанного ПО FESudioFSI были решены две тестовые задачи FSI. В общем процедура решения каждой из них основывалась на максимально полном использовании ранее созданного кода, предназначенного для решения отдельных физических подзадач (CFD and CSD). При этом каждое отдельное приложение использовало свою модель параллелизма, независимую параллельную реализацию численных методов и алгоритмов и свои требования к среде параллельного выполнения. Разработанное ПО FESudioFSI позволило осуществить решение поставленных FSI задач не нарушая указанных ограничений и с минимальными изменениями исходного кода имеющихся приложений. Предлагаемый подход к моделированию задач взаимодействия потока газа и деформируемой конструкции/тела основан на выделении этапа согласования и взаимодействий в отдельную программную подсистему, не зависящую от используемых реализаций расчетов, газодинамических полей и напряженно-деформированного состояния. Таким образом может быть обеспечено взаимодействие для произвольных комбинаций программ решения задач газовой динамики и механики деформирования. В этом случае сопрягаются различные типы сеток, разные виды аппроксимаций. Точность аппроксимации уравнений сохраняется как в областях расчета газодинамики и механики деформирования, так и на границе взаимодействия.

В рассматриваемом случае система уравнений газовой динамики решается произвольным лагранжево-эйлеровым конечно-объемным методом в библиотеке OpenFOAM (<http://www.openfoam.com>), а на этапе решения задач деформирования применяется конечно-элементный пакет FESudio [Kopysov et al., 2004; Kopysov, Novikov, 2006] (<https://www.udman.ru/projects/FESudio/wiki>). Важно отметить, что оба пакета написаны в рамках объектно-ориентированного подхода и это существенно облегчает их совместное использование.



## Программная модель

Приложение FESudioFSI реализуется в соответствии с моделью «клиент-сервер» на объектно-ориентированном промежуточном ПО Ice Zeroc <http://www.zeroc.com/ice.html>. Ice является объектно-ориентированным ППО, то есть предоставляет средства для разработки объектно-ориентированных распределенных приложений. Клиентами называются активные сущности, запрашивающие определенные сервисы у сервера. Серверами называются пассивные сущности, предоставляющие сервисы в ответ на запросы клиентов. Модель программирования Ice основана на понятии Ice-объекта. Это абстрактная сущность, которая может отвечать на запросы клиентов, функционировать на одном сервере или сразу на нескольких. Каждый Ice-объект имеет один или несколько интерфейсов, имеет уникальный идентификатор. Ice-объект имеет как минимум один выделенный интерфейс, называемый главным интерфейсом.

Клиент использует для вызова Ice-объекта так называемый прокси (proxy). Прокси — это представитель Ice-объекта на стороне клиента, в его локальном адресном пространстве. Код прокси для определенного языка программирования генерируется Slice-компилятором. Прокси инкапсулирует в себе информацию, необходимую для вызова Ice-объекта: физический адрес сервера, идентификатор объекта и, дополнительно, идентификатор фасета. Для работы распределенной модели FESudioFSI клиент приложения должен содержать информацию о прокси двух серверов OpenFOAM и FESudio, каждый прокси обладает уникальным именем, которое служит идентификатором при обращении к соответствующему серверу. Удаленный вызов методов, реализованных на стороне сервера, осуществляется посредством созданных прокси. Кроме того, клиент хранит и обновляет вектора перемещений и давления, для их последующего использования в расчете, и управляет логикой работы всей программы.

Со стороны серверов поведение Ice-объектов реализуется с помощью сервантов. Фактически сервант — это экземпляр реализованного класса. Базовый код серванта генерируется Slice-компилятором, от разработчика требуется реализовать методы класса, которые соответствуют операциям из интерфейса Ice-объекта. При поступлении вызова среда выполнения Ice на серверной стороне находит соответствующий вызываемому объекту сервант и делегирует ему обработку вызова. Со стороны серверов реализованы ряд методов, необходимых для распределенного решения задачи: вычисление перемещений (сервант FESudio), определение поля давления (сервант OpenFOAM), передача полученных решений клиенту, получение новых данных о перемещении и давлении, необходимых для следующего шага решения.

На рисунке 2 показана распределенная модель приложения FESudioFSI. Она представляет объединение трех Ice-объектов: серверов OpenFOAM и FESudio и клиента FESudioFSI.

Ice поддерживает модели синхронных и асинхронных вызовов. В последнем случае клиент, вызывая объект при помощи прокси, наряду с обычными параметрами операции, передает прокси специальный callback-объект. После вызова прокси управление сразу возвращается клиенту. При завершении вызова объекта среда выполнения Ice на клиентской стороне вызывает метод callback-объекта, передавая ему результаты вызова или исключения. С точки зрения сервера, обработка синхронных и асинхронных вызовов выглядит абсолютно одинаково. Клиент может и не знать, с помощью какой модели осуществляется обработка его вызова.

## Запуск приложений

В целом функционирование распределенного комплекса выглядит следующим образом. На CPU/GPU процессорах запускаются серверные объекты OpenFOAM и FESudio, в управляющем процессе создается клиентский объект FESudioFSI, который содержит прокси к соответствующим сервантам. Во время запуска серверных объектов инициализируются переменные, необходимые для расчетов каждой из программ OpenFOAM и FESudio, а также загружаются

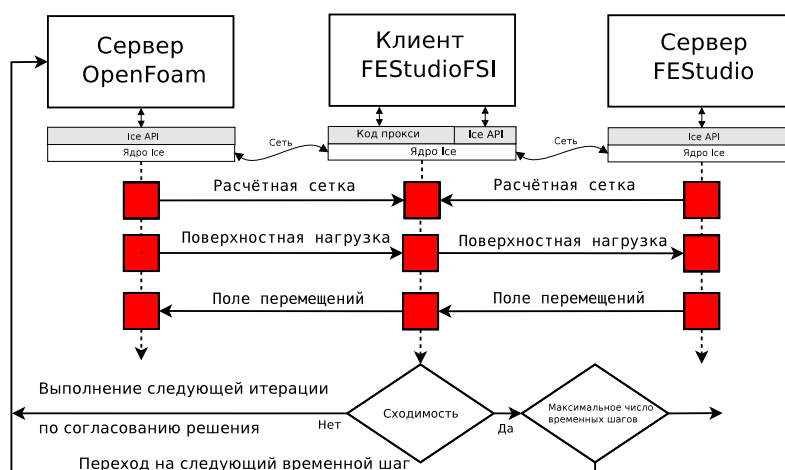


Рис. 2. Распределенная модель решения задачи FSI

расчетные сетки. Далее, после инициализации, как только сервера сигнализируют о готовности, клиент запускает расчет и обмен данными, выполняя соответствующие запросы серверам. Кроме обеспечения взаимодействия двух серверов клиент, в рамках построенной модели, занимается интерполяцией данных между сетками и деформированием сетки в задаче CFD. Время продолжительности расчета определяется клиентом и не зависит от серверов.

Описанный механизм связывания объектно-ориентированной модели и инфраструктуры Ice позволяет, путем замены объектов, получить распределенную объектно-ориентированную программу. Переход к распределенным/параллельным вычислениям осуществляется при минимальной модификации последовательного кода.

Ключевые возможности разрабатываемого ПО FESstudioFSI:

1. Выбор алгоритмов для слабого связывания приложений (последовательные алгоритмы, обмен данными происходит на каждом временном шаге один раз) и сильного связывания (в пределах одного шага по времени реализуется итерационная процедура совместного решения уравнений с последовательной передачей нагрузок и перемещений).
2. Поддержка связывания различных независимых приложений посредством интерфейса прикладного уровня.
3. Поддержка различных методов переноса данных для несогласованных сеток, функций интерполяции и перераспределения данных, деформации расчетных сеток.

## Параллельные алгоритмы, выполняемые на связывающем приложении

Для решения задач интерполяции и деформирования сетки использовался подход, связанный с использованием радиальных базисных функций, обладающий рядом преимуществ [De Boer et al., 2007]: использование движения отдельных точек, а не сетки (позволяет использовать только данные об узлах, не учитывая данные связанные с сеткой, что сокращает затраты на коммуникации); возможность рассмотрения систем меньшего размера при использовании компактных функций; высокая параллельная эффективность, связанная прежде всего с тем, что решение задачи сводится к решению системы уравнений, поиск которого можно достаточно эффективно распараллелить.

Основной идеей метода является поиск необходимого интерполянта в виде линейной комбинации, образованной радиальными базисными функциями. В этом случае интерполянт в виде линейной комбинации запишется следующим образом:

$$w_i(\mathbf{x}) = \sum_{j=1}^{n_s} \lambda_j \phi(\|\mathbf{x} - \mathbf{x}_{s_j}\|) + q(\mathbf{x}),$$

где  $w_i = \{\mathbf{u}_i, p\mathbf{n}_i\}$ ;  $i = \{1, \dots, d\}$ ,  $d$  — размерность задачи; коэффициенты  $\lambda_j$  и полином  $q$  находятся из условия:  $w_i(\mathbf{x}_{s_j}) = \mathbf{W}_{s_j}^i$  и дополнительного соотношения  $\sum_{j=1}^{n_s} \lambda_j p(\mathbf{x}_{s_j}) = 0$  для всех полиномов  $p$  степени меньше или равной степени полинома  $q$ .

Тогда система уравнений для неизвестных коэффициентов для задачи CSD в матричной форме запишется следующим образом:

$$\begin{bmatrix} \Phi_{ss} & \mathbf{B}_s \\ \mathbf{B}_s^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \lambda \\ \mu \end{bmatrix} = \begin{bmatrix} \mathbf{W}_s^i \\ \mathbf{0} \end{bmatrix}, \quad \text{где} \quad \Phi_{ss} = \phi(\|\mathbf{x}_{s_i} - \mathbf{x}_{s_j}\|), \quad \mathbf{B}_s = \begin{bmatrix} 1 & x_{s_j} & y_{s_j} & z_{s_j} \end{bmatrix}.$$

Для параметров на газодинамической сетке запишем:

$$\begin{bmatrix} \Phi_{fs} \mathbf{B}_f \end{bmatrix} \begin{bmatrix} \lambda \\ \mu \end{bmatrix} = \mathbf{W}_f^i.$$

Кроме того, значения на газодинамической сетке (CFD) могут быть получены и непосредственно из соотношения:

$$\mathbf{W}_f^i = \begin{bmatrix} \Phi_{fs} \mathbf{B}_f \end{bmatrix} \begin{bmatrix} \Phi_{ss} & \mathbf{B}_s \\ \mathbf{B}_s^T & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{W}_s^i \\ \mathbf{0} \end{bmatrix}.$$

Аналогично для задачи деформирования газодинамической сетки (CMD) построим интерполирующую функцию в виде линейной комбинации, образованной радиальными базисными функциями. Эта функция выглядит следующим образом:  $w_i(\mathbf{x}) = \sum_{j=1}^{N_\Gamma} \lambda_j \phi(\|\mathbf{x} - \mathbf{x}_{\Gamma_j}\|) + q(\mathbf{x})$ , здесь  $\mathbf{x}_\Gamma = (x_{\Gamma_j}, y_{\Gamma_j}, z_{\Gamma_j})$ ,  $N_\Gamma$  — число граничных узлов  $\Gamma_{fs}$ . Пусть  $\Delta \mathbf{x}_{\Gamma_j}$  приращения координат  $n$  границы, полученные из CSD, тогда коэффициенты  $\lambda_j$  и коэффициенты линейного полинома могут быть получены из решения системы:

$$\begin{pmatrix} \Phi & \mathbf{B}_\Gamma \\ \mathbf{B}_\Gamma^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \lambda \\ \mu \end{pmatrix} = \begin{pmatrix} \Delta \mathbf{x}_\Gamma \\ \mathbf{0} \end{pmatrix}.$$

Для внутренних узлов сетки  $\Omega_f$  приращения  $\Delta \mathbf{x}_{\Omega_f}$  находятся  $\Delta \mathbf{x}_j(\Omega_f) = w_i(\mathbf{x}_j(\Omega_f))$ .

Таким образом, рассматриваемые выше задачи сводятся к решению систем линейных алгебраических уравнений с матрицами разреженности и порядок которых зависит от выбора радиальных базисных функций. Такие системы можно эффективно решать предобусловленным методом сопряженных градиентов на GPU. В работе использовался решатель СЛАУ собственной разработки, который позволяет решать системы на нескольких GPU и на GPU-кластерах [Копысов и др., 2012а].

Качество сетки, получаемое при использовании радиальных базисных функций для деформирования сетки, исследовалась на трехмерной задаче взаимодействия потока газа с отрывающейся заслонкой. Сетка состояла из 74376 тетраэдров и 15154 узлов. Поворот заслонки осуществлялся вокруг оси  $O_x$  за один шаг и составил  $60^\circ$ . На рисунке 3 представлено исходное состояние сетки и сетка после поворота заслонки на  $60^\circ$ .

Качество получаемой сетки оценивалось по нескольким мерам качества [Stimpson et al., 2007]: соотношение максимальной и минимальной длин ребер тетраэдра («Edge Ratio»), минимальный двугранный угол тетраэдра («Minimal Dihedral Angle»), нормированный якобиан («Scaled Jacobian») и др.



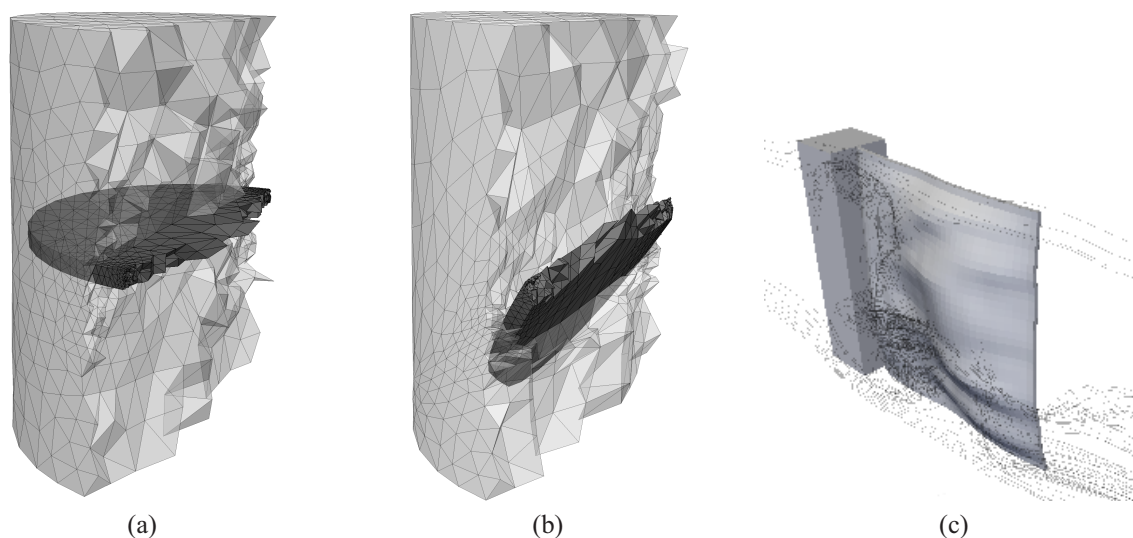


Рис. 3. Примеры сеточных областей. Задача поворота заслонки: (a) начальная сетка, (b) деформируемая сетка и (c) расчета обтекания несжимаемой вязкой жидкости гибкой пластины

В качестве радиальной базисной функции использовалась функция с компактным носителем  $\phi(\|\mathbf{x}\|) = (1 - \|\mathbf{x}\|/r)^2 \in C^0$ , где  $r$  — радиус влияния (в нашем случае  $r$  принималось равным 0.5). Среднее значение параметра «Edge Ratio» для исходной сетки составляло 34.41, после поворота — 34.22; «Minimal Dihedral Angle» —  $41.96^\circ$  и  $41.98^\circ$  соответственно; для параметра «Scaled Jacobian» — 0.49 и 0.25.

Результаты исследования показали, что деформирование сетки практически не изменяет меры качества сетки при высокой степени деформации. Несмотря на большой порядок обусловленности получаемых матриц, использование диагонального предобуславливателя для метода сопряженных градиентов обеспечило сходимость решения. В зависимости от вида RBF-функций число обусловленности матрицы изменялось от  $10^4$  для  $\phi(\|\mathbf{x}\|) = e^{-\|\mathbf{x}\|}$  до  $10^7$  для  $\phi(\|\mathbf{x}\|) = (1 - \|\mathbf{x}\|/r)^2$ . Результаты на рисунке 3 получены на вычислительном кластере с несколькими GPU и использованием технологий OpenMP и CUDA. Время расчета примера одним потоком CPU составило 23.5 минуты, время расчета на GPU — 13 секунд. Таким образом, использование GPU для задачи деформирования расчетной области позволило добиться ускорения решения SLAU в 108 раз. Важно отметить, что процедура деформирования сетки осуществляется на каждом временном интервале и ее эффективная параллельная реализация позволяет существенно сократить суммарные вычислительные затраты при решении задач FSI.

## Тестовые результаты

Для тестирования алгоритма сильного связывания была рассмотрена в трехмерной постановке задача обтекания потоком несжимаемой вязкой жидкости гибкой пластины, закрепленной на неподвижном основании [Vaassen et al., 2010]. Гидродинамическая картина течения (рис. 3с) определяется движением вихрей, образующихся при обтекании жесткого бруса. Вихри зарождаются попеременно на противоположных гранях основания, что приводит к возникновению колебаний давления и вынуждающей силы, отклоняющей пластину.

Использование модели несжимаемой жидкости делает невозможным применение алгоритма слабого связывания, поэтому при решении применялась итерационная процедура согласо-

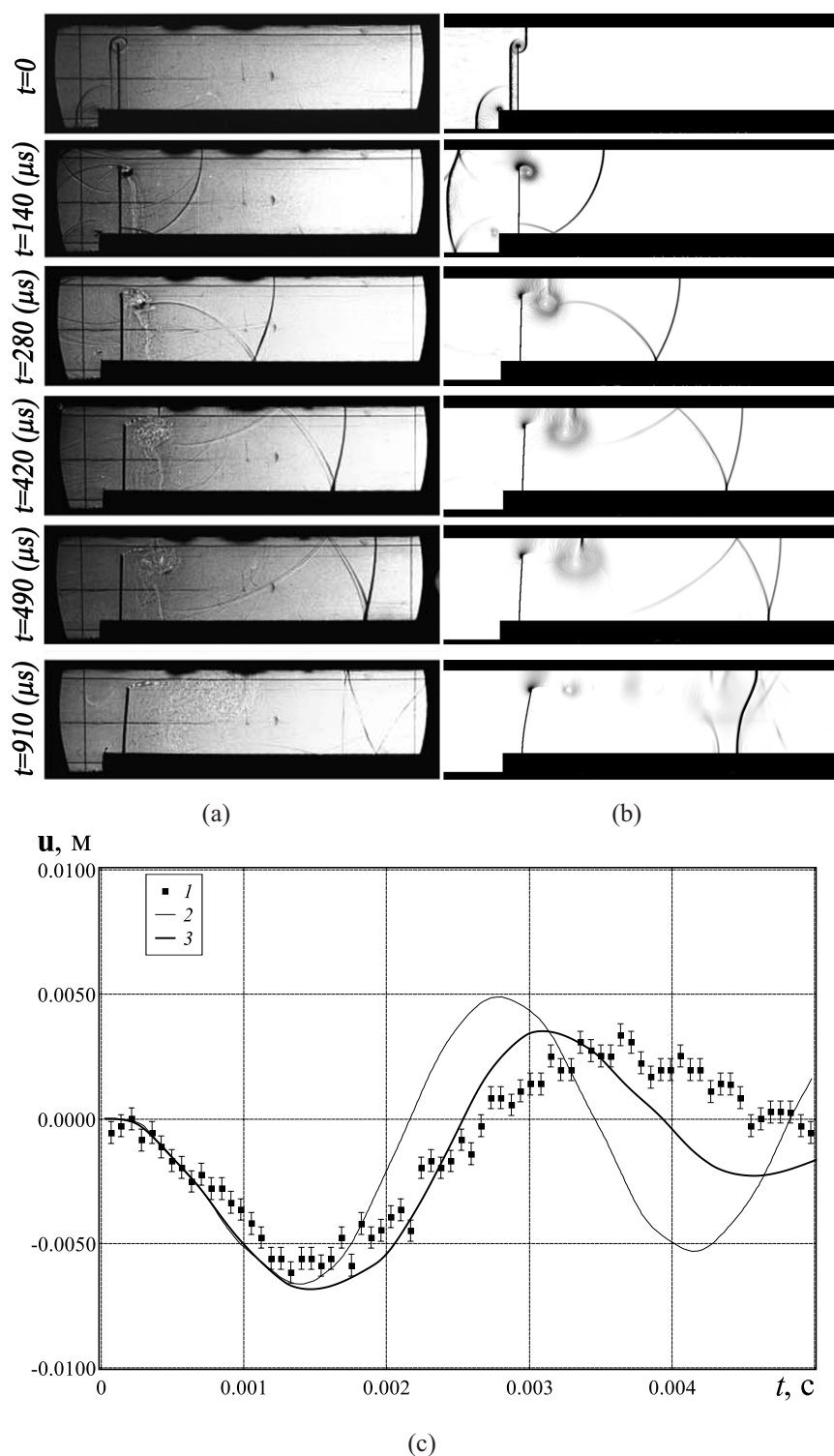


Рис. 4. Картина течения в различные моменты времени ( $h = 50 \text{ мм}$ ): (а) эксперимент [Giordano et al., 2005]; (б) расчет с помощью FESstudioFSI; (с) график зависимостей перемещения края панели: 1 — эксперимент [Giordano et al., 2005]; 2 — расчет [Giordano et al., 2005]; 3 — расчет с помощью FESstudioFSI

ния решений подзадач. Для удовлетворения условия сопряжения на границе раздела требовалось от 5 до 7 итераций внутреннего цикла связывания.

Вторая из рассмотренных задач описывает взаимодействие нестационарного сверхзвукового потока воздуха и деформируемой консоли в ударной трубе. Подробное описание задачи, а также результаты экспериментальных и численных исследований можно найти в [Giordano et al., 2005], [Копысов и др., 2012b].

Физические особенности моделируемого явления допускают применение последовательного алгоритма слабого связывания для отыскания решения. Полученное численное решение этой задачи хорошо согласуется с данными экспериментальных исследований [Giordano et al., 2005]. Сравнение выполнялось как для газодинамической картины падающих и отраженных ударных волн (рис. 4а, б), так и для смещений пластины (результаты и их обсуждение приведены в [Копысов и др., 2012b]).

Предлагаемый подход к моделированию задач взаимодействия потока газа и деформируемой конструкции/тела основан на выделении этапа согласования и взаимодействий в отдельную программную подсистему, не зависящую от используемых реализаций расчетов газодинамических полей и напряженно-деформированного состояния. Создан прикладной программный интерфейс FEStudioFSI с выделением самостоятельного (параллельного) приложения, непосредственно решающего всю совокупность задач связывания (синхронизация, обмен данными, интерполяция, деформирование расчетных сеток). Таким образом можно считать, что междисциплинарный программный комплекс построен по архитектуре клиент-сервер, где каждое приложение-клиент оптимальным образом распараллелено, решает свою узкоспециальную прикладную задачу и обменивается расчетными данными через сервер с другими клиентами. Новый подход обеспечивает наибольшую универсальность и позволяет объединять независимые приложения, от которых требуется лишь использование соответствующего прикладного программного интерфейса.

## Список литературы

- Копысов С.П., Кузьмин И.М., Недождогин Н.С., Новиков А.К. Параллельные алгоритмы формирования и решения системы дополнения Шура на графических ускорителях // Ученые записки Казанского университета. Серия Физико-математические науки. — 2012а. — Т. 154, № 3. — С. 202–215.
- Копысов С.П., Кузьмин И.М., Тонков Л.Е. Моделирование взаимодействия сверхзвукового потока и деформируемой панели в ударной трубе // Вестник Удмуртского университета. Математика. Механика. Компьютерные науки. — 2012b. — № 2. — С. 156–165.
- De Boer A., Van der Schoot M., Bijl H. Mesh deformation based on radial basis function interpolation // Computer and Structures. — 2007. — Vol. 85. — P. 784–795.
- Farhat C., Lesoinne M. Two efficient staggered algorithms for the serial and parallel solution of three-dimensional nonlinear transient aeroelastic problems // Computer Methods in Applied Mechanics and Engineering. — 2000. — Vol. 182. — P. 499–515.
- Giordano J., Jourdan G., Burtschell Y. et al. Shock wave impacts on deforming panel, an application of fluid-structure interaction // Shock Waves. — 2005. — Vol. 14. — P. 103–110.
- Kopysov S. P., Krasnopyorov I. V., Novikov A. K., Rychkov V. N. Parallel Distributed Object-Oriented Framework for Domain Decomposition // Domain Decomposition Methods in Science and Engineering / Barth T. J., Griebel M., Keyes D. E., etc. (ed.). — Vol. 40 of LNCSE. — Springer, 2004. — P. 605–614.
- Kopysov S. P., Novikov A. K. Parallel adaptive mesh refinement with load balancing on heterogeneous cluster // Algorithms and Tools for Parallel Computing on Heterogeneous Clusters / Desprez F., Fleury E., Kalinov A., Lastovetsky A. (ed.). — Nova Science Publishers, 2006. — P. 23–33.

*Stimpson C., Ernst C., Knupp P. et al.* The Verdict Geometric Quality Library: Tech. Rep. Sandia Report SAND2007-175 / Stimpson C., Ernst C., Knupp P. et al.: Sandia National Laboratories, 2007.

*Vaassen J.-M., De Vincenzo P., Hirsch C., Leonard B.* Strong coupling algorithm to solve fluid-structure interaction problems with a staggered approach // 11th Int. WS on Simulation and EGSE facilities for Space Programmes. — Noordwijk, Netherlands: 2010. — 28–30 September. — P. 8.