

УДК 519.688

**ПАРАЛЛЕЛЬНЫЙ ПРОГРАММНЫЙ КОМПЛЕКС NOISETTE
ДЛЯ КРУПНОМАСШТАБНЫХ РАСЧЕТОВ ЗАДАЧ
АЭРОДИНАМИКИ И АЭРОАКУСТИКИ**

И. В. Абалакин¹, П. А. Бахвалов², А. В. Горобец¹, А. П. Дубень¹, Т. К. Козубская¹

Представлен программный комплекс NOISETte, основанный на схемах повышенной точности с определением переменных в узлах неструктурированных сеток, который позволяет моделировать задачи газовой динамики и аэроакустики с использованием десятков тысяч процессорных ядер суперкомпьютера. Приводится обзор лежащих в основе численных методов и моделей, включающий в себя пространственную дискретизацию, интегрирование по времени, модели турбулентности и модели дальнего поля. Подробно описаны особенности программной реализации. Большое внимание уделяется распараллеливанию в рамках двухуровневой модели MPI+OpenMP.

Ключевые слова: газовая динамика, аэроакустика, аэродинамика, параллельные вычисления, MPI, OpenMP.

1. Введение. Задачи вычислительной газовой динамики приобретают в настоящее время все большую важность и актуальность во многих отраслях промышленности, в особенности там, где присутствуют высокоскоростные турбулентные течения — в авиационной промышленности, турбомашиностроении и авиационном двигателестроении, ветроэнергетике, автомобилестроении и т.д. В частности, в связи с постоянно ужесточающимися требованиями по шуму летательных аппаратов на местности, все больше возрастает интерес к разработке моделей и методов, направленных на изучение механизмов генерации шума в турбулентных течениях. В этой области типы задач, решаемых с помощью математического моделирования, можно условно разделить на три категории, соответствующие основным источникам шума самолетов:

- 1) генерация шума высокоскоростной струей — шум от реактивной струи авиадвигателя;
- 2) шум при внешнем обтекании тел сложной геометрии, в частности шум турбулентного следа и шум от взаимодействия турбулентных структур с твердым телом — аэродинамический шум при обтекании планера, механизации крыла, шасси;
- 3) моделирование акустических резонаторов — оптимизация звукопоглощающих конструкций, специальных панелей, расположенных в различных частях мотогондолы и двигателя для снижения шума от вентилятора турбореактивного двигателя.

Этот список можно дополнить также специфическими задачами по моделированию лопаток вентилятора, в частности, с целью снижения интенсивности ударных волн, возникающих на кромках лопаток, движущихся со сверхзвуковой скоростью. Более подробно с состоянием дел в области вычислительной аэроакустики можно ознакомиться, например, в [1].

Моделирование задач аэроакустики, как правило, представляет собой сложный газодинамический расчет, дополненный вычислениями, обеспечивающими получение акустических возмущений в дальнем поле течения на основе интегральных методик. Данные задачи имеют несколько особенностей, обуславливающих высокую вычислительную стоимость. Во-первых, это необходимость использования численных схем повышенного порядка точности, чтобы корректно воспроизводить и турбулентное течение, и акустические волны, имеющие очень небольшую амплитуду по сравнению с турбулентными пульсациями. Во-вторых, необходимость высокого разрешения по пространству, особенно в области ближнего поля, где происходят процессы генерации шума. В-третьих, необходимость высокого разрешения по времени для получения широкополосных спектров в сочетании с необходимостью интегрирования по длительному временному интервалу для получения качественного осреднения и, в частности, корректного спектрального анализа.

¹ Институт прикладной математики им. М. В. Келдыша РАН (ИПМ РАН), Миусская пл., 4А, 125047, Москва; И. В. Абалакин, ст. науч. сотр., e-mail: ilya.abalakin@gmail.com; А. В. Горобец, ст. науч. сотр., e-mail: chererock@gmail.com; А. П. Дубень, мл. науч. сотр., аспирант, e-mail: dualks@gmail.com; Т. К. Козубская, зав. сектором, e-mail: tatiana.kozubskaya@gmail.com

² Московский физико-технический институт (МФТИ), Институтский переулок, 9, 141700, Московская обл., г. Долгопрудный, аспирант, e-mail: bahvalo@mail.ru

Некоммерческий комплекс программ NOISEtte, описываемый в настоящей статье, разработан в секторе вычислительной аэроакустики Института прикладной математики им. М. В. Келдыша РАН и предназначен в первую очередь для крупномасштабных суперкомпьютерных расчетов широкого спектра газодинамических и аэроакустических задач. Программный комплекс включает в себя численные методики решения систем уравнений Эйлера и Навье–Стокса для сжимаемого газа, а также численную реализацию моделей, построенных на основе системы уравнений Эйлера и используемых в вычислительной аэроакустике [2]. Для расчета турбулентных течений в комплексе программ предусмотрены современные методы моделирования турбулентности — RANS (Reynolds-Averaged Navier–Stokes), LES (Large Eddy Simulation), DES (Detached Eddy Simulation), DDES (Delayed Detached Eddy Simulation) и IDDES (Improved Delayed Detached Eddy Simulation).

Основу вычислительного алгоритма составляют схемы повышенной точности с определением переменных в узлах для неструктурированных тетраэдральных сеток, что позволяет моделировать обтекание геометрически сложных объектов. Для пространственной дискретизации используется многопараметрическая конечно-объемная схема повышенного порядка точности [3] (до шестого включительно в зависимости от качества сетки). Для интегрирования по времени используются различные явные и неявные схемы, в том числе метод Рунге–Кутты четвертого порядка и неявная схема второго порядка на основе линеаризации по Ньютону.

Двухуровневое распараллеливание MPI+OpenMP и инфраструктура для работы с большими неструктурированными сетками дает возможность задействовать для одного расчета несколько десятков тысяч процессорных ядер и использовать сетки с числом элементов более миллиарда. Необходимость применения сложной параллельной модели обусловлена особенностями современной архитектуры суперкомпьютеров с многоядерными вычислительными модулями.

В коммерческих пакетах, широко используемых в промышленности, реализованы уже существующие и хорошо отработанные решения. В этом состоит принципиальное отличие от исследовательского кода, в котором постоянно реализуются и отрабатываются новые методы. С обзором по наиболее известным европейским исследовательским кодам в области газовой динамики читатель может ознакомиться, например, в [4]. Одной из основных отличительных особенностей комплекса NOISEtte, в существенной степени повлиявшей на выбор подходов к программной реализации, является его двойное назначение: код ориентирован одновременно как на исследовательскую работу, решение фундаментальных задач, отработку новых моделей, численных методов и вычислительных технологий, так и на решение прикладных промышленных задач, что подразумевает удобство использования, высокую производительности и надежность.

Статья структурирована следующим образом. В разделе 2 приводится краткое описание математической модели и численной реализации. В разделе 3 представлена структура и состав программного комплекса NOISEtte и приводятся подробности его программной реализации. Раздел 4 посвящен параллельным технологиям и производительности вычислений. В разделе 5 представлены некоторые примеры приложений комплекса NOISEtte для решения исследовательских задач.

2. Математическая модель и численная реализация.

2.1. Математическая модель. В программном комплексе NOISEtte за основу взята модель течения сжимаемого газа, описываемая полными уравнениями Навье–Стокса. Как частные случаи этой модели, в NOISEtte реализовано семейство моделей, построенных на основе уравнений Эйлера и линеаризованных уравнений Эйлера [2]. Базовая система уравнений Навье–Стокса дополняется необходимыми источниками членами и уравнениями для моделирования турбулентных течений в рамках подходов RANS (Reynolds Averaged Navier–Stokes — осредненные по Рейнольдсу уравнения Навье–Стокса), LES (Large Eddy Simulation — моделирование крупных вихрей), а также гибридных RAND-LES подходов семейства DES (Detached Eddy Simulation — моделирование отсоединенных вихрей).

Для расчета стационарных турбулентных течений с большими числами Рейнольдса в качестве замыкания RANS-уравнений была выбрана модель турбулентности Спаларта–Аллараса (SA) [5]. Выбор модели SA обусловлен следующими причинами: относительная простота модели, включающей в себя одно дополнительное дифференциальное уравнение, физически разумное моделирование разного типа течений (пограничные слои, струи, отрывные течения и др.) и удобство использования в гибридной модели RANS/LES.

При моделировании с использованием RANS во многих случаях невозможно достичь необходимой точности в случае нестационарных течений. Наиболее подходящими для расчета высокорейнольдсовых турбулентных течений являются гибридные методы, в частности методы семейства DES [6]. Преимущество данного подхода заключается в сочетании сильных сторон этих двух методов — высокой точности полуэмпирической модели RANS в областях присоединенного пограничного слоя и достаточной универ-

сальности модели LES в отрывных областях потока [7].

Отметим, что исходная версия DES некорректно работает при расчете толстых пограничных слоев и узких отрывных зон. В тех областях, где RANS переходит в LES внутри присоединенного пограничного слоя, из-за низкого сеточного разрешения может возникать нефизический отрыв пограничного слоя от стенки. Чтобы избежать этих негативных явлений, в NOISEtte была реализована модифицированная версия DES с задержкой перехода в пограничном слое (Delayed DES — DDES), предложенная в работе [8], и метод IDDES (Improved DDES) [9] с улучшенным пристеночным LES-моделированием.

Для моделирования распространения акустических возмущений в дальнем поле в NOISEtte применяется метод Фокса–Уилльямса–Хокинга (FW/H). Из расчетной сетки вырезается поверхность, окружающая течение в источниковой зоне (ближнем поле). По пространственно-временному распределению газодинамических переменных на ней рассчитывается отклик в дальнем поле как поверхностный интеграл с запаздыванием. С параллельной реализацией метода можно ознакомиться в [10].

2.2. Численная реализация. В NOISEtte используется следующая численная реализация уравнений Навье–Стокса: пространственная дискретизация системы уравнений Навье–Стокса проводится на тетраэдральной сетке с использованием смешанного метода аппроксимации, интегрирование по времени осуществляется явными методами типа Рунге–Кутты и неявными методами с линеаризацией по Ньютону разностной системы уравнений.

2.2.1. Пространственная аппроксимация. Формулировка метода конечных объемов требует разбиения расчетной области на контрольные объемы, которые можно определить различными способами в зависимости от исходной тетраэдральной сетки и желаемой аппроксимации конвективного оператора. Контрольный объем строится вокруг узла сетки с использованием центров ребер, центров граней тетраэдров, опирающихся на этот узел, и центров самих тетраэдров. В NOISEtte определяются два типа контрольных объемов — барицентрический (медианный) и ортоцентрический [11], отличающихся определением центра тетраэдра и центра его граней.

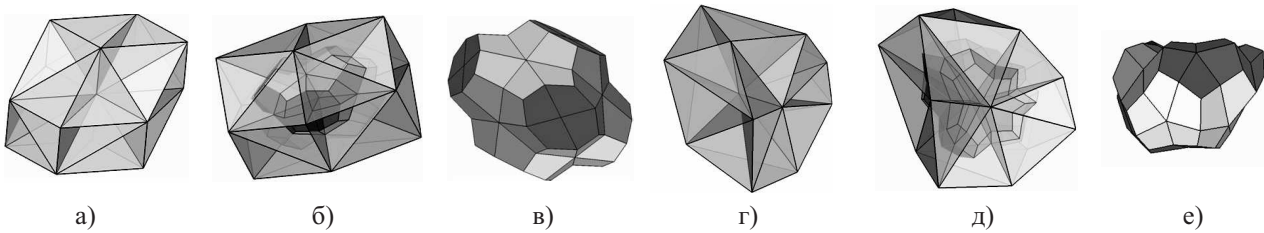


Рис. 1. Барицентрические контрольные объемы: а)–в) для сетки, полученной разбиением декартовой сетки на тетраэдры; г)–е) для произвольной неструктурированной сетки; а), г) тетраэдры, содержащие центр контрольного объема; б), д) объем внутри тетраэдров; в), е) объем в отдельности

Для построения барицентрического объема выбираются центры масс тетраэдра, центры тяжести граней и середины ребер тетраэдра. Пример барицентрического контрольного объема показан на рис. 1. При построении ортоцентрического контрольного объема под центром тетраэдра понимается центр описанной вокруг него сферы, а в качестве центра грани выбирается центр описанной вокруг треугольника окружности — точка пересечения срединных перпендикуляров. Если центры описанной окружности или сферы оказываются вне симплекса (тетраэдра или треугольника), то в качестве центра такого симплекса выбирается середина его наибольшей грани.

Для получения разностного аналога системы уравнений Навье–Стокса воспользуемся ее интегральным представлением. Применим метод конечных объемов к конвективной части системы и метод конечных элементов с использованием линейных финитных базисных функций (P1-метод Галеркина) [12] к ее диффузионной части (вторые производные).

Для диффузионной части системы полученный объемный интеграл преобразуется с учетом финитности базисной функции в систему линейных соотношений (согласно [12]), связывающих значения функции в узлах тетраэдра и координаты узлов тетраэдра (матрица жесткости).

Объемный интеграл от конвективных производных преобразуется в сумму поверхностных интегралов по границам (сегментам) ∂C_{ij} расчетной ячейки, разделяющим соседние точки i и j , от газодинамических потоков, спроецированных на нормаль \mathbf{n}_{ij} к этой границе. Отметим, что поверхность ∂C_{ij} в общем случае не всегда лежит в одной плоскости, а представляет собой объединение поверхностей, лежащих в разных плоскостях. В этом случае в качестве нормали к сегменту выбирается среднее значение $\mathbf{n}_{ij} = \iint_{\partial C_{ij}} \mathbf{n} d\sigma$.

Нормальный газодинамический поток на сегменте заменяется на численный поток. Для определения численного потока используется схема Роу [13] в предположении постоянства газодинамических переменных на соседних ячейках или противопоточная схема Хуанга [14] в предположении постоянства газодинамических потоков. В указанных предположениях численный поток через грань ячейки аппроксимируется с первым порядком точности. Для повышения порядка точности численного потока производится замена кусочно-постоянных газодинамических или потоковых переменных на кусочно-полиномиальное распределение. За основу берется многопараметрическая семиточечная одномерная схема, имеющая 3–6 порядки [15].

Обобщение этой схемы на многомерный случай производится следующим образом. Через грань расчетной ячейки проводится прямая ij , соединяющая соседние точки, разделенные этой гранью. На этой прямой определяются многомерные аналоги первых и вторых конечных разностей. Для этого вводятся следующие три типа градиентов:

- а) центральный градиент — разность между соседними точками i и j ;
- б) градиенты по противопотоковым тетраэдрам (тетраэдрам, имеющим в качестве вершин выбранные соседние точки, либо i , либо j , и имеющим пересечение с прямой ij); градиент по тетраэдру вычисляется по линейной функции аналогично методу конечных элементов;
- в) узловые градиенты, вычисляемые как среднее по ячейке от градиентов по тетраэдрам, содержащим узел i или j .

Пример шаблона трехмерной схемы показан на рис. 2. Если рассмотреть случай сетки, полученной из декартовой сетки разбиением параллелепипедов на тетраэдры, то можно подобрать такую линейную комбинацию градиентов, что результирующая схема будет обладать 3–6 порядками аппроксимации аналогично одномерному случаю [3]. Однако в общем случае произвольной неструктурированной тетраэдральной сетки теоретический порядок схемы будет не выше второго.

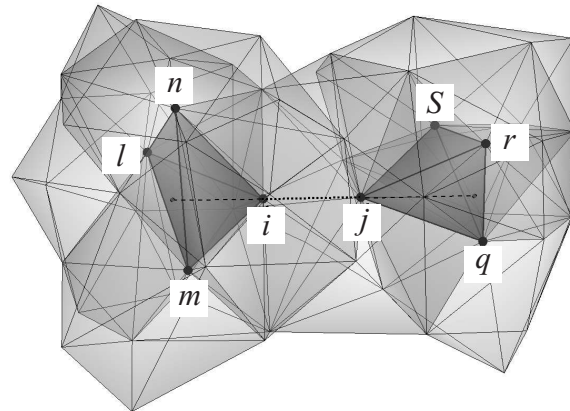


Рис. 2. Шаблон схемы для расчета конвективной части потоков через грань контрольного объема между узлами i и j . Базовая часть шаблона содержит узлы левого и правого противопотоковых тетраэдров, а для расчета узловых градиентов используются все соседние с ними узлы

2.2.2. Аппроксимация по времени. При интегральном представлении системы уравнений Навье–Стокса предположим, что значение производной по времени постоянно на расчетной ячейке. Тогда, зная пространственную аппроксимацию, имеем полудискретную систему уравнений Навье–Стокса, которая представляет собой систему обыкновенных дифференциальных уравнений. Для решения этой системы используется явная классическая схема Рунге–Кутты четвертого порядка точности. Кроме того, реализовано неявное интегрирование по времени с использованием трехслойной схемы второго порядка точности с поточечной линеаризацией по Ньютону пространственной разностной системы уравнений. Получающаяся в результате линеаризации линейная система уравнений решается стабилизированным методом бисопряженных градиентов (BiCGSTAB) [16].

2.2.3. Постановка граничных условий. В NOISEtte реализованы разные типы граничных условий (ГУ), определяемые потоками на граничных поверхностях расчетной области:

- 1) условие прилипания (равенство нулю скоростей на границе) на твердой стенке: задание температуры стенки (изотермическая поверхность), задание на стенке потока тепла и, как частный случай, адиабатическая поверхность (равенство нулю теплового потока);
- 2) условие симметрии (или непротекания — твердая стенка при отсутствии вязкости), при котором нормальная к поверхности компонента скорости равна нулю, что приводит к нулевым потокам массы и полной энергии и ненулевым потокам импульса;
- 3) условия Дирихле — задание полного набора газодинамических параметров на границе; применение ГУ такого типа ограничено случаем сверхзвуковых входных условий;
- 4) ГУ на дозвуковых входных и выходных границах определяются потоками, расщепленными по знаку характеристических скоростей на границе поверхности [17]; сами потоки вычисляются с использованием значений газодинамических параметров на границе и значений вне расчетной области, которые определяются либо сносом по характеристикам изнутри расчетной области, либо задаются;
- 5) неотражающие ГУ Тама [18], основанные на асимптотическом решении линейных уравнений Эй-

лера.

3. Программная реализация. Исходный код NOISEtte написан на языке C++, выбор подходов к программной реализации был обусловлен следующими требованиями.

1. Переносимость — совместимость с различными операционными системами (ОС), в том числе Windows XP/Vista/7 и Linux.

2. Совместимость с различными типами вычислительных систем от рабочих станций до крупных суперкомпьютеров и с разными архитектурами процессоров, в частности Intel, AMD и IBM.

3. Совместимость с различными компиляторами C++, в том числе Microsoft VS C++, Intel C++, GNU C++ и IBM XL C++.

4. Автономность — базовая конфигурация должна работать без подключения каких-либо дополнительных библиотек, отличных от стандартных библиотек C++.

Этот список дополняют такие естественные требования, как удобство работы с исходным кодом, надежность и защищенность кода от внесения ошибок, высокая производительность вычислений. Переносимость и совместимость подразумевают, что базовая конфигурация не использует никаких программных решений, зависящих от типа ОС и конкретной архитектуры ВС. Однако это не исключает расширений базовой конфигурации, в которых используется средства конкретных ОС, настройка под конкретный тип процессора и использование возможностей конкретного компилятора. Аналогичным образом, автономность не исключает дополнения базовой конфигурации функционалом внешних библиотек.

Параллельная конфигурация использует MPI (версия стандарта 1.2) для модели с распределенной памятью и OpenMP (стандарт 3.0) для модели с общей памятью. Работа с кодом ведется в интегрированной среде разработки (IDE) Microsoft Visual Studio 8.0 (MVS) и выше. Отладчик и статический анализатор — входящие в состав MVS. Для профилирования и поиска ошибок по доступу к памяти используется Intel Parallel Studio. Для управления версиями используется SubVersion (SVN).

Комплекс NOISEtte состоит из вычислительного ядра и обширной инфраструктуры, включающей в себя средства для постановки задачи, подготовки сеточных данных и обработки результатов. Вычислительное ядро реализовано в виде библиотеки статической компоновки, от которой питаются исполнимые файлы расчетного кода и инфраструктуры. Набор программных средств комплекса NOISEtte может быть условно разделен на три категории: препроцессор — программы, выполняющиеся перед началом расчета; инфраструктура вычислительного ядра — внутренние вспомогательные средства; постпроцессор — программы обработки результатов расчета.

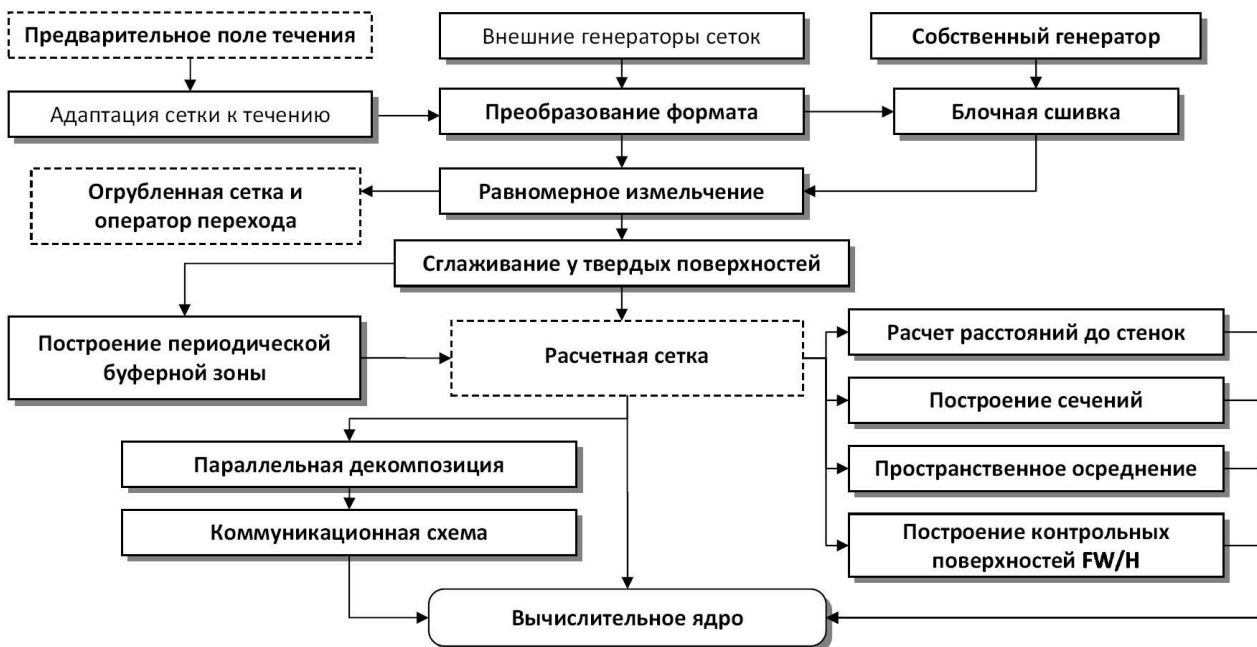


Рис. 3. Схема работы средств обработки сеточных данных

3.1. Инфраструктура препроцессора. Основу препроцессора составляют параллельные средства для работы с неструктурированными сетками большого объема (с числом элементов более миллиарда). Общая схема средств препроцессора показана на рис. 3.

3.1.1. Построение сетки. Для генерации сеток используются либо коммерческие генераторы, либо собственный сеточный генератор. Ограничения последовательных генераторов не позволяют получать сетки с большим числом элементов (порядка 10^8 – 10^9). Поэтому приходится либо сшивать сетку из множества блоков, либо равномерно измельчать, либо и то и другое. Необходимая функциональность реализована в следующих программах:

- преобразователь сеток из внешних форматов коммерческих генераторов и кодов (Gambit, Icem и CFX);
- генератор двумерных сеток путем сшивки блоков, топологически эквивалентных декартовой решетке, и разбиением на треугольники;
- построитель трехмерных тетраэдральных сеток из двумерных треугольных вытягиванием с постоянным шагом по третьей оси и разбиением получающихся призматических элементов на тетраэдры;
- сшивка сетки из блоков с совпадающими границами;
- параллельное средство для равномерного измельчения сетки; результирующая сетка имеет примерно в два раза большее пространственное разрешение в каждом из трех направлений;
- сглаживание сетки у твердых поверхностей; операция равномерного измельчения не улучшает качество представления геометрии расчетной области, поскольку новые узлы добавляются на существующие ребра тетраэдров; для повышения качества геометрии новые узлы, принадлежащие поверхностям расчетной области, перемещаются в соответствии с параметризацией геометрии;
- построение буферной зоны для периодических граничных условий методом топологического замыкания краев сетки фиктивными тетраэдрами;
- параллельная программа расчета расстояний до твердых поверхностей; для работы модели турбулентности в каждом узле сетки вычисляется расстояние до ближайшей твердой поверхности.

3.1.2. Построение подсеток и пространственное осреднение. Для обработки результатов и визуализации при расчете на больших сетках реализованы программные средства:

- построение последовательности сгущающихся сеток и операторов перехода между ними;
- вырезка параметрически заданной поверхности из трехмерной расчетной области (для расчета дальнего поля или визуализации); на выходе — двумерная сетка и оператор перехода с расчетной сетки для быстрой выгрузки данных;
- проецирование трехмерной сетки на двумерное сечение с целью пространственного осреднения по однородной оси (например, периодической) для повышения качества осредненных полей течения;
- построение топологии соответствия частей симметричной расчетной области с целью пространственного осреднения с учетом симметрии.

3.1.3. Декомпозиция расчетной области. Для параллельных вычислений неструктурированная сетка должна быть разбита на подобласти MPI-процессов. К разбиению предъявляются следующие очевидные требования: минимизация площади границ подобластей (для уменьшения объема обмена данными), минимизация максимального числа соседних подобластей (для уменьшения количества пересылок) и балансировка загрузки. После разбиения должна быть построена коммуникационная схема для конкретного шаблона численной аппроксимации, т.е. для каждого узла сетки необходимо определить, какому процессу, помимо процесса-владельца, он требуется для вычислений по своим узлам. Эту задачу решают следующие программы: программа декомпозиции на основе библиотек Metis и ParMetis [19]; параллельная программа декомпозиции GridSpider [20], разработанная в ИПМ РАН; параллельная программа построения коммуникационной схемы для заданной ширины шаблона численной аппроксимации.

3.2. Структура вычислительного ядра. Исходный код библиотеки вычислительного ядра условно разделен на следующие функциональные модули:

- модуль граничных условий: содержит обработчик поверхностных граней расчетной области и набор граничных условий;
- модуль точных решений: содержит обработчик точных решений в узлах сетки и набор точных решений для различных модельных задач;
- геометрический модуль: построение геометрии контрольных объемов;
- модуль формирования начальных данных и среднего поля;
- модуль неявного интегрирования по времени: содержит алгоритм неявного интегрирования и необходимую инфраструктуру — функции линейной алгебры для блочных разреженных неструктурированных матриц и решатель линейных систем на основе метода бисопряженных градиентов с локальным предобусловителем;
- модуль ввода-вывода: интерпретатор пользовательского ввода, функции чтения и записи полей в бинарном и текстовом формате, функции записи динамических данных;

- модуль численной схемы (пространственная аппроксимация): расчет шага по времени, расчет потоков через грани контрольных объемов, расчет узловых и тетраэдральных градиентов, реконструкция потоков и др.;
- модуль вязкости: вычисление диссипативных потоков в уравнении Навье–Стокса;
- параллельный модуль: содержит дополнительные структуры данных и инфраструктуру для параллельных вычислений под MPI и OpenMP;
- модуль источников: задает источниковый член в правой части уравнения для различных типов задач;
- турбулентный модуль: реализует набор моделей турбулентности RANS, LES, DES, DDES и IDDES;
- вспомогательные подпрограммы — таймер, солвер линейных систем малой размерности, алгоритмы сортировки и т.д.

3.2.1. Представление данных расчетной области. Расчетная область на неструктурированной тетраэдральной сетке с заданием значений сеточных функций в узлах определяется в NOISEtte списком координат узлов сетки, списком тетраэдров (топологией сетки) и списком треугольников (внешних граней сетки). Топология задается перечислением индексов узлов (вершин), составляющих каждый сеточный элемент — треугольник или тетраэдр. Вокруг узлов сетки выстраиваются контрольные объемы, каждому ребру сетки соответствует грань (или группа граней) контрольного объема. Таким образом, ребра и узлы сетки составляют граф связности контрольных объемов.

В состав расчетной области входят следующие элементы: узел сетки и его контрольный объем, тетраэдр, грань контрольного объема, внешняя грань — треугольник. Данные представляются в виде блочных векторов размерности от N по M элементов в блоке, где N — число элементов и M — число величин, заданных в элементе. Например, поля физических переменных представляются в виде блочного вектора с размером блока, равным 5: плотность, три компоненты скорости и давление.

Блочные векторы естественно представить в виде двумерных массивов размерности $N \times M$. Для этих целей реализован шаблонный класс — простейшая “обертка” над одномерным вектором данных: оператор `[]` возвращает указатель на заданный блок: `inline ValueType *operator [] (int i){return V+i*M;}`. Здесь `ValueType` — тип данных, `i` — номер блока, `V` — указатель на начало вектора данных. Такой способ отличается низкими накладными расходами на позиционирование и высокой скоростью доступа к данным. Кроме того, для класса реализованы операторы присваивания и арифметические операции, что позволяет обращаться с блочными векторами как со скалярными переменными по аналогии с возможностями языка Fortran.

Характерно, что для реализации элементов расчетной области не используются средства объектно-ориентированного программирования C++. Все данные расчетной области размещаются в блочных векторах, чтобы избежать потерь в производительности и затратах памяти, при этом оригинальная реализация доступа к данным с использованием макроподстановки позволяет имитировать доступ к свойствам класса. Рассмотрим пример сложной структуры данных — грань контрольного объема, обладающую следующими свойствами: номера “левого” и “правого” узлов, которые разделяет грань, номера “левого” и “правого” противопотоковых тетраэдров (только для схемы повышенного порядка), нормированные компоненты вектора нормали и площадь, шесть коэффициентов разложения вектора реконструкции по естественному базису противопотоковых тетраэдров (только для схемы повышенного порядка), значения потоков через грань (только для OpenMP) — пять потоков без дифференциальной модели турбулентности и шесть (и более) с этой моделью.

Представить грань в виде структуры или класса C++ было бы неэффективно из-за того, что в зависимости от типа численной схемы или параллельного режима существенно меняется состав свойств. Большое число свойств оказались бы не задействованными, что увеличило бы затраты памяти, потери в кэш и снизило бы скорость доступа. Данные размещаются в блочном векторе, при этом размерность блока определяется по фактическим параметрам расчета. Над блочным вектором реализована некоторая надстройка для позиционирования внутри блока. В общем виде доступ к некоторому свойству `Value` осуществляется по схеме: метод `inline ValueType *VALUE(){return ((ValueType*)V + ValueOffst);}`, где `V` — указатель на начало блока, `ValueType` — тип данных свойства, `ValueOffst` — смещение свойства в блоке. Эта схема выполняет позиционирование, а макроподстановка `#define Value VALUE()` имитирует доступ к полю структуры. При этом тип `ValueType` может быть различным для различных свойств, преобразование типа указателя осуществляется методом позиционирования. При таком подходе доступ, например, к номеру левого узла i -й грани выглядит следующим образом: `Face[i].Node[0]` или `Face[i].NodeL` через макросы `#define Node NODE()` и `#define NodeL NODE()[0]` соответственно, где `NODE` — соответствующий метод класса, производного от блочного вектора.

Основные блочные векторы, определяющие расчетную область: координаты узлов, индексы узлов тетраэдров — топология сетки, грани контрольных объемов, поля физических переменных в узлах. Топология сетки ставит в соответствие сеточным элементам (тетраэдрам и граням) набор составляющих их узлов (вершин) сетки. Обратная топология для каждого узла сетки хранит содержащиеся его элементы в стандартном виде построчной разреженной матрицы в формате CSR (Compressed Sparse Row). С помощью обратной топологии осуществляется быстрый доступ к соседним узлам выбранного узла, ко всем тетраэдрам, содержащим узел, или ко всем граням контрольного объема вокруг узла.

3.2.2. Инфраструктура вычислительного ядра. *Менеджер памяти* — промежуточный интерфейс, через который динамически выделяется память. Он контролирует количество затраченной памяти, чтобы избежать проблем с попаданием в режим подкачки (swap), на некоторых системах являющийся проблематичным. Менеджеру может передаваться текстовая метка, идентифицирующая потребителя, что позволяет разработчику контролировать распределение памяти по конкретным структурам данных.

Модуль таймирования — встроенная система ручного инструментального профилирования, замеряющая время для множества участков программы (каналов), идентифицируемых по текстовым меткам. Модуль рассчитан на гибридный параллельный режим и может выполнять измерения для разных нитей и процессов, что дает возможность измерять дисбаланс загрузки. Измерения усредняются на заданном числе шагов по времени для повышения точности измерений и накопления статистики по каналам.

Интерпретатор пользовательского ввода обрабатывает набор входных текстовых файлов параметров, имеющих единую структуру. Для удобства работы с постановкой задачи формат пользовательского ввода был максимально упрощен: каждый файл содержит список пар — именованных параметров и их значений. Каждый параметр может иметь свой тип данных и свойство обязательности. Если не задан обязательный параметр, происходит выдача соответствующей диагностики и остановка, а для необязательного параметра берется значение по умолчанию (также с выдачей диагностики). Модули вычислительного ядра могут иметь свои конфигурационные файлы, что позволяет формировать постановку задачи из готовых блоков.

Счетчик FLOPs измеряет получаемую полезную производительность. Для каждого блока каждой из подпрограмм вычислительного ядра подсчитано фактическое количество операций с плавающей точкой. Количество выполненных операций накапливается в глобальном счетчике и усредняется на заданном числе шагов по времени. Такой подход позволяет достаточно точно оценить реально получаемую производительность, отбросив все сопутствующие операции, не входящие в алгоритм.

Параллельный решатель линейных систем на основе метода бисопряженных градиентов с локальным предобусловливателем [16] применяется при неявном интегрировании по времени.

3.2.3. Алгоритм шага по времени. При запуске вычислительного ядра выполняется инициализация всех подключенных в расчете модулей, загружается сетка, параметры пользовательского ввода, коммуникационная схема и т.д. Затем управление передается главному циклу интегрирования по времени. Шаг по времени имеет следующий алгоритм в случае явной схемы:

- 1) расчет шага по времени;
- 2) цикл многошагового метода Рунге–Кутты:
 - 2.1) расчет пространственной аппроксимации;
 - 2.2) шаг интегрирования по методу Рунге–Кутты;
- 3) обработка результатов.

В случае неявной схемы шаг по времени имеет следующий алгоритм:

- 1) расчет шага по времени;
- 2) цикл итераций по Ньютону:
 - 2.1) расчет пространственной аппроксимации;
 - 2.2) построение якобиана;
 - 2.3) решение линейной системы;
 - 2.4) расчет невязки ньютоновского процесса;
- 3) обработка результатов.

Расчет пространственной аппроксимации состоит из следующих операций:

- 1) расчет конвективных потоков:
 - 1.1) расчет потоков в узлах;
 - 1.2) расчет узловых градиентов;
 - 1.3) реконструкция значений потоков слева и справа от граней контрольных объемов и расчет результирующего потока;
- 2) расчет диссипативных потоков с учетом вклада модели турбулентности;

- 3) расчет вклада источников в правую часть;
- 4) расчет граничных условий.

Схема работы вычислительного ядра показана на рис. 4.

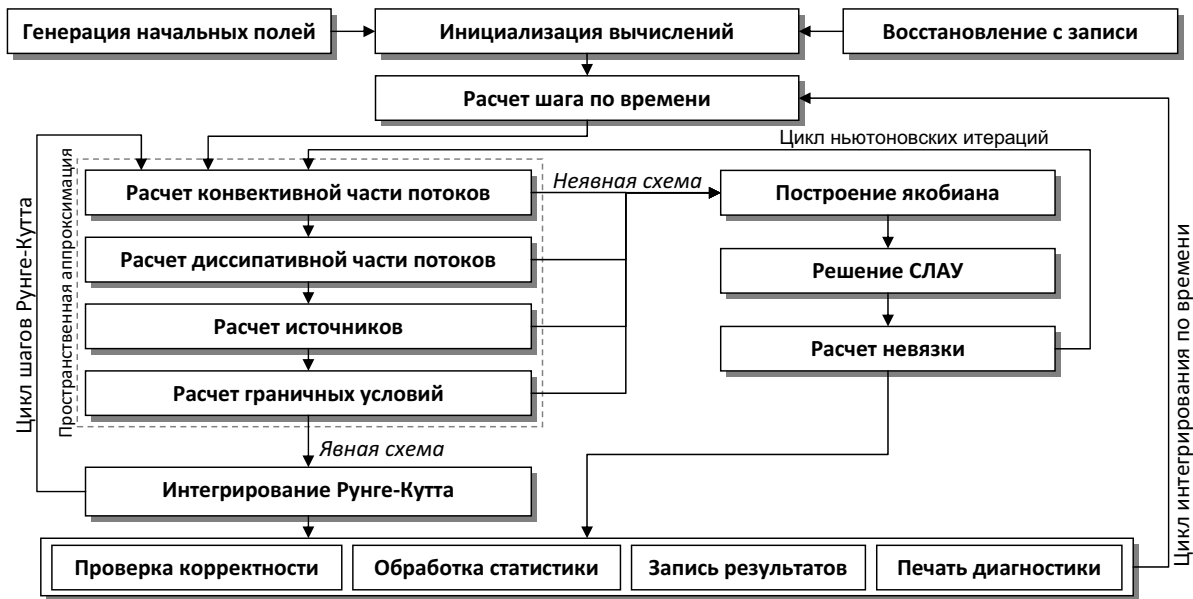


Рис. 4. Схема работы вычислительного ядра

3.3. Средства обработки результатов расчета — постпроцессор. Результаты записываются параллельными процессами в распределенные бинарные файлы специального формата. Файлы состоят из фрагментов данных, для каждого фрагмента указан размер для проверки целостности файла и контрольный код для определения порядка байтов (для переносимости данных между системами с разным порядком байтов, например между архитектурами Intel и IBM). Реализованы следующие типы записей:

- 1) распределенная запись набора полей на расчетной сетке или заданной подсетке — моментальные поля, осредненные поля, записи восстановления счета;
- 2) скалярные динамические данные — эволюция во времени выбранных значений в контрольных точках или глобальных величин (сопротивление, подъемная сила и др.);
- 3) динамические данные на поверхности интегрирования FW/H для моделирования акустики в дальнем поле [10].

Для обработки большого объема данных (запись первого типа может занимать десятки гигабайт, третьего типа — более терабайта) реализован следующий набор параллельных средств:

- суммирование интервалов осреднения; осредненные поля записываются интервалами, поскольку заранее не известен момент выхода течения на статистически однородный режим; после анализа данных и выбора момента начала интегрирования плохие интервалы отбрасываются, а хорошие — объединяются;
- расчет статистики течения, т.е. дополнительных полей, которые можно вычислить на этапе обработки по основным полям, записанным в процессе счета;
- сборка записи из распределенного формата: формирует единый файл в бинарном формате Tecplot для всей расчетной области или для заданной подсетки (сечения или огрубленной сетки);
- сборка интервалов записи динамических данных с автоматическим удалением перекрытий и интерполяцией на постоянный шаг по времени (для фурье-анализа);
- расчет дальнего поля по эволюции на контрольной поверхности для заданных позиций наблюдателя;
- средства фурье-анализа для построения спектров, диаграмм направленности, и т.д.

Структура постпроцессора показана на рис. 5.

4. Параллельные вычисления. Двухуровневое распараллеливание MPI+OpenMP, реализованное в NOISEtte, соответствует современной архитектуре суперкомпьютеров с многоядерными вычислительными модулями (узлами). Суперкомпьютер представляет собой систему с распределенной памятью, так как узлы имеют свое адресное пространство оперативной памяти, при этом на узлах установлены многоядер-

ные процессоры, а узлы сами по себе являются параллельной системой с общей памятью. В многоуровневой модели на первом уровне используется MPI для объединения узлов в рамках модели с распределенной памятью. На втором уровне внутри узлов применяется OpenMP в рамках модели с общей памятью.

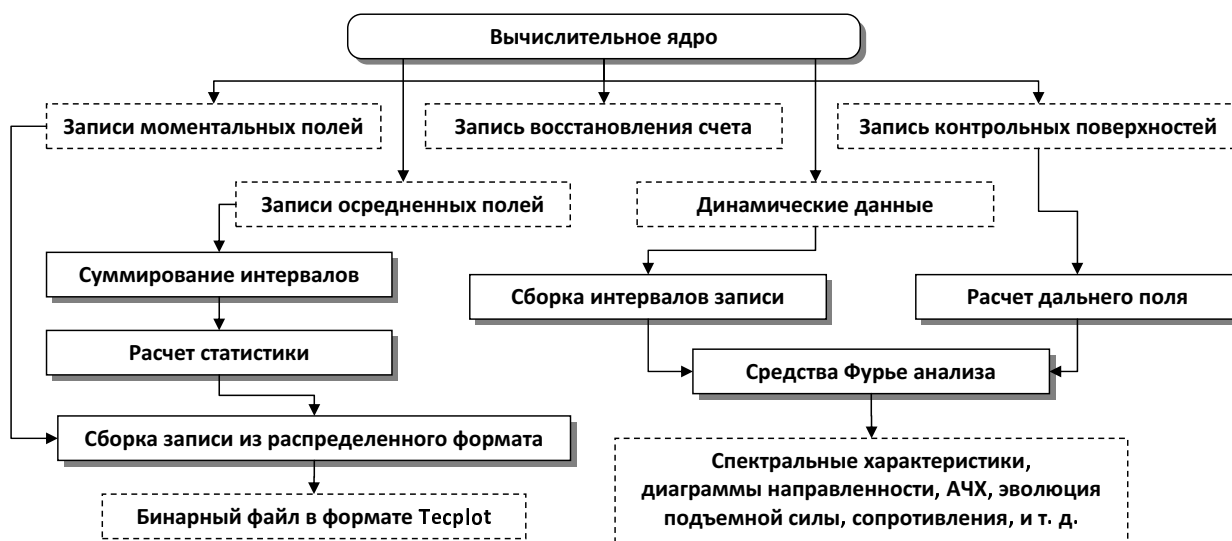


Рис. 5. Схема работы средств обработки результатов расчета

4.1. Распараллеливание MPI+OpenMP. MPI-распараллеливание NOISEtte выполнено на основе традиционного геометрического параллелизма. Расчетная область разбивается на подобласти MPI-процессов. Для обмена данными между соседними подобластями неструктурированной сетки используется неблокирующая пересылка типа точка-точка (MPI_Isend, MPI_Irecv). Более подробно с MPI-распараллеливанием можно ознакомиться в [21].

Использование OpenMP в дополнение к MPI позволяет повысить эффективность при расчетах на большом числе процессоров. Число MPI-процессов P_p уменьшается в P_t раз при том же числе задействованных ядер $P = P_p \times P_t$, где P_t — число нитей. Суммарный объем обмена данными уменьшается примерно в P_t раз пропорционально размеру границ между подобластями, сокращаются потери на латентность за счет уменьшения числа процессов в групповых обменах и повышается скорость обмена, поскольку процессам не приходится разделять коммуникационные ресурсы узла. Однако при этом приходится иметь дело с более сложной параллельной моделью и особенностями OpenMP, которые необходимо учитывать. Типичные проблемы при применении OpenMP в случае неструктурированных сеток, подробно рассматривающиеся, например, в [22], в основном связаны с целостностью доступа к общей памяти, состояниями гонки и др.

Рассмотрим OpenMP-распараллеливание на примере одной из основных операций конечно-объемного алгоритма на неструктурированной сетке — расчет конвективной части потоков через грани контрольных объемов. Расчет состоит из следующих этапов:

- 1) заполнение потоковых переменных (или примитивных переменных в случае MUSCL-аппроксимации [23]) в узлах;
- 2) расчет узловых градиентов — для каждого узла суммируются градиенты по тетраэдрам, содержащим данный узел;
- 3) расчет потоков через грани по схеме Роу с реконструкцией потоковых (или примитивных) переменных.

Первая операция — это цикл по узлам, в котором все итерации выполняются независимо и который может быть без проблем разбит на части, как и другие подобные операции, например, шаг интегрирования по методу Рунге-Кутты. Второй шаг — это цикл по тетраэдрам, в котором осуществляется запись в данные по узлам. Такой цикл уже не может быть просто так разделен, поскольку при одновременной обработке тетраэдров, имеющих общий узел, возникнет конфликт по доступу к данным. Конечно, атомарные операции или критические секции OpenMP решили бы проблему, но при этом скорость стала бы даже ниже, чем в последовательном режиме. Аналогичная ситуация возникает и на третьем шаге, на котором присутствует цикл по граням с записью данных в узлы. В общем случае имеем цикл записи данных по узлам по некоторым элементам расчетной области, которые состоят из двух и более узлов. Для решения

этой проблемы реализованы несколько способов, представленных ниже.

1. Дополнительный массив для хранения промежуточных значений в элементах: значения вычисляются точно так же, но записываются в массив по элементам. Затем выполняется операция суммирования в узлы с использованием обратной топологии. Для этого в цикле по узлам на каждой итерации перебираются элементы, связанные с узлом.

2. Вычисление в несколько тактов: к графу топологии, описывающему связи между элементами и узлами, применяется алгоритм раскраски. В случае обработки граней контрольных объемов узлы графа — это узлы сетки, ребра — это связи узлов через общие грани контрольных объемов. Ребра одного цвета составляют такое подмножество ребер, что каждый из узлов встречается в них не более одного раза. Таким образом, каждое из подмножеств элементов можно обрабатывать параллельно, не опасаясь пересечений по данным.

3. Декомпозиция второго уровня. Подобласть MPI-процесса разбивается далее на части аналогичным образом.

3.1. Затем каждая нить обрабатывает только те элементы, которые имеют хотя бы один узел, принадлежащий своей части, при этом запись результата нить производит только в свои узлы.

3.2. Каждая нить обрабатывает только те элементы, все узлы которых ей принадлежат, а затем главная нить последовательно обрабатывает интерфейсные элементы, имеющие узлы из разных частей. Для расчета узловых градиентов наибольшую эффективность показал вариант 3.2, для расчета потоков — вариант 1, для расчета вязкости (взятие второй производной по пространству) — вариант 3.1. Вариант 2 используется при вычислениях на GPU.

В случае неявной схемы все реализовано аналогично. Для каждой операции добавляется часть, отвечающая за заполнение якобиана, не меняющая общую структуру вычислений. Матрица Якоби представляется в блочном разреженном CSR-формате, матрично-векторное произведение в решателе линейных систем выполняется в цикле по узлам с использованием обратной топологии и не представляет проблемы для распараллеливания.

4.2. Производительность вычислений и параллельная эффективность. Производительность вычислений измеряется встроенным в код счетчиком операций с плавающей точкой и встроенной системой таймирования. При расчетах по явной схеме “чистая” производительность составляет 15–16% от пиковой производительности CPU-ядра, в случае неявной схемы примерно 10–12%, что, по мнению авторов, является высоким показателем для данного класса газодинамических алгоритмов на неструктурированных сетках, для которых характерна, с одной стороны, очень низкая удельная вычислительная стоимость на единицу данных и, с другой стороны, нерегулярный доступ к памяти с неизбежными потерями в кэш в силу неструктурированной топологии.

Для демонстрации работы OpenMP-распараллеливания был выполнен тест на суперкомпьютере “Ломоносов” на разделе с 8-ядерными узлами (процессоры Intel Xeon X5570) для MPI-группы размера 32 и 128 процессов. Специально использовалась небольшая сетка, содержащая всего 2 миллиона узлов. Число OpenMP-нитей варьируется от 1 до 8, задействуется суммарно от 32 до 1024 ядер. Как видно из результатов на рис. 6а, эффективность по OpenMP для 8 нитей составила 67% для группы 32 процесса и 63% для группы 128 процессов. Такое отклонение от линейного ускорения связано с тем, что по мере ускорения вычислений относительный вклад накладных расходов MPI увеличивается в несколько раз. Это подтверждается заметной разницей в 4% между группами 32 и 128 процессов. Без MPI эффективность по OpenMP составила бы около 80%, но такой режим не представляет практического интереса.

Ускорение MPI также показано на тесте со сравнительно небольшой сеткой, содержащей 16 миллионов узлов, чтобы вывести код на близкий к предельному режим, когда параллелизм практически исчерпывается. Тест выполнен на суперкомпьютере “Ломоносов” на разделе с 8-ядерными узлами, библиотека MPI OpenMPI 1.4. Задействовано до 24 000 ядер, что соответствует загрузке всего 670 узлов сетки на ядро, близкой к минимальной (в [24] приводятся тесты NOISEtte до 500 узлов на ядро). Тест соответствует расчету реальной задачи — обтекание цилиндра дозвуковой струей, выполненному в рамках совместных с Центральным аэрогидродинамическим институтом (ЦАГИ) исследований источников шума в турбулентном следе [25]. Результаты полученного ускорения при переходе с 1024 до 24000 ядер показаны на рис. 6б. Время вычислений на 1 шаг метода Рунге–Кутты составило 0.65 сек. на 1024 ядрах и 0.04 сек. на 24 000 ядрах.

5. Приложения. Комплекс программ NOISEtte предназначен для расчета задач аэродинамики и аэроакустики. С его помощью могут решаться как модельные задачи, направленные на фундаментальные исследования, так и задачи, связанные с моделированием реальных турбулентных течений в сложных геометрических областях. Благодаря реализованным алгоритмам повышенной точности, NOISEtte обеспе-

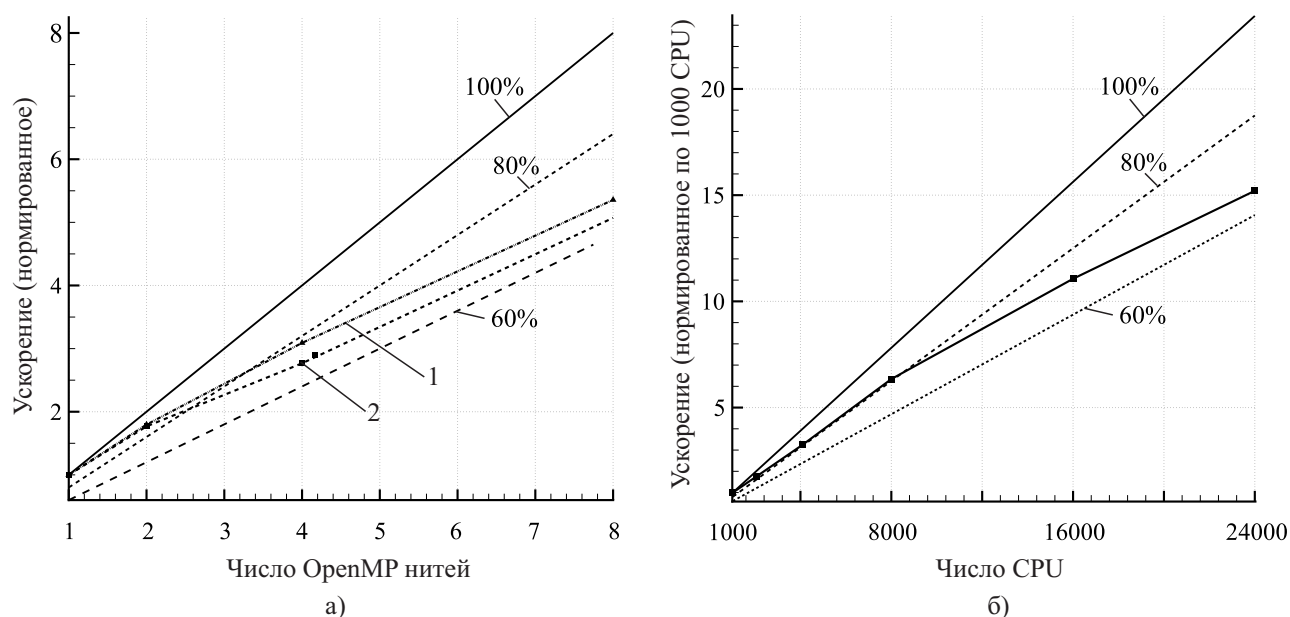


Рис. 6. Ускорение на суперкомпьютере “Ломоносов”: а) ускорение с OpenMP для групп 32 (линия 1) и 128 (линия 2) MPI-процессов, б) ускорение с MPI, 8 OpenMP-нитей на процесс

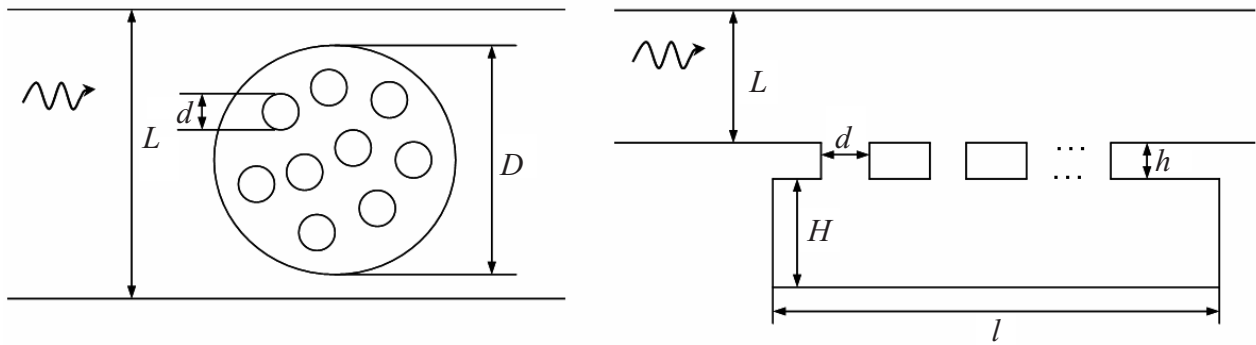
чивает возможность численного воспроизведения нестационарных режимов и окружающих акустических полей, что является отличительной особенностью комплекса.

Помимо аэродинамических и аэроакустических приложений, NOISEtte может применяться для других задач, поддающихся математическому описанию при помощи уравнений газовой динамики для идеального сжимаемого газа. Он также допускает расширение для моделирования динамики реальных газов.

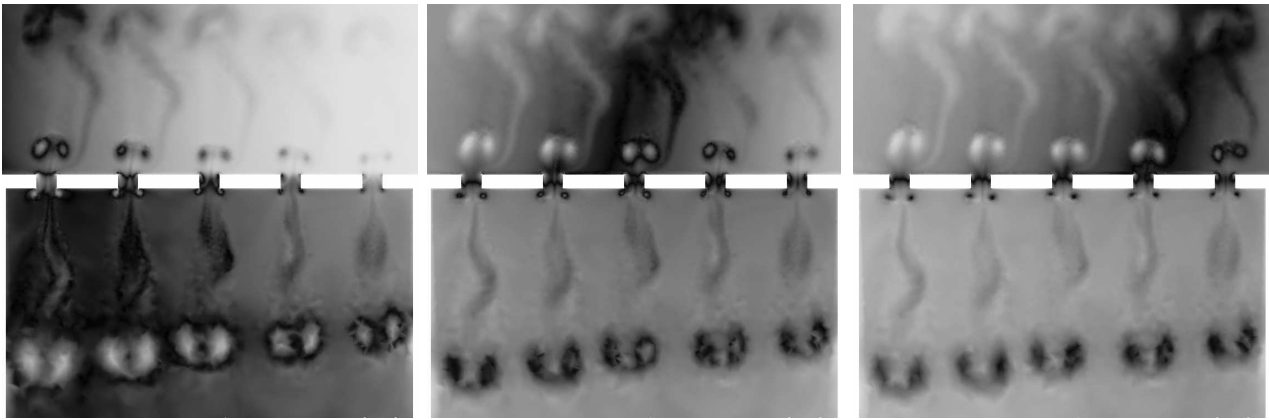
5.1. Верификация и валидация. Верификация численных алгоритмов, реализованных в программном комплексе NOISEtte, проводилась на множестве тестовых задач с известными точными или эталонными решениями. Среди таких задач: распространение начального возмущения в виде гауссового импульса при наличии постоянного фонового течения [18], расчет мощности акустического излучения в дальнем поле от монопольного точечного источника в дозвуковом потоке [26], рассеяние акустических волн на цилиндре [27], рассеяние акустических волн на единичном вихре (вихрь Рэнкина) [28, 29], распад произвольного разрыва [30], течения в канале, сверхзвуковое течения вокруг прямого уступа [31], расчет характеристик сверхзвукового течения газа в плоском канале с клином [32].

Валидация комплекса осуществлялась, в основном, на задачах о моделировании турбулентных течений путем сравнения результатов с известными эмпирическими законами и/или имеющимися экспериментальными данными. В частности, корректность результатов, получаемых при помощи комплекса NOISEtte, подтверждена на решениях следующих задач: моделирование изотропной турбулентности [33], ламинарного и турбулентного пограничных слоев на пластине [34], турбулентного течения в канале и турбулентного течения вокруг обратного уступа в канале [9], расчет стационарных аэродинамических характеристик профилей (NACA0012, NACA23012 и др.), крыла ONERA M6 [35], модели DPW4 [36] и др.

5.2. Исследовательские расчеты. Комплекс программ NOISEtte использовался при решении ряда исследовательских задач, ориентированных на реальные приложения в аэродинамике и аэроакустике. Примером одной из таких задач является численное исследование поглощения акустической энергии падающих волн при прохождении звукопоглощающих конструкций резонансного типа, широко применяющихся для снижения шума газотурбинных двигателей. В процессе исследования ставился специальный вычислительный эксперимент на микроуровне, а именно, рассматривался один резонатор реальной конфигурации, “вмонтированный” в стенку волновода [37]. На вход волновода подавалось звуковое излучение в виде плоских волн на фоне входящего дозвукового течения (или без него). На основе измерений мощности акустических возмущений до и после резонатора проводился расчет характеристик исследуемого резонатора — коэффициента прохождения и акустического импеданса. На рис. 7а представлена схема расчета задачи с перфорированным резонатором. Рис. 7б показывает поля возмущения плотности в окрестности резонаторных отверстий на разные моменты времени. Видно, что система вихревых колец, попеременно образующихся сверху и снизу от отверстия, с одной стороны, приводит к оттоку акустической энергии проходящих по волноводу волн за счет ее трансформации в энергию вихревого движения, а с другой

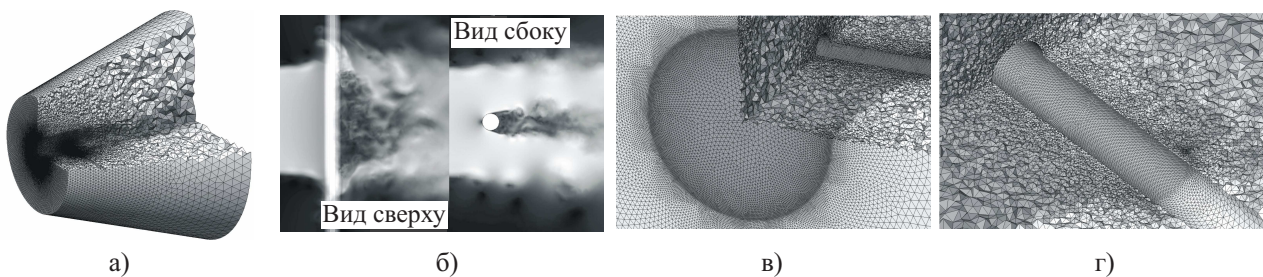


а)



б)

Рис. 7. Схема расчета задачи с перфорированным резонатором (а), поле плотности в центральном сечении в разные моменты времени при прохождении акустической волны (б)



а)

б)

в)

г)

Рис. 8. Пример тетраэдральной сетки (а)–(г) для сложной расчетной области — сгущение одновременно к границе круглой струи и к поверхности поперечного цилиндрического препятствия, б) показана картина течения (поле модуля скорости) при обтекании цилиндра струей

стороны, препятствует “работе” линейного резонанса.

Еще один пример — совместное с ЦАГИ численное исследование структуры распределенного акустического источника, формирующегося в турбулентном следе за цилиндром, и возможностей влияния на него с целью уменьшения шума в дальнем поле. В основе данного исследования лежит идея, предложенная учеными ЦАГИ, о снижении акустической мощности источника в результате отсечения сектора с тыльной стороны цилиндра [25]. Фундаментальное исследование имеет явную прикладную направленность, так как цилиндры при рассматриваемых режимах течения имитируют стойки шасси самолета, шум при обтекании которых на этапе посадки негативно влияет на экологию аэропортов и прилегающих к ним территорий.

На первом этапе работы моделировалась круглая струя, набегающая на поперек лежащий цилиндр. Такая постановка соответствует проводимому в ЦАГИ физическому эксперименту. На рис. 8 показана структура используемых в расчетах сложных неструктурированных тетраэдральных сеток размерностью

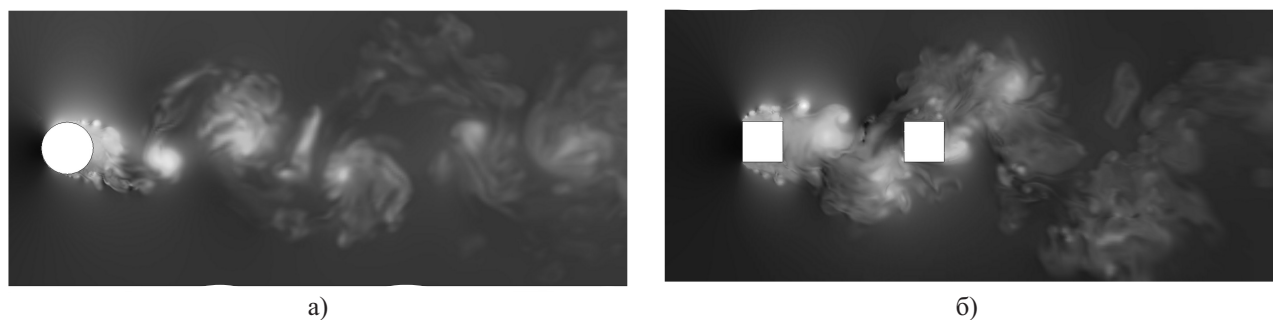


Рис. 9. Поле плотности при обтекании круглого цилиндра (а) и системы из двух цилиндров с квадратным сечением (б)

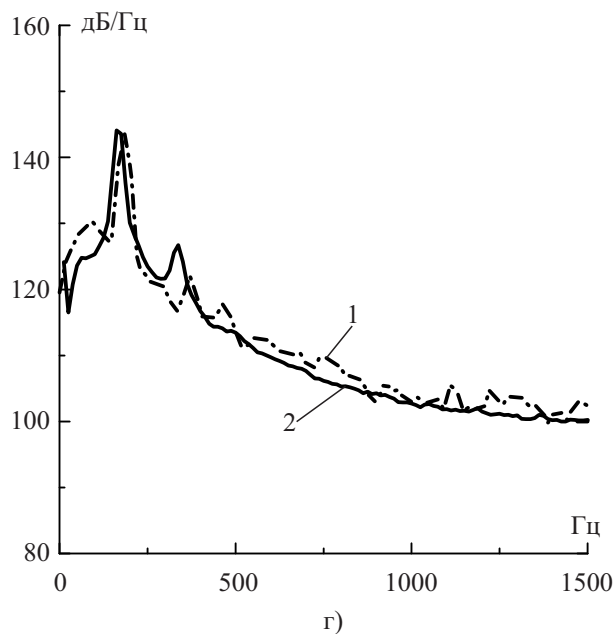
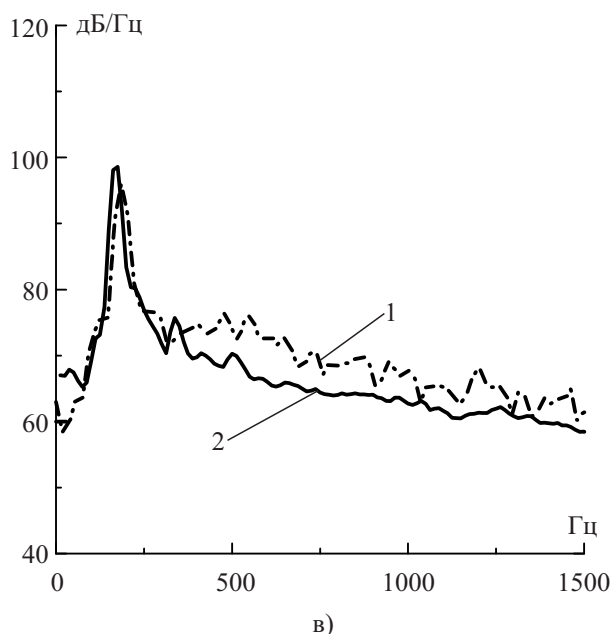
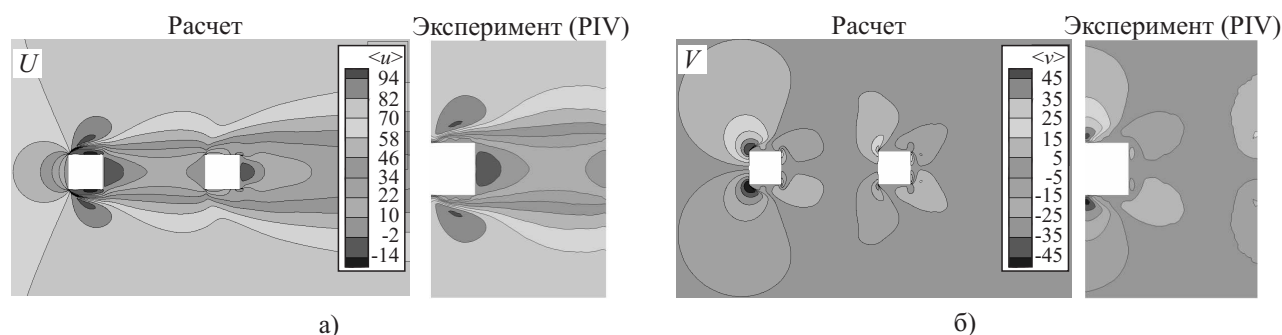


Рис. 10. Сравнение с экспериментом: осредненные поля продольной (а) и вертикальной (б) компоненты скорости в центральном сечении; акустический спектр в дальнем поле (в) и спектр пульсаций поверхностного давления (г); линия 1 — численные результаты, линия 2 — эксперимент

до 16 миллионов узлов и 100 миллионов тетраэдров. Кроме того, на рис. 8б показано поле модуля скорости в системе “струя–цилиндр”, которое дает представление о сложном характере турбулентного течения.

На рис. 9 показаны поля плотности при обтекании круглого цилиндра и системы из двух цилиндров с квадратным сечением. Задача о турбулентном течении вокруг тандема цилиндров с квадратным сечением является модельной постановкой, рассматриваемой в рамках европейского проекта FP7 VALIANT, посвященного исследованию возможностей численного моделирования для решения реальных инженерных задач по предсказанию аэродинамического шума при обтекании конструктивных элементов планера. Проведенные при помощи комплекса NOISEtte расчеты показали полное соответствие численных резуль-

татов экспериментальным данным, полученным аэрокосмическим центром NLR (Нидерланды), одним из партнеров по проекту VALIANT.

На рис. 10 в качестве примера показано сравнение с экспериментом для осредненных полей течения, для спектра пульсаций поверхностного давления на препятствиях и для спектра в дальнем поле для одной из множества позиций наблюдателя.

6. Заключение. Представленный в настоящей статье программный комплекс NOISEtte, основанный на схемах повышенной точности с центрами в узлах для неструктурированных сеток, представляет собой одновременно как инструмент для решения исследовательских фундаментальных задач, выполнения крупномасштабных суперкомпьютерных расчетов, отработки новых методов и алгоритмов, так и средство для реальных промышленных приложений. Программный комплекс NOISEtte использовался при проведении совместных исследований с ОАО “ОКБ Сухого”, ОАО “Авиадвигатель”, ЦНИИМаш, ЦАГИ, ФАЛТ МФТИ и ОАО “Камов”. Он также задействован в европейском проекте FP7 VALIANT, где совместно с ЦАГИ, ONERA, NLR, TUB и другими партнерами из 6 европейских стран проводятся исследования потенциала современных численных методов и подходов для моделирования аэродинамического шума. Многоуровневое распараллеливание MPI+OpenMP позволяет задействовать десятки тысяч процессорных ядер суперкомпьютеров, параллельные средства инфраструктуры позволяют выполнять расчеты на неструктурированных сетках с числом элементов более миллиарда.

В настоящее время активно ведутся работы по внедрению гетерогенных вычислений и адаптации NOISEtte к гибридным суперкомпьютерам с массивно-параллельными ускорителями — графическими процессорами GPU (Graphics Processing Unit). GPU-устройства задействуются в рамках многоуровневой параллельной модели посредством открытого вычислительного языка OpenCL. Выбор OpenCL обусловлен тем, что, в отличие, например, от CUDA, принадлежащего NVIDIA, это открытый стандарт, который поддерживается основными производителями компьютерного оборудования, в том числе Intel, AMD, IBM, NVIDIA, Apple, Sony. Использование этого языка позволит применять вычислительное программное обеспечение на системах с различными типами ускорителей, такими как GPU производства NVIDIA и AMD (ATI), ускорители Intel MIC, гибридные процессоры-ускорители CELL, модули программируемой логики (FPGA).

Работа выполнена при поддержке РФФИ (коды проектов 12-01-00486-а, 11-01-12096-офи-м-2011) и Европейской комиссии (проект FP7 VALIANT). Для расчетов использовались суперкомпьютеры “Ломоносов” НИВЦ МГУ и K100 ИПМ РАН. Авторы выражают благодарность этим организациям.

СПИСОК ЛИТЕРАТУРЫ

1. *Singer B.A., Guo Y.* Development of computational aeroacoustics tools for airframe noise calculations // *Int. J. of Computational Fluid Dynamics*. 2004. **18**, N 6. 455–469.
2. *Abalakin I., Dervieux A., Kozubskaya T.* Computational study of mathematical models for noise DNS // *AIAA Paper 2002-2585*. 2002.
3. *Abalakin I., Dervieux A., Kozubskaya T.* High accuracy finite volume method for solving nonlinear aeroacoustics problems on unstructured meshes // *Chinese J. of Aeronautics*. 2006. **19**, N 2. 97–104.
4. *Vos J.B., Rizzi A., Darracq D., Hirschel E.H.* Navier–Stokes solvers in European aircraft design // *Progress in Aerospace Sciences*. 2002. **38**, N 8. 601–697.
5. *Spalart P.R., Allmaras S.R.* A one-equation turbulence model for aerodynamic flows // *30th Aerospace Science Meeting, Reno, Nevada, 1992*. AIAA Paper 92-0439, 1992.
6. *Haase W., Braza M., Revell A.* DESider — A European effort on hybrid RANS-LES modelling. Berlin: Springer, 2009.
7. *Волков К.Н., Емельянов В.Н.* Моделирование крупных вихрей в расчетах турбулентных течений. М.: Физматлит, 2008.
8. *Spalart P.R., Deck S., Shur M., Squires K., Strelets M., Travin A.* A new version of detached-eddy simulation, resistant to ambiguous grid densities // *Theor. Comput. Fluid Dyn.* 2006. **20**, N 3. 181–195.
9. *Shur M., Spalart P.R., Strelets M., Travin A.* A hybrid RANS-LES approach with delayed-DES and wall-modelled LES capabilities // *Int. J. of Heat and Fluid Flow*. 2008. **29**, N 6. 1638–1649.
10. *Бахвалов П.А., Козубская Т.К., Корнилина Е.Д., Морозов А.В., Яковлевский М.В.* Технология расчета акустических возмущений в дальнем поле течения // *Матем. моделирование*. 2011. **23**, № 11. 33–47.
11. *Barth T.* Numerical methods for conservation laws on structured and unstructured meshes // *VKI for Fluid Dynamics. Lectures series*. 2003-03. Sint-Genesius-Rode, Belgium: von Karman Institute of Fluid Dynamics, 2003.
12. *Флетчер К.* Численные методы на основе метода Галеркина. М.: Мир, 1988.
13. *Roe P.L.* Approximate Riemann solvers, parameter vectors and difference schemes // *J. Comput. Phys.* 1981. **43**, N 2. 357–372.

14. *Huang L.C.* Pseudo-unsteady difference schemes for discontinuous solution of steady-state. One-dimensional fluid dynamics problems // *J. Comput. Phys.* 1981. **42**, N 1. 195–211.
15. *Абалакин И.В., Козубская Т.К.* Многопараметрическое семейство схем повышенной точности для линейного уравнения переноса // *Матем. моделирование.* 2007. **19**, № 7. 56–66.
16. *Saad Y.* Iterative methods for sparse linear systems. Second Edition. Philadelphia: SIAM, 2003.
17. *Hirsch Ch.* Numerical computation of internal and external Flows. Vol. 2: Computational Methods for Inviscid and Viscous Flows. New York: Wiley, 1990.
18. *Tam C.K.W., Webb J.C.* Dispersion-relation-preserving finite difference schemes for computational acoustics // *J. Comput. Phys.* 1993. **107**, N 2. 262–281.
19. *Schloegel K., Karypis G., Kumar V.* Parallel static and dynamic multi-constraint graph partitioning // *Concurrency and Computation: Practice and Experience.* 2002. **14**, N 3. 219–240.
20. *Головченко Е.Н.* Параллельный пакет декомпозиции больших сеток // *Матем. моделирование.* 2011. **23**, № 10. 3–18.
21. *Gorobets A.V., Abalakin I.V., Kozubskaya T.K.* Technology of parallelization for 2D and 3D CFD/CAA codes based on high-accuracy explicit methods on unstructured meshes // *Parallel Computational Fluid Dynamics 2007. Lecture Notes in Computational Science and Engineering.* Vol. 67. Berlin: Springer, 2009. 253–260.
22. *Aubry R., Houzeaux G., Vazquez M., Cela J.M.* Some useful strategies for unstructured edge-based solvers on shared memory machines // *Int. J. Numer. Meth. Engng.* 2011. **85**, N 5. 537–561.
23. *Dervieux A., Debiez C.* Mixed element volume MUSCL methods with weak viscosity for steady and unsteady flow calculation // *Computer and Fluids.* 1999. **29**, N 1. 89–118.
24. *Савин Г.И., Четверушкин Б.Н., Суков С.А., Горобец А.В., Козубская Т.К., Вдовикин О.И., Шабанов Б.М.* Моделирование задач газовой динамики и аэроакустики с использованием ресурсов суперкомпьютера МВС-100К // *Докл. РАН.* 2008. **423**, № 3. 312–315.
25. *Kopiev V., Abalakin I., Faranosov G., Gorobets A., Kozubskaya T., Ostrikov N., Zaitsev M.* Experimental and numerical localization of noise sources for cylinder in round jet // *Proc. of Trilateral Russian–French–German Workshop on Computational Experiment In Aeroacoustics.* Svetlogorsk, September 22–25, 2010. Moscow: MAKS Press, 2010. 75–78.
26. *Crighton D.G., Dowling A.P., Ffowcs Williams J.E., Heckl M.A., Leppington F.A.* Modern methods in analytical acoustics. Berlin: Springer, 1992.
27. *Morris P.J.* Scattering of sound by a sphere: category 1: problems 3 and 4 // *Proc. of Second Computational Aeroacoustics (CAA) Workshop on Benchmark Problems.* NASA Conference Publication 3352. Hampton: Langley Research Center, 1997. 15–17.
28. *Karabasov S.A., Kopiev V.F., Goloviznin V.M.* On a classical problem of acoustic wave scattering by a free vortex: numerical modelling // *Proc. 15th AIAA/CEAS Aeroacoustics Conference, Miami, FL, 11–13 May 2009.* AIAA Paper 2009-3234, 2009.
29. *Colonijs T., Lele S.K., Moin P.* The scattering of sound waves by a vortex: numerical simulations and analytical solutions // *J. Fluid Mech.* 1994. **260**. 271–298.
30. *Жмакин А.И., Фурсенко А.А.* Об одной монотонной разностной схеме сквозного счета // *Ж. вычисл. матем и матем. физ.* 1980. **20**, № 4. 1021–1031.
31. *Woodward P., Colella P.* The numerical simulation of two-dimensional fluid flow with strong shocks // *J. Comput. Phys.* 1984. **54**, N 1. 115–173.
32. *Du X., Corre C., Lerat A.* A third-order finite-volume residual-based scheme for the 2D Euler equations on unstructured grids // *J. Comput. Phys.* 2011. **230**, N 11. 4201–4215.
33. *Bunge U., Mockett C., Thiele F.* Guidelines for implementing detached-eddy simulation using different models // *Aerospace Science and Technology.* 2007. **11**, N 5. 376–385.
34. *Лойцянский Л.Г.* *Механика жидкости и газа.* М.: Дрофа, 2003.
35. *Schmitt V., Charpin F.* Pressure distributions on the ONERA-M6-Wing at transonic Mach numbers. Experimental data base for computer program assessment. Report of the Fluid Dynamics Panel Working Group 04, AGARD AR 138. 1979.
36. *Vassberg J.C., DeHaan M.A., Rivers S.M., Wahls R.A.* Development of a common research model for applied CFD validation studies. AIAA Paper 2008-6919, 2008.
37. *Дубень А.П., Козубская Т.К., Миронов М.А.* Численное исследование резонаторов в волноводе // *Изв. РАН. Механ. жидкости и газа.* 2012. № 1. 146–156.

Поступила в редакцию
27.08.2012