

Санкт-Петербургский государственный политехнический университет
Институт международных образовательных программ
Кафедра физики и математического моделирования в механике

Диссертация допущена к защите

Зав. кафедрой

_____ В.В.Краснощеков

“25” июня 2014 г.

ДИССЕРТАЦИЯ
на соискание ученой степени
МАГИСТРА

Тема: **МАССИВНАЯ ПАРАЛЛЕЛИЗАЦИЯ ГИДРОДИНАМИЧЕСКИХ
ЗАДАЧ ДЛЯ ПРОВЕДЕНИЯ РАСЧЕТОВ НА СОВРЕМЕННЫХ
СУПЕРКОМПЬЮТЕРАХ**

Направление: 151600 «Прикладная механика»

Магистерская программа: «Математическое моделирование механических
систем»

Выполнила:

студентка гр. 6145/10

_____ (подпись)

Уварова С. А.

Руководитель,

(кандидат физ.-мат. наук, доцент)

_____ (подпись)

Гарбарук А.В.

г. Санкт-Петербург

2014 г.

Оглавление

1. ВВЕДЕНИЕ.....	3
2. СУПЕРКОМПЬЮТЕРЫ И ИХ ПРИМЕНЕНИЕ	6
3. ПРОГРАММА ПОДГОТОВКИ ДАННЫХ ДЛЯ МАССИВНО-ПАРАЛЛЕЛЬНЫХ РАСЧЕТОВ	16
3.1 Описание программы.....	17
3.2. Тестирование программы	21
3.3. Демонстрация эффективности программы.....	21
4. МОДЕЛИРОВАНИЕ ОБТЕКАНИЯ КРЫЛОВОГО ПРОФИЛЯ В АЭРОДИНАМИЧЕСКОЙ ТРУБЕ	24
4.1 Постановка задачи.....	25
4.2 Метод решения	25
4.3 Результаты расчетов	26
5. ЗАКЛЮЧЕНИЕ	29
6. СПИСОК ЛИТЕРАТУРЫ.....	30
Приложение 1. Форматы файлов «NTS code».....	32

1. ВВЕДЕНИЕ

На каждом этапе развития средств вычислительной техники всегда находятся задачи, требующие для своего решения вычислительных мощностей, которые превышают мощности компьютеров, доступных на рынке. Поэтому во все времена развития компьютерной отрасли ведущие страны мира занимались разработкой, реализацией и внедрением все более мощной вычислительной техники. Скорость развития ЭВМ подчиняется эмпирическому закону, сформулированного Гордоном Муром в 1965 г. Из всех известных формулировок этого закона его основную суть наилучшим образом отражает вариант: «производительность вычислительных систем удваивается каждые 24 месяца». Стоит отметить, что в последние несколько лет возможности увеличения мощности процессоров на основе повышения тактовой частоты оказались фактически исчерпаны, а для повышения производительности машин стали использовать распараллеливание на различных уровнях, что активно используется при построении современных суперкомпьютеров.

Суперкомпьютер ([англ. supercomputer](#), СуперЭВМ) - это вычислительная машина, значительно превосходящая по своим техническим параметрам большинство существующих на данный момент компьютеров. Существует два подхода построения суперкомпьютеров – системы с общей памятью и так называемые кластеры.

Для систем с общей памятью характерны единая оперативная память, единая операционная система, единая подсистема ввода-вывода и только процессоры образуют множество. В таких суперкомпьютерах операционная система сама распределяет процессы по процессорам и выделяет им ресурсы. Следует отметить, что затраты на распределение ресурсов и управление растут существенно быстрее роста количества процессоров, поэтому такой подход давно проиграл кластерному типу построения суперкомпьютеров.

Вычислительный кластер – это мультикомпьютер, состоящий из множества отдельных компьютеров (узлов), связанных между собой единой коммуникационной системой. При этом общей физической оперативной памяти для узлов не существует, а каждый узел имеет свою локальную оперативную память. Коммуникационная система позволяет узлам взаимодействовать между собой только посредством передачи сообщений, поэтому для эффективного использования кластеров необходимо обеспечить равномерную загрузку узлов кластера с помощью распараллеливания задачи.

Распараллеливание задачи сводится к структуризации, то есть разбиению поставленной задачи на подзадачи (так называемые блоки) меньшего размера, которые не требуют последовательного исполнения и могут, соответственно, быть выполнены на разных процессорах независимо друг от друга. Однако при разбивке необходимо учитывать, что производительность системы будет определяться самым медленным ее компонентом и поэтому для эффективного использования кластеров нагрузка на каждый узел должна быть одинакова. Если алгоритм расчета для всех блоков одинаков, то это значит, что размер всех подзадач (блоков) должен быть одинаков. Кроме того, желательно обеспечить минимальный обмен данными между блоками, поскольку это также влияет на эффективное использование ресурсов кластеров. Наиболее распространенным способом обмена данными в параллельном программировании является MPI (Message Passing Interface), который может быть применен как для расчетов на системах с раздельной памятью, так и для расчетов на многопроцессорных и многоядерных системах.

Одной из программ, в которой успешно реализована параллелизация расчетов, является солвер «NTS code». В программе используется двухуровневая параллельность. На верхнем уровне при помощи средств MPI задача разбивается на несколько групп MPI. Каждая группа MPI содержит один или несколько блоков сетки, которые обмениваются граничными условиями. На нижнем уровне при помощи OMP производится

распараллеливание рассматриваемой группы MPI. Однако, как это уже было сказано, при распараллеливании эффективность использования кластера существенно зависит от равномерности загрузки процессоров поэтому блоки, на которые разбивается задача, должны быть одинаковы, так как скорость решения задачи будет определяться наиболее медленным блоком.

В настоящее время разбивка в «NTS code» происходит вручную, что значительно увеличивает трудозатраты при постановке и решении задачи. Автоматизация разбивки на блоки позволит существенно упростить постановку, ускорить решение задач и более эффективно использовать ресурсы кластеров.

Целью работы является:

- разработка комплекса программ для разбиения данных газодинамического солвера «NTS code» на произвольное количество блоков;
- тестирование разработанного комплекса программ;
- демонстрация его эффективности на примере расчета трехмерного крылового профиля при докритическом угле атаки.

2. СУПЕРКОМПЬЮТЕРЫ И ИХ ПРИМЕНЕНИЕ

В последние годы в применении суперкомпьютеров произошли революционные изменения. На сегодняшний день мощность суперкомпьютерных технологий позволяет производить моделирование реальных процессов самого широкого спектра задач науки, образования и промышленности. Ведущие исследовательские организации и промышленные предприятия в мире используют суперкомпьютерные ресурсы, расположенные на базе университетов и национальных лабораторий. В России ведущие научно-исследовательские и промышленные организации также применяют методы компьютерного моделирования и высокопроизводительных расчетов с использованием суперкомпьютеров.

Традиционной сферой применения суперкомпьютеров всегда были научные исследования, где человеческие знания всегда опирались на теорию и опыт. Однако теперь ученые сталкиваются с тем, что многие испытания стали практически невозможными - в некоторых случаях из-за своих масштабов, в других - дороговизны или опасности для здоровья и жизни людей. В этот момент и приходят на помощь мощные компьютеры. Позволяя экспериментировать с электронными моделями реальной действительности, они становятся «опорой» современной науки и производства.

Среди задач, решаемых сегодня на суперкомпьютерах, можно отметить следующие: физика плазмы и статистическая механика, физика конденсированных сред, молекулярная и атомная физика, теория элементарных частиц, газовая динамика и теория турбулентности, астрофизика. В химии - различные области вычислительной химии: квантовая химия (включая расчеты электронной структуры для целей конструирования новых материалов, например, катализаторов и сверхпроводников), молекулярная динамика, химическая кинетика, теория поверхностных явлений и химия твердого тела, конструирование лекарств. Естественно, что ряд областей применения находится на стыках

соответствующих наук, например, химии и биологии, и перекрывается с техническими приложениями. Так, задачи метеорологии, изучение атмосферных явлений и, в первую очередь, задача долгосрочного прогноза погоды, для решения которой постоянно не хватает мощностей современных суперЭВМ, тесно связаны с решением ряда перечисленных выше проблем физики. Среди технических задач, для решения которых используются суперкомпьютеры, можно выделить задачи аэрокосмической и автомобильной промышленности (например, этапы проектирования, моделирование краш-тестов), ядерной энергетики, предсказания и разработки месторождений полезных ископаемых, нефтедобывающей и газовой промышленности (в том числе проблемы эффективной эксплуатации месторождений, особенно трехмерные задачи их исследования), и, наконец, конструирование новых микропроцессоров и компьютеров, в первую очередь самих суперЭВМ.

Суперкомпьютеры применяются и для военных целей. Кроме очевидных задач разработки оружия массового уничтожения и конструирования самолетов и ракет, можно упомянуть, например, конструирование бесшумных подводных лодок и др. Самый знаменитый пример - это американская программа СОИ (англ. *Strategic Defense Initiative*, стратегическая оборонная инициатива), объявленная президентом США Рональдом Рейганом 23 марта 1983 года, как долгосрочная программа научно-исследовательских и опытно-конструкторских работ. Основной целью СОИ являлось создание научно-технического задела для разработки широкомасштабной системы противоракетной обороны с элементами космического базирования, исключающей или ограничивающей возможное поражение наземных и морских целей из космоса.

Суперкомпьютеры являются особым классом вычислительной техники, который развивается гигантскими темпами. Для измерения их производительности используется единица FLOPS (также flops, flop/s, флопс или флоп/с, англ. *FL*oating-point *O*perations *P*er *S*econd, произносится как

флопс) — внесистемная единица, используемая для измерения производительности компьютеров, показывающая, сколько операций с плавающей запятой в секунду выполняет данная вычислительная система. Как и большинство других показателей производительности, данная величина определяется путём запуска на испытуемом компьютере тестовой программы, которая решает задачу с известным количеством операций и подсчитывает время, за которое она была решена. Наиболее популярным тестом производительности на сегодняшний день является программа Linpack, используемая, в том числе, при составлении рейтинга суперкомпьютеров Топ-500.

Список Топ-500 – это рейтинг самых высокопроизводительных вычислительных систем мира, обновляемый дважды в год. Представленные в списке Топ-500 данные позволяют проследить характерные тенденции развития индустрии в сфере суперкомпьютерных вычислений. Места в списке распределяются по реальной производительности, достигнутой при выполнении теста Linpack. Больше всего в списке в настоящее время представлено кластерных систем, в частности, они занимают все первые места. Последняя редакция списка, а именно 42-й рейтинг Топ-500 огласили на конференции SC13, состоявшейся в Денвере в ноябре 2013 года, в котором числятся мощнейшие суперкомпьютеры на сегодняшний день. На официальном сайте Топ-500 можно ознакомиться с названиями суперкомпьютеров разных лет — <http://top500.org/timeline/>.

Новый китайский суперкомпьютер «Тяньхэ-2» («Млечный путь-2») возглавил свежий рейтинг Топ-500 самых мощных вычислительных систем в мире. Производительность этого суперкомпьютера по тесту Linpack составляет 33,86 PFLOPS, а теоретическая пиковая производительность – 54,9 PFLOPS, что почти вдвое превышает производительность прежнего лидера из октябрьского списка Топ-500, суперкомпьютера Titan, который теперь опустился на второе место с показателем по тесту Linpack – 17,59 PFLOPS и пиковой производительностью – 27,11 PFLOPS. Машина

спроектирована и собрана Китайским национальным университетом оборонных технологий (NUDT). Тяньхэ-2 состоит из 16 тысяч узлов, каждый из которых включает в себя 2 процессора Intel Xeon E5-2692 на архитектуре Ivy Bridges 12 ядрами каждый (частота 2,2 ГГц) и 3 специализированных сопроцессора Intel Xeon Phi 31S1P (на архитектуре Intel MIC, по 57 ядер на ускоритель, частота 1,1 ГГц, пассивное охлаждение). На каждом узле установлено 64 ГБ (16 модулей) оперативной памяти типа DDR3 ECC и дополнительно по 8 ГБ GDDR5 в каждом Xeon Phi (всего 88 ГБ). В общей сложности, количество вычислительных ядер достигает 3,12 миллиона (384 тысячи Ivy Bridge и 2736 тыс. Xeon Phi). За полгода после создания «Тяньхэ-2» он нашел успешное применение в разработке крупного авиалайнера C919, в сферах наблюдения над Землей с высокой степенью разрешения, определения последовательности расположения генов, биомедицины, интеллектуальных городов, электронного административного управления, облачных вычислений, информационных услуг и других областях.

В последнюю редакцию рейтинга Топ-500 вошли пять крупнейших суперкомпьютеров России. Наиболее высокопроизводительной системой в России на данный момент является суперкомпьютер, установленный в МГУ имени М.В.Ломоносова, построенный компанией «Т-Платформы». Суперкомпьютер имеет гибридную архитектуру. В суперкомпьютере используются восемь видов вычислительных узлов и процессоры с различной архитектурой, что позволяет получить необходимую производительность для широкого спектра приложений. Кроме классических многоядерных узлов архитектуры x86 на базе четырехъядерных и шестиядерных процессоров Intel Xeon X5570 Nehalem и X5670 Westmere, в системе имеются гибридные узлы TB2-TL на процессорах Intel Xeon и Nvidia Tesla, а также узлы на процессорах PowerXCell 8i. Кроме того, в суперкомпьютере используется 30 вычислительных узлов на базе процессоров Cell от IBM. Оперативная память суперкомпьютера составляет 92 ТБ, а общий объем дисковой памяти вычислителя – 1,75 ПБ. Согласно

рейтингу вычислительных систем России и СНГ за сентябрь 2013 г. эта система занимает первое место. В мировом рейтинге Топ-500 за ноябрь 2013 г. «Ломоносов» занимает 37-ю строчку. Пиковая производительность данного суперкомпьютера составляет 1,7 PFLOPS, а по тесту Linpack составляет 872,5 TFLOPS (Терафлопс). Суперкомпьютерные ресурсы МГУ используется, в первую очередь, для выполнения фундаментальных научных исследований в нефтегазовой, аэрокосмической, биомедицинской, атомной и многих других научных областях. Среди таких задач - масштабные работы по глобальному изменению климата и динамике мирового океана, обработке сейсмических данных, постгеномной медицине, механизмам формирования галактик и другие. В настоящее время на базе этого суперкомпьютера уже был реализован целый ряд научных проектов по заказу как государственных, так и коммерческих корпораций.

Так, например, специалисты компании «Деко-геофизика» апробировали решение прямой задачи сейсморазведки для заданной глубинно-скоростной модели при применении алгоритма полноволнового сейсмического моделирования. Для решения этой задачи на суперкомпьютере «Ломоносов» было задействовано 1023 узла (16368 ядер). Вычисления заняли 1.5 дня. Для сравнения, на кластере в 16 узлов (96 ядер), подобная задача заняла бы его на 270 дней.

Другим примером использования суперкомпьютера «Ломоносов» можно отнести работу специалистов компании ГЕОЛАБ, которые решая проблему подавления кратных волн-помех, характерную для сейсмических исследований на акваториях провели обработку сейсмических широкоазимутальных данных размером порядка 1 Терабайта, полученных на акватории Мексиканского залива с площади 400 кв. км. с использованием алгоритма 3D SRME. В процессе вычислений производится около 100 миллиардов операций дискретной свертки одномерных массивов, каждый из которых состоит из 2000 элементов. Трафик данных составляет более 1 Петабайта. Оценочное время расчета на одном серверном узле составляет

порядка 1 года. С практической точки зрения это лишено всякого смысла. При использовании 500 узлов суперкомпьютера «Ломоносов» расчет занял 1 день.

Примером применения суперкомпьютера «Ломоносов» к задачам вычислительной гидродинамики, является работа совместной группой мехмата МГУ и ИПМ им. М. В. Келдыша РАН, которой были получены уникальные результаты по численному моделированию формирования и развития концевых вихрей на крыле летательного аппарата при сверхзвуковых режимах. В этой задаче рассматриваются нестационарные течения, содержащие детали различного масштаба, причем расчетная область достаточно велика, чтобы в ней «поместилась» вся головная ударная волна, формирующаяся на крыле летательного аппарата. В окрестности крыла используются ячейки малого размера, чтобы правильно воспроизвести пограничный слой, толщина которого много меньше характерных размеров задачи в целом. На базе этой модели, с необходимо малым шагом по времени наблюдались процессы разного масштаба по времени: высокочастотные колебания на границе вихря и низкочастотные эволюции области взаимодействия в целом.

Одним из существенных преимуществ кластеров перед системами с общей памятью, предопределившим их повсеместное распространение, было и остается построение из стандартных массово-выпускающихся компонент, как аппаратных, так и программных. На сегодняшний момент:

- 75% систем в списке Топ-500 построены на основе процессоров компании Intel,
- чуть больше 13% – на процессорах компании IBM,
- 11% – компании AMD.

Первая десятка лидеров списка Топ-500 построена как на основе процессоров компании Intel, так и на основе процессоров компании IBM. Ярким представителем компании Intel является процессор Intel Xeon E5-2692 на архитектуре Ivy Bridge, используемый в самом мощном суперкомпьютере

мира (согласно, 42 версии рейтинга Топ-500) – Тяньхе-2. Самая мощная архитектура суперкомпьютеров, построенная на базе процессоров IBM, является Blue Gene/Q. Blue Gene/Q является эволюционным продолжением архитектур Blue Gene/L и Blue Gene /P, работающим на более высокой частоте и потребляющей меньше энергии на один FLOPS производительности. BlueGene/Q работает на базе 16-ядерных процессоров IBM PowerPC A2 с частотой 1,6 ГГц. В качестве операционной системы применяется Linux.

Одним из примеров системы, созданной специалистами IBM и базирующейся на архитектуре Blue Gene/Q, является суперкомпьютер Mira, который ввели в эксплуатацию инженеры из Аргоннской национальной лаборатории при Министерстве энергетики США. Согласно ноябрьскому рейтингу Топ-500, в настоящее время Mira является пятыми по производительности суперкомпьютером в мире. Средняя производительность Mira по тесту Linpack составляет 8.586 PFLOPS, пиковая – 10.066 PFLOPS. 48 вычислительных блоков по 1024 узла комплектуются процессорами PowerPC A2 (всего 786432 ядер). Работает суперкомпьютер под управлением операционной системы Linux (RHEL/CentOS). Мощность машины 3945 КВт. С помощью суперкомпьютера Mira исследователи из Аргоннской национальной лаборатории проводят моделирование происходящих во Вселенной процессов. Вычислительные мощности машины позволяют отслеживать взаимодействие триллионов частиц во время процесса расширения вещества. Ученые хотят создать модель ранней Вселенной и наблюдать за ее развитием с момента Большого взрыва и до настоящего времени (13,7 миллиардов лет спустя).

Вскоре, после запуска Mira, Роберт Мозер и его коллеги из Техасского университета запустили один из первых полномасштабных проектов в области прямого численного моделирования гидродинамики. Цель работы Мозера заключается в использовании Mira для прямого численного моделирования при высоких числах Рейнольдса потока жидкости и изучить

сложную физику пристенной турбулентности, для того, чтобы восполнить недостающие знания в этих областях и получить более точные модели турбулентности. В конечном счете, эта работа может привести к снижению сопротивления трубопроводов и воздухопроводов, что сделает их более энергоэффективными.

Следует отметить, что в настоящее время наблюдается непрерывный рост производительности микропроцессоров за счет увеличения количества ядер. Кроме того, стоимость высокопроизводительной вычислительной техники неуклонно снижается, и программно-аппаратные решения, которые еще пару лет назад были исключительно дорогостоящими, сегодня становятся доступными для университетов. Так, СПбГПУ (Санкт-Петербургский государственный политехнический университет) в ближайшее время планирует запустить свой собственный кластер.

С учетом растущей доступности многопроцессорной вычислительной техники возникает возможность получения более подробного и точного решения различных задач и задачи гидродинамики не являются исключением. Благодаря растущим вычислительным ресурсам появляется возможность проводить расчеты гидродинамических задач на максимально детальной сетке, что позволит получить более точные результаты.

Так, одним из физических явлений гидродинамики, к которому проявляется заслуженный интерес, является отрыв потока. Отрыв потока — это одно из физических явлений, возникающих при движении газов или жидкостей над твердой поверхностью или, наоборот, при движении тела в неподвижной жидкости или газе, заключающееся в том, что поток перестает двигаться вдоль поверхности и отходит от нее. Необходимым условием отрыва потока от поверхности является возрастание давления в направлении течения, т. е. положительный (неблагоприятный) градиент давления. Такие условия возникают, например, при обтекании крыла, когда из-за выпуклой формы его верхней поверхности происходит появление неблагоприятного

градиента давления и отрыва потока в задней части профиля, а за точкой отрыва возникает возвратное течение от задней кромки крыла к передней.

Еще одним фундаментальным физическим явлением, возникающим в потоках жидкостей или газов, который может быть рассмотрено во взаимосвязи с отрывом потока является переход течения в пограничном слое из ламинарного состояния в турбулентное. Так, с одной стороны, возникновение отрыва ускоряет переход к турбулентному течению, а с другой стороны, отрыв турбулентного пограничного слоя происходит при более высоком неблагоприятном градиенте давления, чем отрыв ламинарного пограничного слоя.

Таким образом, одной из областей науки, в которой востребованы современные суперкомпьютеры, является вычислительная гидродинамика. Решение практических задач гидродинамики требует использования большого числа ячеек и расчетных сеток больших размеров, что приводит к необходимости использования суперкомпьютерных технологий. В этой связи, одной из наиболее актуальных задач вычислительной аэродинамики является адаптация современных расчетных кодов для работы на кластерах.

Для программной реализации параллельных вычислительных алгоритмов задач гидродинамики применяются различные программные коды. Одним из примеров такого кода в вычислительной гидродинамике является гидродинамический солвер «NTS code» (англ. *Numerical Turbulence Simulation code*), который является CFD-CAA кодом, применимым к широкому кругу течений, включая несжимаемые и сжимаемые, установившиеся и переходные, ламинарные и турбулентные. «NTS code» использует неявную схему конечных объемов высокого порядка на структурированных многоблочных перекрывающихся сетках. Для изучения турбулентности в его арсенале имеется возможность решения стационарных и нестационарных уравнений Рейнольдса с широким спектром моделей турбулентности и отфильтрованных уравнений Навье-Стокса для применения вихреразрешающих подходов.

С помощью «NTS code» было решено большое количество задач, в частности, при решении задачи моделирования шума от тандема цилиндров был использован кластере Intrepid (Argonne National Laboratory) и при ее решении были одновременно задействованы 8192 процессора (32768 ядер).

Развитию возможностей использования газодинамического солвера «NTS code» посвящена настоящая работа.

3. ПРОГРАММА ПОДГОТОВКИ ДАННЫХ ДЛЯ МАССИВНО-ПАРАЛЛЕЛЬНЫХ РАСЧЕТОВ

Гидродинамический солвер «NTS code» является конечно-объемным кодом, использующим структурные многоблочные перекрывающиеся сетки. Это позволяет применять конечно-разностные схемы повышенного порядка, разработанные для структурных сеток аппроксимации, а также обеспечивает возможность проведения расчетов в областях различной степени сложности.

Для массивного распараллеливания задачи необходимо разбить сетку на соответствующее количество подблоков. Другие данные, такие как поля и информация о межблочных интерфейсах, должны соответствовать сетке.

В «NTS code» хранятся следующие файлы данных: файл сетки, файл поля, файл интерфейсов и файл расстояний до стенки. Все файлы хранятся в неформатном виде.

В рамках «NTS code» расчеты интерфейсов и расстояний до стенки не могут быть массивно распараллелены, поэтому эти процессы требуют больших затрат времени. Так, при вычислении расстояний до стенки необходимо найти все точки, лежащие на поверхности, а потом для каждой точки расчетной области перебрать все точки на стенке и найти ближайшую из них. При расчете межблочных интерфейсов для каждой точки межблочного интерфейса необходимо перебрать все ячейки сетки, чтобы найти ту, в которую попадает данная точка.

Для уменьшения этих трудозатрат при постановке и решении задач, был разработан комплекс программ для разбиения данных гидродинамического солвера «NTS code». Данный комплекс программ предназначен для разрезания файлов сетки, поля, расстояний до стенки и интерфейсов на произвольное количество блоков. Необходимо отметить, что данный комплекс программ использует минимальное количество памяти и способен, используя ресурсы персонального компьютера, разбивать практически любую задачу на произвольное количество блоков.

3.1 Описание программы

Комплекс программ для разбиения данных гидродинамического солвера «NTS code» состоит из следующих модулей:

- cutgrid.f – модуль разрезания сетки на блоки;
- cutinter.f – модуль чтение и разрезание интерфейсов на блоки;
- cutpole.f – модуль чтение и разрезание поля на блоки;
- cutlw.f – модуль чтение и разрезание расстояний до стенки;
- grid.f – модуль чтение сетки;
- mail_inc.f – модуль, который содержит все константы необходимые для программы.

Управление работой программы производится через входной файл, который содержит следующую информацию:

- имя файла исходной сетки;
- имя файла сетки, полученной при разрезании;
- имя файла исходных полей;
- имя файла полей, полученных при разрезании;
- имя файла исходных интерфейсов;
- имя файла интерфейсов, полученных при разрезании;
- имя файла исходных расстояний до стенки;
- имя файла расстояний до стенки, полученных при разрезании;
- количество переменных;
- количество блоков в исходной сетке;
- для каждого блока его номер и способ его разбивки.

После прочтения входного файла работу программы можно разбить на следующие пункты:

- чтение файла сетки: количество блоков сетки, размеры блоков сетки, создание массива под каждый блок сетки, определение нового количества блоков и новых границ блоков;
- запись нового количества блоков и их границ в созданный файл сетки;
- чтение файла данных: количество блоков сетки, создание массива под каждый блок;
- запись информации о количестве блоков сетки для нового файла данных;
- чтение файла интерфейсов: создание массивов для каждого блока;
- чтение файла расстояний до стенки: количество блоков сетки, создание массива под каждый блок;
- запись информации о количестве блоков сетки для нового файла расстояний до стенки;
- для каждого блока задачи:
 - чтение всей имеющейся информации о каждом блоке сетки и координат ее точек, запись данных в созданные массивы;
 - запись полученной информации о каждом блоке сетки и координат ее точек в новых границах блоков, запись данных в созданные массивы, а затем в новый файл сетки;
 - чтение всей имеющейся информации о каждом блоке для файла данных, интерфейсов и расстояний до стенки, запись данных в специально созданные массивы;
 - запись полученной информации в новых границах блоков для новых файлов данных, интерфейсов и расстояний до стенки.

3.1.1 Разбиение сетки

Сетка в программе «NTS code» является криволинейной структурированной многоблочной сеткой. Сетка состоит из блоков, которые могут (и должны) перекрываться. При этом внутри области перекрытия блоков каждый блок должен иметь не менее двух линий «в глубину».

Основным понятием при описании блока сетки является понятие линии, которая может быть одного из 3 типов: линия I, линия J и линия K. Каждая линия делится на зоны. Зона имеет тип, начало, конец (целочисленные номера точек) и два граничных условия (в начале и в конце зоны). На границе блоков используются граничные условия межблочного интерфейса.

Разбиение файла сетки происходит в несколько этапов. На первом этапе читается исходная сетка в формате «NTS code» (см. прил.1). На втором этапе на основе данных из входного файла определяются границы блоков сетки, полученной при разрезании, по следующим формулам:

$$NewN1 = N1 - 1 + \frac{(nc - 1) \cdot (N2 - N1 + 1)}{NCut}$$

где $NewN1$ – граница начала нового блока,
 $N1$ – граница начала старого блока,
 $N2$ – граница конца старого блока,
 $NCut$ – количество частей, на которое необходимо разрезать старый блок в данном направлении,
 nc – текущая часть разреза старого блока.

$$NewN2 = N1 + \frac{nc \cdot (N2 - N1 + 1)}{NCut}$$

где $NewN2$ – граница конца нового блока,
 $N1$ – граница начала старого блока,

- $N2$ – граница конца старого блока,
 $NCut$ – количество частей, на которое необходимо разрезать старый блок в данном направлении,
 nc – текущая часть разреза старого блока.

Количество подблоков и способ разбиения задаются во входном файле. Между блоками сохраняется перекрытие в один шаг сетки. На межблочных границах используется условие межблочного интерфейса.

В процессе разбивки формируются данные, соответствующие разрезанной сетке и записываются в файл в соответствующем формате (см. прил. 1). Следует отметить, что для удобства в новых блоках индексы i, j, k сохраняются в новой сетке.

3.1.2 Разбиение файлов данных и расстояний до стенки

Файлы данных и расстояний до стенки имеют одинаковую структуру. Данные для каждого блока являются обычным трехмерным массивом, поэтому их разбиение не составляет большого труда. Соответствующие данные поблочно считываются из файла в массив, а потом этот массив записывается в файл частями, соответствующими подблокам.

3.1.3 Разбиение файла интерфейсов

В файле интерфейсов хранятся интерполяционные коэффициенты для определения значений в граничных точках на межблочных границах.

Разбивка файла интерфейсов происходит в 2 этапа. На первом этапе преобразуются данные исходной сетки, пересчитываются номера блоков для уже имеющихся интерполяционных коэффициентов (поскольку индексы точки i, j, k при разрезании не меняются, остальная информация о старых точках файла интерфейсов сохраняется). На втором этапе происходит поиск

новых интерфейсных точек, и вычисляются интерполяционные коэффициенты для них.

Если исходная задача имела периодические граничные условия и по данному направлению сеточные линии разрезаются, то такие граничные условия заменяются условиями межблочного интерфейса. После создания всей необходимой информации об интерполяционных точках, она записывается согласно формату файла интерфейсов (см. прил.1).

3.2. Тестирование программы

После написания программы разбивки на блоки, на примере ряда небольших задач было проведено тщательное тестирование написанной программы. Тестирование проводилось путем сравнения файлов, полученные двумя способами: с помощью написанной программы и с помощью стандартных средств, предоставляемых программой «NTS code». Полученные файлы были идентичны, что свидетельствует о правильности работы разработанной программы.

3.3. Демонстрация эффективности программы

Эффективность совместной работы «NTS code» и написанной программы была проверена на примере задачи расчета трехмерного крылового профиля при докритическом угле атаки (см. раздел 4). Для этого производилась разбивка на разное количество блоков с использованием разработанной программы. Расчет проводился программой «NTS code» на кластере лаборатории ВГАиТ. Измерялась скорость вычисления одной итерации.

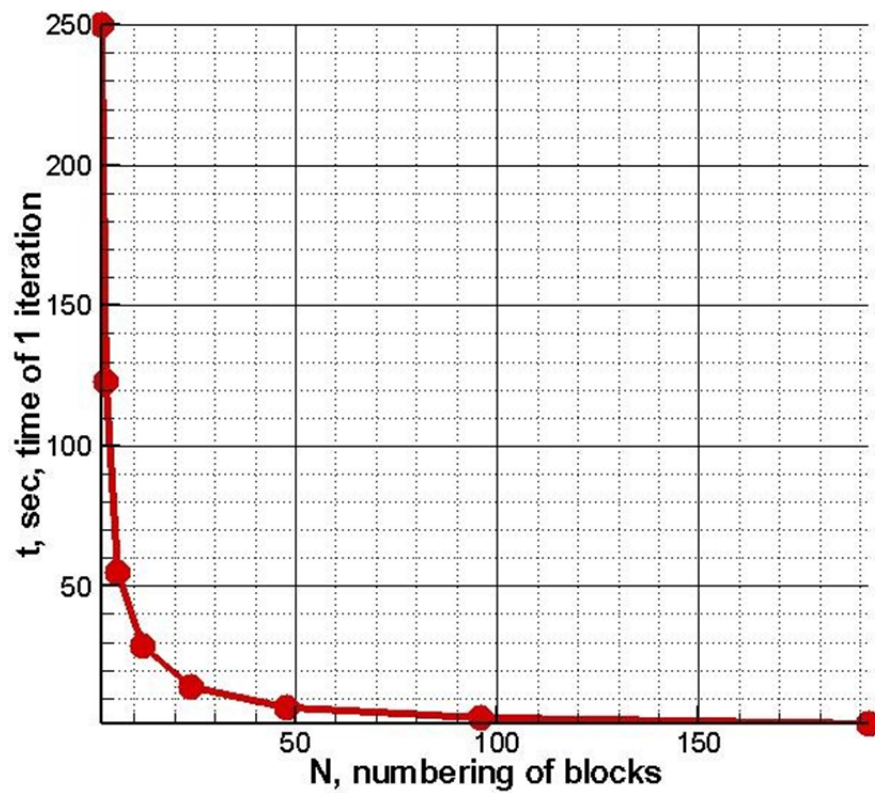


Рис.1. График зависимости времени расчета одной итерации от количества блоков.

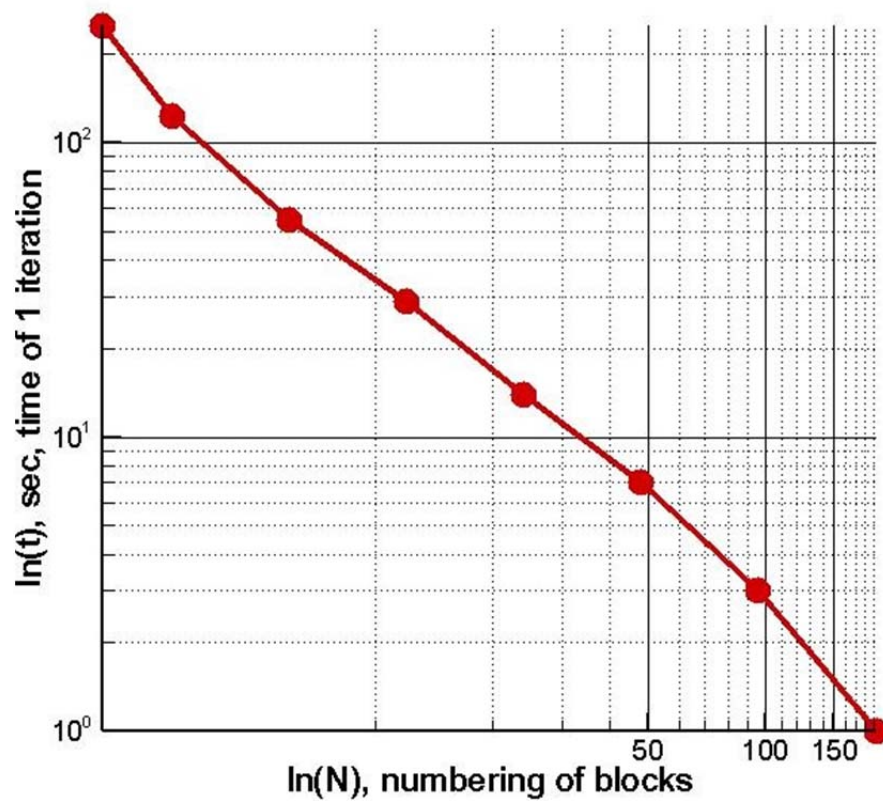


Рис.2. График зависимости времени расчета одной итерации от количества блоков в логарифмических координатах.

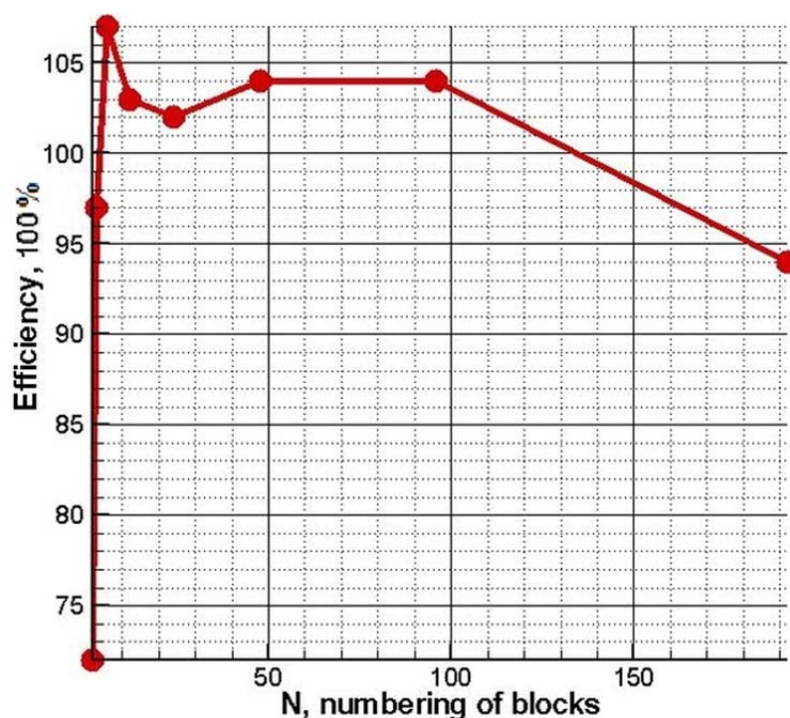


Рис.3. График эффективности.

Полученные результаты приведены на рис. 1-3. Видно, что с увеличением количества подблоков, время расчета одной итерации уменьшается (см. рис.1) и обратно пропорционально количеству используемых узлов (см. рис.2). Столь высокая эффективность параллелизации (см. рис.3) обеспечивается равномерностью разбивки на блоки.

Следует отметить, что, иногда, при увеличении количества подблоков эффективность оказывается несколько выше 100%. Причиной дополнительного ускорения является более эффективное использование кэш-памяти, которая хранит копии наиболее часто используемых участков оперативной памяти. Поскольку время доступа к кэш-памяти в несколько раз меньше, чем к оперативной памяти, а в большинстве случаев необходимая информация находится в кэш-памяти, то кэширование позволяет существенно ускорить работу компьютера. Поэтому при уменьшении размера блока и происходит дополнительное ускорение расчета.

4. МОДЕЛИРОВАНИЕ ОБТЕКАНИЯ КРЫЛОВОГО ПРОФИЛЯ В АЭРОДИНАМИЧЕСКОЙ ТРУБЕ

Изучение отрыва потока при обтекании аэродинамических профилей было и остается задачей многочисленных исследований, которые свидетельствуют о том, что, при определенных режимах течения, структура потока становится трехмерной. Такое вихревое движение, возникающее при отрыве потока, обнаружено в исследованиях различными экспериментальными методами: визуализацией течений с помощью шелковинок, струйками дыма и масляными пленками, нанесенными на поверхность моделей.

По результатам исследований было обнаружено, что в области отрыва существуют крупномасштабные вихри в форме «грибообразных» структур, создающих перетекание потока в поперечном направлении. Обнаружены как общие свойства таких течений, выражающиеся в образовании крупномасштабных парных вихрей, вращающихся в плоскости крыла, так и отличия в топологии вихревых структур в зависимости от типа отрыва, краевых условий, удлинения крыла и других факторов. Область отрыва обладает сложной трехмерной вихревой структурой, в которой доминируют скоррелированные между собой крупномасштабные квазистационарные и нестационарные вихревые образования.

В настоящее время опубликовано ограниченное количество работ, посвященных численному моделированию трехмерной структуры отрыва, возникающего на прямом двумерном крыле, что свидетельствует о недостаточной исследованности данной области. Поэтому работы в данном направлении являются довольно актуальными, что и предопределило выбор задачи для исследования в настоящей работе.

4.1 Постановка задачи

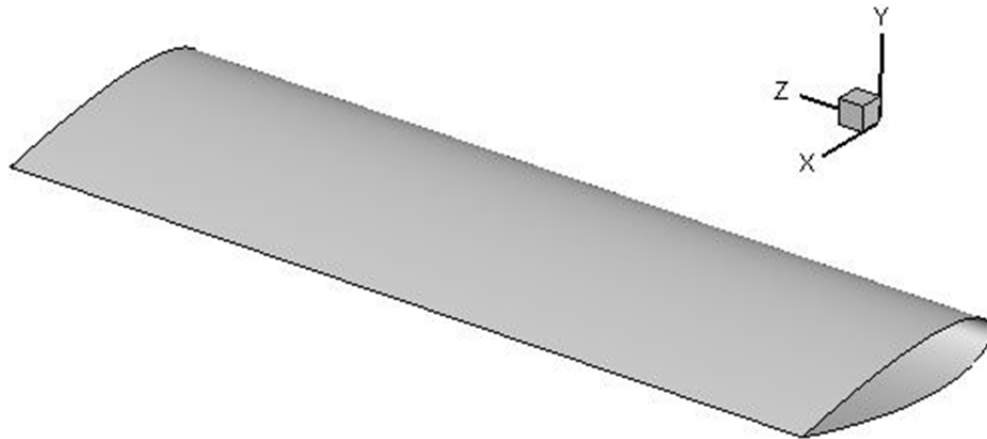


Рис. 4. Профиль NASA 0021.

Рассматривается задача моделирования обтекания трехмерного крылового профиля NASA 0021 в аэродинамической трубе при угле атаки 17° и удлинении крыла $L_z/C = 3$. В задаче использовалась двухблочная сетка с размерами: 1 блок (392x201x102) и 2 блок (101x401x102). На боковых стенках ставились периодические граничные условия, на крыле – условия прилипания, а на свободной границе характеристические граничные условия. В качестве начального приближения было использовано двумерное решение для переменной, распространяющейся по координате z и периодические возмущения компоненты скорости w ($w = 0.1 \cdot \sin(2\pi z/L_z)$).

4.2 Метод решения

Расчет рассматриваемого течения проводился в рамках «NTS code» с использованием стационарных уравнений Рейнольдса, замкнутых с помощью модели Спаларта-Аллмареса. Для получения решения использовался метод установления во времени. При этом производные по времени аппроксимировались с первым порядком точности, и для ускорения сходимости к стационарному решению применялся локальный (в каждой точке расчетной сетки) выбор величины шага интегрирования по времени (итерационного параметра) в соответствии с заданным, единым для всего

поля течения, числом Куранта. Производные от вязких потоков аппроксимировались с помощью симметричных разностей второго порядка точности, а производные от конвективных потоков аппроксимировались с третьим порядком точности для газа и с первым – для турбулентности.

Решение системы разностных аналогов исходных уравнений относительно приращений искомых функций на каждой глобальной итерации осуществлялось с помощью релаксации Гаусса-Зейделя по плоскостям для газодинамических уравнений и с помощью модифицированного (с диагональным преобладанием) метода приближенной факторизации по пространственным направлениям для уравнений переноса турбулентных характеристик (кинетической энергии турбулентности и удельной скорости ее диссипации).

4.3 Результаты расчетов

В результате расчетов было получено сошедшееся стационарное решение, максимальное значение невязок не превышало величины 10^{-5} . Полученное решение содержит две пары «грибообразных» структур (рис. 5-7), что неудивительно, поскольку это количество соответствует начальному возмущению поля. Сравнение поверхностных линий тока, полученных в расчете (рис. 7), с масляными визуализациями, полученными в ходе эксперимента [1] (рис. 8), свидетельствует об их качественном согласии. К сожалению, более детальные количественные сравнения невозможны из-за отсутствия детальных измерений в эксперименте.

Таким образом, проведенный расчет продемонстрировал способность предсказания в рамках уравнений Рейнольдса «грибообразных» структур, наблюдаемых в эксперименте.

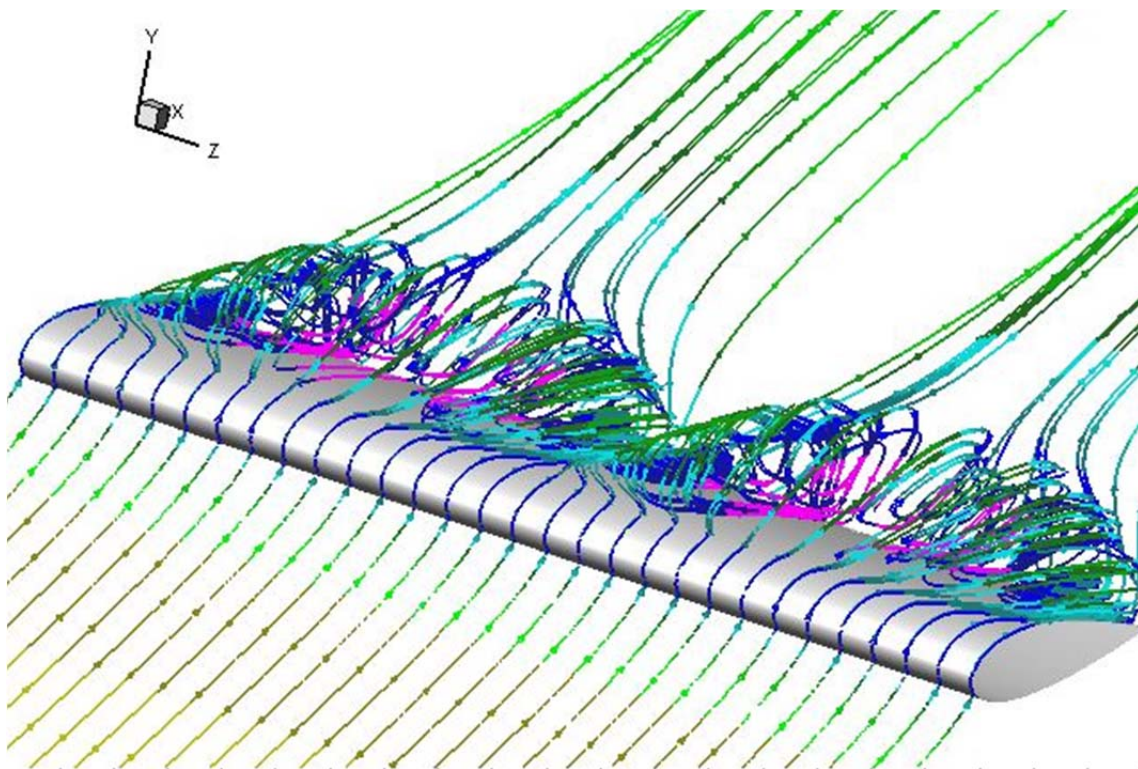


Рис. 5. Трехмерные структуры отрыва

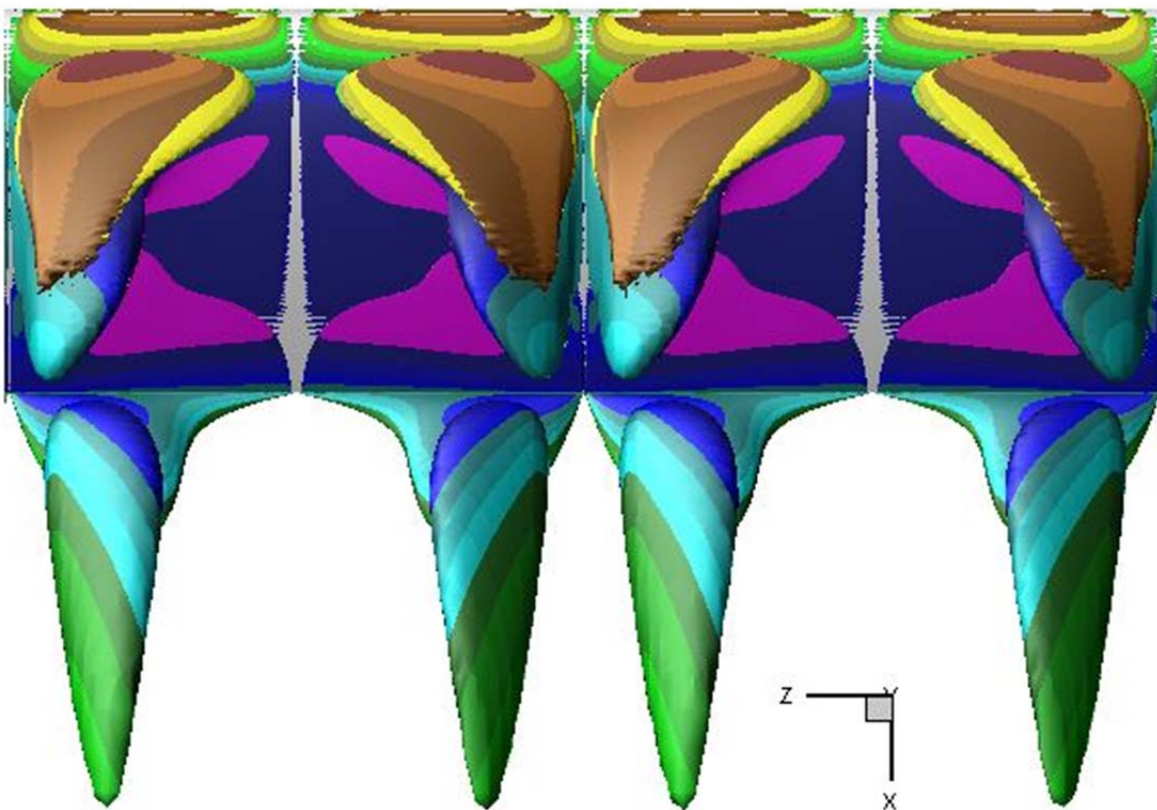


Рис. 6. Проекция завихренности на ось Ox .

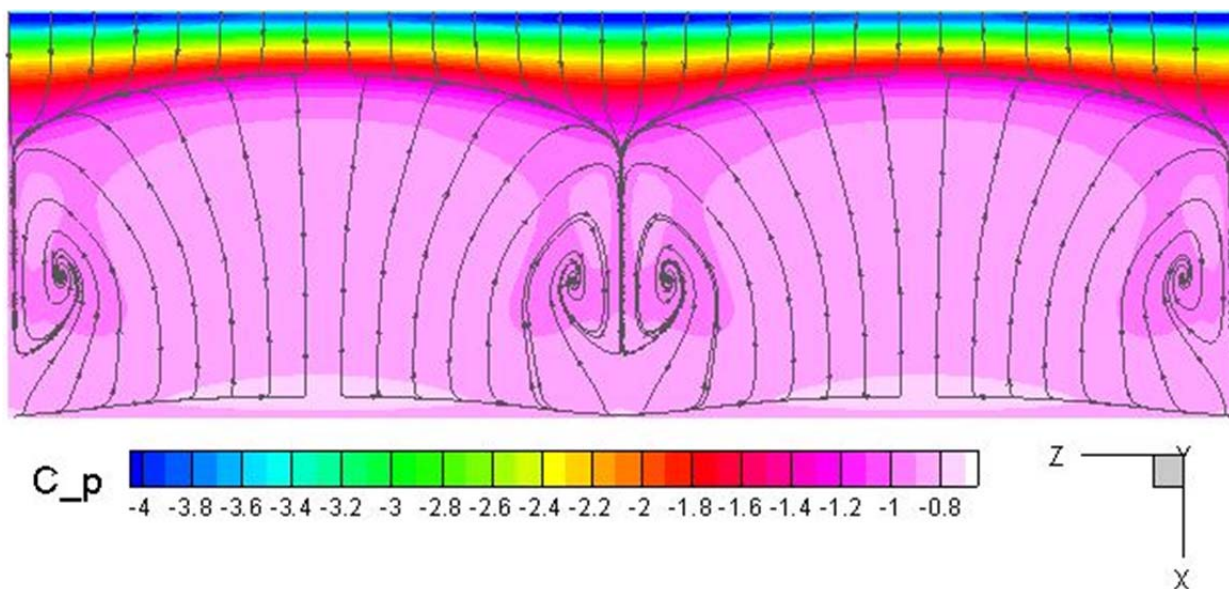
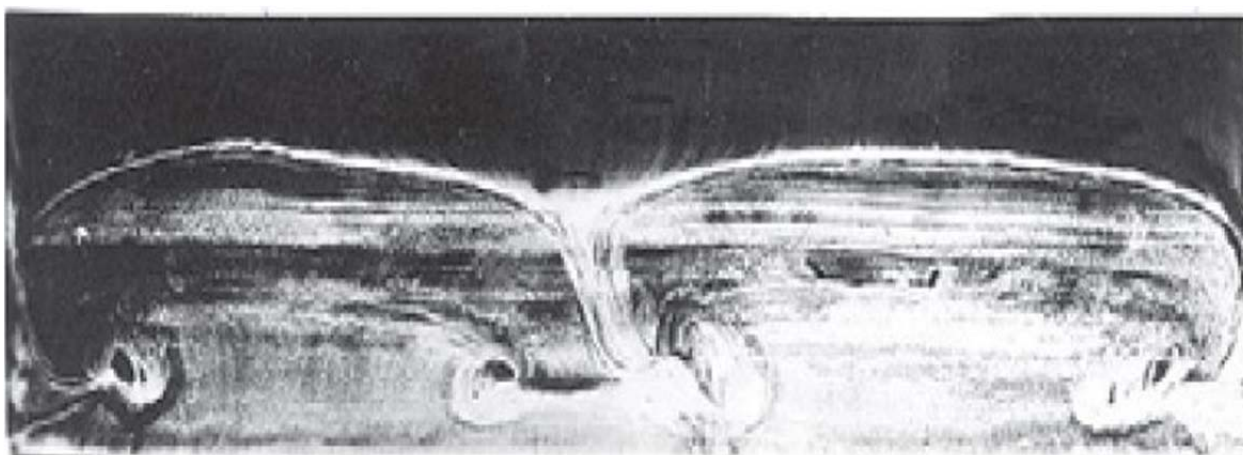


Рис. 7. Отрыв турбулентного пограничного слоя при угле атаки 17°



*Рис. 8. Отрыв турбулентного пограничного слоя при угле атаки $16,8^\circ$
(экспериментальные данные).*

5. ЗАКЛЮЧЕНИЕ

По результатам данной работы можно сделать следующие выводы:

- разработан комплекс программ для разбиения данных газодинамического солвера «NTS code» на произвольное количество блоков;
- проведено всестороннее тестирование разработанного комплекса программ;
- продемонстрирована его эффективность на примере расчета трехмерного крылового профиля при докритическом угле атаки;
- результаты расчетов продемонстрировали возможность предсказания в рамках уравнений Рейнольдса трехмерных «грибообразных» структур, наблюдаемых в эксперименте

6. СПИСОК ЛИТЕРАТУРЫ

1. Katrina Elise Swalwell. The Effect of Turbulence on Stall of Horizontal Axis Wind Turbines. BSc, BEng(Hon), Department of Mechanical Engineering, Monash University. October, 2005, p. 307.
2. Balakrishnan, R., Garbaruk, A., Shur, M.L., Spalart, P., Strelets, M. DDES and IDDES of Tandem Cylinders Flow. Trilateral Russian-French-German workshop "Computational experiment in aeroacoustics", Svetlogorsk, September 22-25, 2010. M.: Maks Press, 2010, p.10-12.
3. Б. Ю. Занин, В. В. Козлов. Вихревые структуры в дозвуковых отрывных течениях: Учеб. пособие / Новосиб. гос. ун-т. Новосибирск, 2011. 116 с.
4. Strelets, M. K. Detached Eddy Simulation of Massively Separated Flows. AIAA Paper, 2001-0879.
5. Kenli Li Zheng, Xiao YanWang, Jiayi Du, Keqin Li (Eds.). Parallel Computational Fluid Dynamics. 25th International Conference, ParCFD 2013 Changsha, China, May 20-24, 2013.
6. М. С. Клинов, С. Ю. Лапшина, П. Н. Телегин, Б. М. Шабанов. «Особенности использования многоядерных процессоров в научных вычислениях», Уфа, 2012, Вестник УГАТУ, Т.16, №6(51), с. 25-31.
7. TOP500 Supercomputer Sites – мировой рейтинг самых мощных суперкомпьютеров мира разных лет: [Электронный ресурс]. 1993-2014. URL: <http://top500.org/timeline/>.
8. TOP500 Supercomputer Sites – мировой рейтинг пятисот самых мощных компьютеров мира за ноябрь 2013г.: [Электронный ресурс]. 1993-2014. URL: <http://top500.org/lists/2013/11/>
9. First Mira Runs Break New Ground with Turbulence Simulations: [Электронный ресурс]. July 16, 2013. URL: <http://www.alcf.anl.gov/articles/first-mira-runs-break-new-ground-turbulence-simulations>.

10. Mira: An Engine for Discovery: [Электронный ресурс]. September 27, 2013. URL: [http:// www.alcf.anl.gov/articles/mira-engine-discovery](http://www.alcf.anl.gov/articles/mira-engine-discovery).
11. Ф. А. Мурзин, С. А. Полетаев. История развития суперкомпьютерной вычислительной техники.
12. В. Воеводин, С. Жуматий, С. Соболев, А. Антонов, П. Брызгалов, Д. Никитенко, К. Стефанов, В. Воеводин. Практика суперкомпьютера «Ломоносов». «Открытые системы», №07, 2012.
13. Суперкомпьютер: понятие, устройство и классификация: [Электронный ресурс]. 23 февраля 2014. URL: <http://pc-vestnik.ru/superkompyuter-ponyatie-ustrojstvo-i-klassifikaciya>.
14. Blue Gene: [Электронный ресурс]. 1998-2014. URL: http://www.nsc.ru/win/elbib/data/show_page.dhtml?77+1328.
15. Кэш-память. Международный компьютерный еженедельник «Computerworld Россия», № 15, 2000.

Приложение 1. Форматы файлов «NTS code»

Файл сетки

read(fw) NumB – количество блоков

Цикл по блокам

Do n = 1, NumB

read(fw) ((ijk(n,j,k),j=1,2),k=1,3) – размеры блоков

EndDo

Цикл по блокам

Do n = 1, NumB - конфигурация

Цикл по линиям: i,j,k

Цикл по плоскости, перпендикулярной линии

read(fw) nz – количество зон

read(fw) (tz, i1, i2, b1, b2, m=1,nz) тип зоны, границы, ГУ

EndDo

Цикл по k,j,i

read(fw) (XYZ(i,j,k), m=1,3) - координаты

Файл интерфейсов

read(fw) NumB – количество блоков

Цикл по блокам

Do n = 1, NumB

read(fw_inter) Inter_Num – количество точек в блоке, для которых нужен интерфейс

Цикл по всем интерфейсным точкам конкретного блока

Do n = 1, Inter_Num

read (fw_inter) номер блока-донора, количество точек в блоке-доноре, используемых для интерфейса, номера точек, веса интерполяции

EndDo

EndDo

Файл данных

Формат полей

Read(fw) NumB – количество блоков

Цикл по блокам

Do n = 1, NumB

read (fw_pole) номер блока, параметр (номер слоя по времени),
количество точек в блоке, ni1, ni2, nj2, nj2, nk1, nk2

Цикл по k,j,i

Read(fw_pole) (XYZ(i,j,k), m=1,3) поля

EndDo

EndDo

Формат расстояний до стенки

Read(fw) NumB – количество блоков

Цикл по блокам

Do n = 1, NumB

read (fw_lw) номер блока, количество точек в блоке, ni1,
ni2, nj2, nj2, nk1, nk2

Цикл по k,j,i

Read(fw_lw) (XYZ(i,j,k), m=1,3) поля

EndDo

EndDo