



ИПМ им.М.В.Келдыша РАН • [Электронная библиотека](#)

[Препринты ИПМ](#) • [Препринт № 78 за 2014 г.](#)



[Луцкий А.Е.](#), [Северин А.В.](#)

GridMan3D - библиотека
подпрограмм для
параллельных вычислений
на нерегулярных сетках

Рекомендуемая форма библиографической ссылки: Луцкий А.Е., Северин А.В. GridMan3D - библиотека подпрограмм для параллельных вычислений на нерегулярных сетках // Препринты ИПМ им. М.В.Келдыша. 2014. № 78. 22 с. URL: <http://library.keldysh.ru/preprint.asp?id=2014-78>

**Ордена Ленина
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
имени М.В.Келдыша
Российской академии наук**

А.Е. Луцкий, А.В. Северин

**GridMan3D – библиотека подпрограмм
для параллельных вычислений
на нерегулярных сетках**

Москва — 2014

Луцкий А.Е, Северин А.В.

GridMan3D – библиотека подпрограмм для параллельных вычислений на нерегулярных сетках

В работе описана библиотека подпрограмм, облегчающая создание программ для решения задач математической физики на нерегулярных сетках. Библиотека включает в себя функции для чтения геометрических данных в формате CGNS и для межпроцессорного обмена данными при помощи интерфейса MPI.

Ключевые слова: параллельные вычисления, нерегулярные сетки

Alexander Evgenjevich Lutsky, Alexander Vladimirovich Severin

GridMan3D – a subroutine library for parallel computing on unstructured grids

A subroutine library that facilitates the creation of programs for solving problems of mathematical physics on unstructured grids is presented. The library includes functions for reading the geometry data in CGNS format and for interprocessor data exchange via an MPI.

Key words: parallel computing, unstructured grids

Работа выполнена при поддержке Российского фонда фундаментальных исследований, проект 14-01-00566.

Оглавление

1. Назначение библиотеки	3
2. Основные возможности библиотеки	4
3. Особенности параллельного счета	7
4. Справочник пользователя.....	8
4.1. Организация цикла.....	8
4.2. Подготовка сетки и начальных данных	10
4.3. Чтение и запись шага вычислений	12
4.4. Распараллеливание.....	13
4.5. Расстояние до поверхности	15
4.6. Параметры файла cgnsqm.h.....	15
4.7. Служебные программы.....	16
5. Пример применения в аэродинамическом расчете.....	17
Литература	22

1. Назначение библиотеки

Создание современных программ для решения задач математической физики сложно не только с точки зрения математики, но и с точки зрения программирования. Такие программы должны иметь возможность работать с регулярными многоблочными и нерегулярными сетками, считывать геометрические данные, созданные на внешнем редакторе, в одном из стандартных форматов и распараллеливать вычисления на несколько процессоров.

Важной задачей является также создание адаптивных алгоритмов [1, 2] для математического моделирования многомасштабных процессов и явлений в механике жидкости и газа. Актуальность исследований в этой области определяется наличием существенно многомасштабных явлений и процессов в многочисленных прикладных задачах. Например, это трансзвуковое ($M=0.8 - 1.1$) обтекание крыльев и головных частей ракет-носителей. В указанном диапазоне формируется внутренняя сверхзвуковая область, которая заканчивается замыкающим скачком (ударной волной). При взаимодействии ударной волны с пограничным слоем могут возникать отрыв и сложная ударно-волновая конфигурация. Положение скачка и весь режим течения могут быть неустойчивыми и нестационарными. При малых изменениях числа Маха или угла атаки структура течения может существенно меняться. В данном случае именно локализация и динамика движения замыкающего скачка являются определяющими факторами, от которых зависят аэродинамические характеристики.

Адаптированная сетка практически всегда оказывается нерегулярной. Определение структуры данных нерегулярной сетки и считывание данных в универсальном формате типа CGNS представляют собой нетривиальные задачи. А распараллеливание – отдельная большая работа, требующая долгой отладки. В связи с этим появилось множество средств, облегчающих математикам выполнение программистской части работы. Пакет GridMan3D – одно из таких средств.

GridMan3D представляет собой библиотеку подпрограмм, дающих доступ к геометрии разностной сетки и полю физических величин, считывающих файл геометрии в формате CGNS, записывающих и читающих данные шага вычислений и осуществляющих обмен данными с другими процессорами. Для синхронизации данных с другими процессорами достаточно вызвать одну подпрограмму в конце каждой итерации и одну – после завершения итераций, таким образом, параллельная программа почти не отличается от однопроцессорной.

2. Основные возможности библиотеки

Библиотека GridMan3D работает с трехмерными разностными сетками, состоящими из ячеек, имеющих форму многогранников следующих типов: тетраэдров, шестигранников, треугольных призм и четырехугольных пирамид (рис. 1).

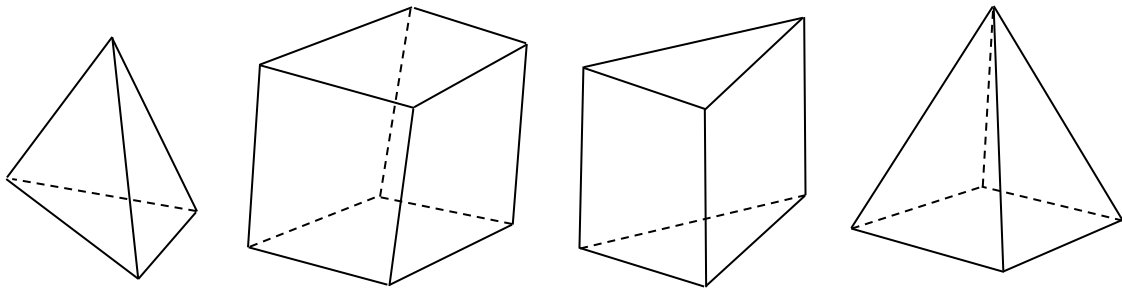


Рис. 1.

Таким образом, это могут быть и многоблочные сетки, состоящие из шестигранников, и тетраэдрические сетки, и смешанные сетки, построенные на основе тетраэдрических.

С точки зрения структуры данных эти сетки всегда нерегулярные, но, поскольку они могут состоять из шестигранников, это могут быть нерегулярные сетки, конвертированные из многоблочных.

Для каждой ячейки определены отношения соседства с другими ячейками и внешними границами.

В каждой ячейке хранится определенное пользователем количество величин, то есть каждой ячейке соответствует одномерный массив. В массиве могут содержаться любые переменные, в зависимости от того, какую программу мы пишем. Это могут быть газодинамические величины – плотность, компоненты скорости, энергия, могут быть их значения в текущий и следующий моменты времени, их производные по пространству и времени. Длина массива может быть любой.

Для организации вычислений используется принцип двух циклов – цикла по соседям внутри цикла по ячейкам. В начале каждого цикла по ячейкам вызывается подпрограмма, возвращающая данные новой ячейки или информацию о том, что ячейки кончились. В начале каждого цикла по соседям вызывается подпрограмма, возвращающая данные нового соседа или информацию о том, что соседи текущей ячейки кончились. После выполнения необходимых вычислений вызываются специальные подпрограммы, заносящие данные в структуру текущей ячейки и ячейки-соседа. Таким образом, структура

данных полностью скрыта от пользовательской программы, вся работа с ними сводится к вызову подпрограмм новой ячейки и нового соседа (рис. 2).

Кроме того, в библиотеке GridMan3D существует возможность инвертировать порядок перебора ячеек, то есть сделать так, чтобы следующий цикл по ячейкам перебирал их в обратном порядке. Это необходимо для некоторых методов решения систем линейных уравнений, например LU-SGS (Lower-Upper Symmetric-Gauss-Seidel – ниже-верхний симметричный Гаусса-Зейделя). [6]

Другая группа подпрограмм библиотеки предназначена для чтения и записи данных. Она включает в себя подпрограмму для чтения файла геометрических данных, который должен быть предварительно сформирован в каком-нибудь редакторе, поддерживающем стандарт CGNS, подпрограммы для установки граничных условий и подпрограммы чтения и записи вычисляемых данных. Вычисляемые данные записываются в специальном формате, но существует конвертер, преобразующий их в формат CGNS, для того чтобы их можно было посмотреть в поддерживающем CGNS постпроцессоре.

Отдельная программа вычисляет расстояния от центра каждой ячейки до ближайшей границы типа "стенка" и записывает их в специальный файл, который может прочитать еще одна подпрограмма библиотеки. Расстояния необходимы для некоторых методов аэродинамики, но их вычисление – очень длительная операция (например, при количестве ячеек 5 млн. она занимает около суток), поэтому рекомендуется выполнять ее только один раз для каждой задачи и в дальнейшем хранить результаты.

Последняя группа подпрограмм связана с организацией параллельных вычислений. Она включает в себя подпрограмму для синхронизации данных между процессорами и подпрограмму согласования итераций.

Подпрограммы библиотеки написаны на языке C и адаптированы для возможности вызова из программы, написанной на языке Fortran.

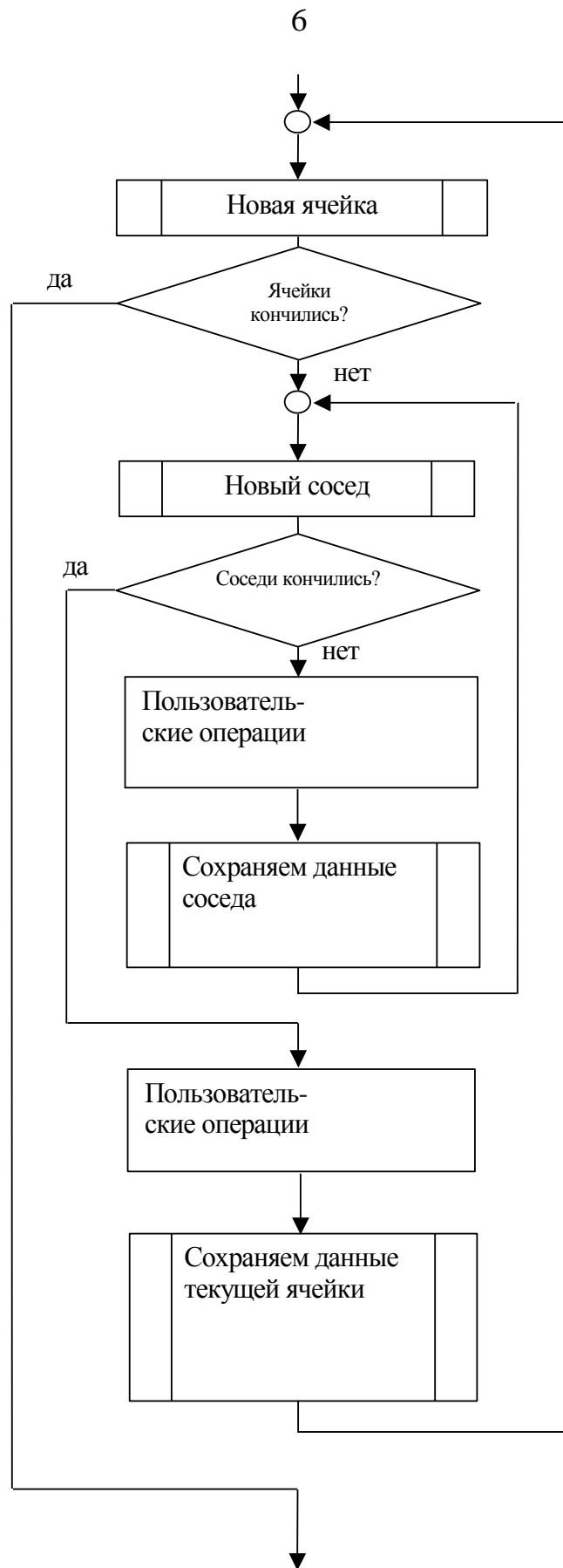


Рис. 2.

3. Особенности параллельного счета

Библиотека GridMan3D адаптирована для многопроцессорных систем с распределенной памятью, использующих интерфейс MPI. Имеется успешный опыт использования библиотеки на многопроцессорных кластерах с UNIX-подобной операционной системой и на персональном компьютере под Windows 7 с установленной Intel MPI Library.

Для решения задач, требующих большей памяти, чем память одного процессора, данные частично распределяются по процессорам, то есть распределяются только вычисляемые данные, геометрические – дублируются. Последний фактор накладывает некоторое ограничение, но, так как обычно геометрические координаты составляют лишь небольшую часть данных, на практике это ограничение несущественно.

Данные ячеек, обрабатываемых другими процессорами, представлены фиктивными ячейками. Фиктивные ячейки исключены из последовательности перебора текущих ячеек, но присутствуют в цикле по соседям. Специальная подпрограмма синхронизирует данные, то есть переписывает в фиктивные ячейки данные с соответствующих им реальных ячеек на соседних процессорах. Подпрограмма синхронизации может переписать все переменные или только часть их. На разных этапах вычислений можно синхронизировать разные группы переменных.

В явном методе достаточно синхронизировать данные для текущего момента времени в начале шага. В неявном необходимо также синхронизировать данные следующего момента времени после каждой итерации. Если порядок аппроксимации по пространству выше первого, нужно синхронизировать производные.

В конце итерации вызывается подпрограмма, проверяющая критерий завершения итераций на других процессорах. Как и все параллельные операции, она максимально упрощена: у нее всего два параметра.

4. Справочник пользователя

4.1. Организация цикла

CALL RESCELL (R, F, V, XYZ)

Выходные параметры:

INTEGER R – 0 при нормальном завершении; -1, если в сетке нет ни одной ячейки;

DOUBLE PRECISION F (NCTOT) – массив переменных из первой ячейки, **NCTOT** – общее число переменных в ячейке (см. описание параметров файла cgnsqm.h);

V – объем ячейки;

DOUBLE PRECISION XYZ (3) – координаты центра ячейки.

Приводит структуру данных в начальное состояние, устанавливает прямое направление перебора и возвращает данные из первой ячейки.

CALL REVCELL (R, F, V, XYZ)

Выходные параметры:

INTEGER R – 0 при нормальном завершении; -1, если в сетке нет ни одной ячейки;

DOUBLE PRECISION F (NCTOT) – массив переменных из первой ячейки, **NCTOT** – общее число переменных в ячейке (см. описание параметров файла cgnsqm.h);

V – объем ячейки;

DOUBLE PRECISION XYZ (3) – координаты центра ячейки.

Приводит структуру данных в начальное состояние, устанавливает обратное направление перебора и возвращает данные из последней ячейки.

CALL GETNEWCELL (R, F, V, XYZ)

Выходные параметры:

INTEGER R – 0 при нормальном завершении; -1, если ячейки кончились;

DOUBLE PRECISION F(NCTOT) – массив переменных из первой ячейки, **NCTOT** – общее число переменных в ячейке (см. описание параметров файла `cgnsqm.h`);

V – объем ячейки;

DOUBLE PRECISION XYZ(3) – координаты центра ячейки.

Возвращает данные из новой ячейки и делает ее текущей, обнуляет счетчик соседей текущей ячейки.

CALL GETNEWCELLF(R, F, V, XYZ)

Аналогична подпрограмме **GETNEWCELL**, но перебираются все ячейки, включая фиктивные.

CALL GETNEWNEIG(R, F, V, XYZ, XYZN, XYZS)

Выходные параметры:

INTEGER R – 0, если соседняя ячейка еще не встречалась в качестве текущей; 1 – если встречалась; -1, если соседние ячейки закончились; тип границы, если вместо ячейки попала граница. Типы границ определяются в файле `cgnsqm.h`, стандартные типы: 3 – стенка, 4 – вдув, 5 – экстраполяция, 6 – плоскость симметрии, 7 – характеристическая, 8 – пристеночные функции.

DOUBLE PRECISION F(NCTOT) – массив переменных соседней ячейки, **NCTOT** – общее число переменных в ячейке (см. описание параметров файла `cgnsqm.h`);

V – объем соседней ячейки;

DOUBLE PRECISION XYZ(3) – координаты центра соседней ячейки;

DOUBLE PRECISION XYZN(3) – вектор нормали к грани между ячейками, модуль вектора равен площади грани;

DOUBLE PRECISION XYZS(3) – координаты центра грани между ячейками.

Возвращает данные из соседней ячейки.

CALL SETCURVAL(F)

Входной параметр:

DOUBLE PRECISION F (NCTOT) – массив счетных переменных.

Заносит массив F в текущую ячейку.

CALL SETNEIVAL (F)

DOUBLE PRECISION F (NCTOT) – массив счетных переменных.

Заносит массив F в соседнюю ячейку.

4.2. Подготовка сетки и начальных данных

CALL READCGNSGEOM (R)

Выходной параметр:

INTEGER R – 0, если файл считался нормально, или номер ошибки чтения.

Читает геометрию сетки и типы граничных условий из файла `geom.cgns`, который должен находиться в рабочей директории. После считывания сетка еще не готова к работе, необходимо считать соседство подпрограммой `READNEIGHB`.

В этой подпрограмме происходит инициализация параллельного счета, до ее вызова программа работает как один поток вычислений, после – как несколько независимых потоков на разных процессорах. Поэтому все чтения и записи файлов, общие для всех потоков, рекомендуется завершить до вызова этой подпрограммы, в противном случае это может привести к конфликту за доступ к файлу между потоками.

CALL READNEIGHB (R)

Выходной параметр:

INTEGER R – 0, если файл считался нормально, или номер ошибки чтения.

Считывает ссылки на соседей из файла `neighb.dat`, который должен находиться в рабочей директории.

Файл `neighb.dat` содержит информацию об отношениях соседства между ячейками и должен быть сформирован перед запуском основной задачи при помощи программы `edge.exe` на основе данных `geom.cgns`.

CALL SETBNDCOND (F)

Входной параметр:

DOUBLE PRECISION F (NCTOT) – массив расчетных величин, **NCTOT** – общее число переменных в ячейке (см. описание параметров файла `cgnsigm.h`).

Заносит массив `F` во все границы, кроме определенных подпрограммой **SETEXBNDGLOB**. После этого массив `F` будет возвращаться функцией **GETNEWNEIG**, когда граница попадет в качестве соседа. В газодинамике обычно используется на границе типа «вдвух» для хранения параметров набегающего потока.

CALL SETINITDATA (F)

Входной параметр:

DOUBLE PRECISION F (NCTOT) – массив расчетных величин

Заносит массив `F` во все ячейки сетки.

CALL SETEXBNDGLOB (MARK, FE)

Входные параметры:

CHARACTER MARK (32) – метка границы;

DOUBLE PRECISION FE (NCBASE) – массив расчетных величин, **NCBASE** – количество базовых переменных (см. описание параметров файла `cgnsigm.h`).

Добавляет элемент в список дополнительных граничных условий. Это необходимо, когда нужно задать разные граничные условия на границах одного типа, например, вдвухвание двух разных потоков через два отверстия. Признаком,

по которому граница узнается, является буквенная метка **MARK**. Она должна совпадать с несколькими последними байтами поля «bosoname» в файле CGNS. В CGNS-файле, сгенерированном ANSYS, bosoname получается из имени детали (part) добавлением префикса.

Сама по себе подпрограмма не устанавливает граничных условий, она только запоминает **MARK** и **FE** в глобальных переменных. Установление граничных условий происходит при следующем вызове **SETBNDCOND**, тогда массив **FE** заносится в отмеченные границы вместо массива **F**. Эта подпрограмма должна быть вызвана до чтения файла геометрии, тогда как **SETBNDCOND** вызывается после него.

4.3. Чтение и запись шага вычислений

CALL WRITESTEP (R, NAME, NS, TIME, PREC)

Входные параметры:

CHARACTER *n NAME – имя задачи (вместо n – любое целое число);

INTEGER NS – номер шага;

DOUBLE PRECISION TIME – время;

INTEGER PREC – точность: 4 – если данные записываются с одинарной точностью или 8 – если с двойной.

Выходной параметр:

INTEGER R – 0, если файл записался нормально, или номер ошибки записи.

Записывает данные в файл. Имя файла формируется следующим образом: вначале идет имя задачи, потом номер шага, потом точка и расширение bin. В номере шага всегда 8 знаков, высшие разряды заполняются нулями. То есть имя файла с шагом 10 будет выглядеть так: test00000010.bin

Формат начала файла выглядит следующим образом:

<номер шага, 4 байта><время, 8 байт><точность, 1 байт><число ячеек, 4 байта>

Далее идут данные первых **NCBASE** переменных (см. описание параметров файла cgnsgm.h), рассортированные по типам, а не по ячейкам, то

есть сначала идет элемент **F(1)** из всех ячеек, потом элемент **F(2)** из всех ячеек и т. д.

Для преобразования файла шага в вид, пригодный для просмотра, существует программа `convert.exe` (см. соответствующий раздел справочника). Она конвертирует файлы геометрии и шага в один файл в формате CGNS, содержащий и геометрию, и данные. Такой файл можно просматривать в различных программах визуализации научных данных, например, в Tecplot.

CALL READSTEP (R, NAME, NS, TIME)

Входной параметр:

CHARACTER *n NAME – имя задачи (вместо *n* – любое целое число).

Выходные параметры:

INTEGER R – 0, если файл прочитался нормально, или номер ошибки чтения;

INTEGER NS – номер шага;

DOUBLE PRECISION TIME – время.

Читает данные из файла. Точность считывается из файла, при считывании данные преобразуются в тот тип, который задан в include-файле `cgnsqm.h` параметром **PRECISION**.

4.4. Распараллеливание

CALL PROCDATEX (EXDATBEGIN, EXDATEND)

Входные параметры:

INTEGER EXDATBEGIN – начало пересылаемой группы переменных, т.е. первая пересылаемая переменная при нумерации с нуля.

INTEGER EXDATEND – конец пересылаемой группы переменных, т. е. число, на единицу больше номера последней пересылаемой переменной при нумерации с нуля.

Осуществляет обмен данными между процессорами, то есть заносит в фиктивные ячейки данные из соответствующих им реальных ячеек на других

процессорах. Может вызываться несколько раз в процессе каждой итерации для пересылки разных групп переменных. Для правильной работы подпрограммы в файле `cgnsqm.h` должен быть задан параметр **EXLENGTH** – максимальное количество пересылаемых переменных, т.е. наибольшая из разностей **EXLENGTH=EXDATEND-EXDATBEGIN**.

CALL NEWITERATION (KI , KO)

Входной параметр:

INTEGER KI – входной флаг.

Выходной параметр:

INTEGER KO – выходной флаг.

KO=0, если **KI=0** на всех процессорах. Если хотя бы на одном процессоре

KI=1, то **KO=1**. Используется для согласования критерия завершения итераций на нескольких процессорах. **KI** – флаг продолжения итераций на данном процессоре, т.е. если итерации надо продолжать, то **KI=1**. Внутри подпрограммы происходит обмен сообщениями с другими процессорами, и если хотя бы один должен продолжать итерации, на выходе будет 1.

CALL SETNEXTTIMEDATA

Копирует первые **NCBASE** переменных в следующие **NCBASE**. Используется, когда надо переписать данные текущего шага по времени в следующий. Делает это быстрее, чем пользовательский цикл, и копирует данные не только в реальных ячейках, но и в фиктивных. В параллельных вычислениях вызов этой подпрограммы может сэкономить часть обменов данными.

CALL SETCURRENTTIMEDATA

Копирует на место первых **NCBASE** переменных следующие **NCBASE**. Используется, когда надо переписать данные следующего шага по времени в текущий. Делает это быстрее, чем пользовательский цикл, и копирует данные

не только в реальных ячейках, но и в фиктивных. В параллельных вычислениях вызов этой подпрограммы может сэкономить часть обменов данными.

CALL MPIFIN

Завершает работу MPI. Должна быть вызвана в самом конце.

4.5. Расстояние до поверхности

CALL READDIST (R)

Выходной параметр:

INTEGER R – 0, если файл прочитался нормально, или номер ошибки чтения.

Считывает расстояние до поверхности из файла `distance.dat`, сформированного программой `dist.exe`, который должен находиться в рабочей директории, и заносит его в последний элемент массива расчетных величин.

4.6. Параметры файла `cgnsqm.h`

#define PRECISION 4

Одинарная или двойная точность. Если 4, то данные хранятся в виде чисел одинарной точности и ковертируются в двойную точность при передаче в пользовательскую часть программы. Если 8, то данные хранятся с двойной точностью.

#define WIN_OR_UNIX 1

Следует установить: 1 – для операционной системы семейства Windows, 0 – для UNIX-подобной операционной системы. Связано с особенностями передачи символьных переменных из Fortran в C.

#define NCBASE 5

Число базовых переменных. Первые **NCBASE** переменных записываются и читаются подпрограммами чтения-записи шагов вычислений. Кроме того, это число фигурирует в подпрограммах копирования данных **SETNEXTTIMEDATA** и **SETCURRENTTIMEDATA**.

```
#define NCTOT 47
```

Общее число переменных.

```
#define WallGM 3
#define InflowGM 4
#define MirrorGM 6
#define OutflowGM 7
#define ExtrapolateGM 5
#define WallViscousGM 8
```

Типы граничных условий.

```
#define IBW 10
#define IBE 11
#define IBI 12
```

Номера переменных, которые используются в качестве буферов межпроцессорных обменов при чтении и записи. Введены для экономии памяти, чтобы не захватывать память под эти буферы специально. Следует выбрать для этого такие переменные, которые во время чтения и записи не нужны.

```
#define EXLENGTH 15
```

Максимальное число переменных, пересылаемое подпрограммой **PROCDATEX** за один раз.

4.7. Служебные программы

edges.exe

Устанавливает соседство между ячейками и изменяет последовательность узлов в ячейке в соответствии с принятыми в данной библиотеке

соглашениями. Во время запуска программы в рабочей директории должен находиться файл `geom.cgns`. Результатом работы являются файлы `neighb.dat` и `nodes.dat`, которые необходимы для запуска основной программы.

`prostar.exe [n] [m] [l]`

Распределяет ячейки по процессорам. Разбиение производится по геометрическому критерию, то есть к одному процессору относятся ячейки, центры которых попадают в один и тот же интервал геометрических координат. **n**, **m**, **l** – число интервалов по каждой из координат *x*, *y*, *z* соответственно. То есть общее число процессоров равно **n*m*l**. Во время запуска программы в рабочей директории должен находиться файл `geom.cgns`. Результатом работы является файл `prostar.dat`, который необходим для запуска основной программы.

`dist.exe`

Вычисляет расстояние от центра каждой ячейки до ближайшей стенки. Нужно для некоторых методов вычислительной аэродинамики. Довольно длительная операция, для сетки размером в 5 миллионов ячеек занимает около суток. Во время работы выводит процент обработанных ячеек. Во время запуска программы в рабочей директории должен находиться файл `geom.cgns`. Результатом работы является файл `distance.dat`, который необходим для запуска, если ваша программа использует подпрограмму **READDIST**.

`convert.exe [name]`

Преобразует файл шага вычислений в файл формата CGNS, который можно посмотреть в программе визуализации научных данных, например, Tecplot. **name** – имя преобразуемого файла. Во время запуска программы в рабочей директории должен находиться файл `geom.cgns`.

5. Пример применения в аэродинамическом расчете

С использованием библиотеки GridMan3D создан трехмерный солвер для решения уравнений Рейнольдса методом конечных объемов. Интегрирование по времени осуществляется при помощи неявного метода LU-SGS. [6] Аппроксимация по пространству второго порядка. Потоки вещества, импульса

и энергии через грани вычисляются по методу Годунова. Турбулентная вязкость моделируется по формуле Смагоринского. [7] Длина пути смещения определяется в пристеночной области по расстоянию до стенки, а на удалении от стенки – по характерному размеру ячейки.

При помощи данного солвера выполнен тестовый расчет стандартного крыла ONERAM6. Число Маха набегающего потока $M=0.8395$, число Рейнольдса $Re=11.72 \cdot 10^6$, угол атаки $\alpha=3.06^\circ$.

Вокруг крыла была построена многоблочная сетка из 870880 ячеек (рис. 3 и 4). Наименьшая толщина ячейки вблизи поверхности тела около 0.3 мм.

На поверхности крыла было установлено граничное условие типа пристеночных функций (wall-functions) [8], на передней и боковых границах счетной области – условие вдува, в плоскости основания крыла – условие симметрии, на задней границе – характеристическое условие вытекания.

Расчеты выполнялись на многопроцессорном суперкомпьютере с распределенной памятью «Чебышёв» суперкомпьютерного комплекса ВЦ МГУ и на персональном компьютере с 8-ядерным процессором фирмы AMD. Использовалось от 1 до 64 процессоров на суперкомпьютере «Чебышёв» и от 1 до 8 ядер на персональном компьютере. Достигнута обычная для такого рода задач эффективность распараллеливания (табл. 1).

Таблица 1.

Число ядер		2	4	8	16	32	64
Эффективность распараллеливания, %	«Чебышёв»	101.3	98.9	76.9	76.1	56	40
	РС	95.6	97.1	93.2			

Аномалии зависимости эффективности распараллеливания от числа процессоров, в частности, более чем стопроцентная эффективность для двух процессоров на «Чебышёве», небольшое снижение для 16 по сравнению с 8 там же и более высокая эффективность распараллеливания для 4, чем для 2, на персональном компьютере, могут быть объяснены особенностями доступа к памяти.

В результате расчета получена физически адекватная картина течения, воспроизводящая все его основные особенности, включая турбулентный пограничный слой, зону сверхзвукового течения и ударную волну на выходе из сверхзвуковой зоны (Рис. 5-7).

Результаты расчета хорошо согласуются с экспериментом. [5]

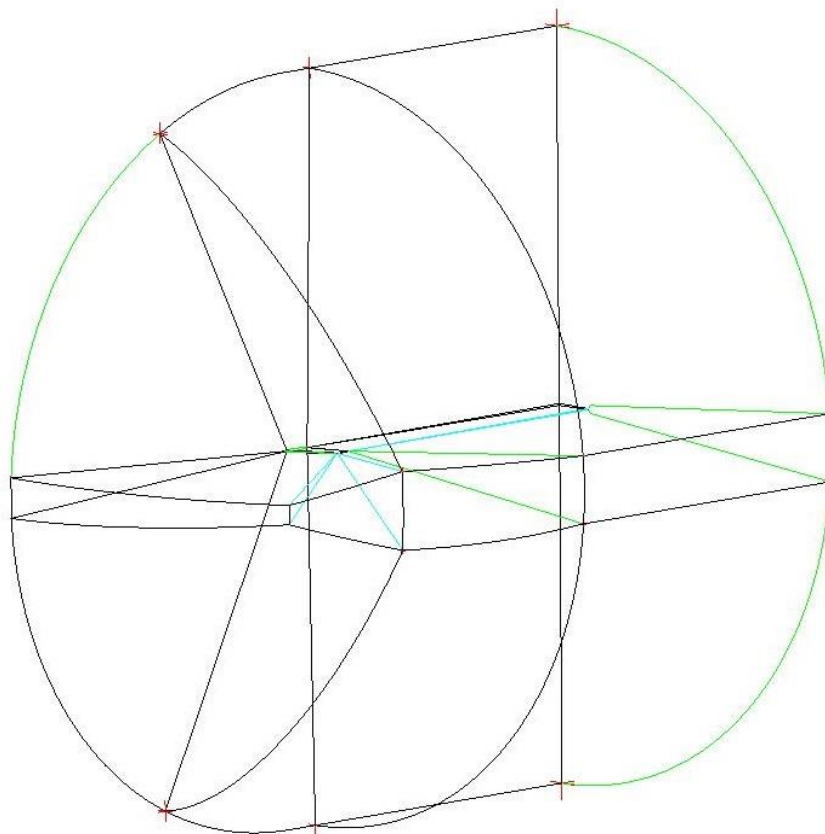


Рис. 3. Блочная структура сетки вокруг крыла ONERAM6.

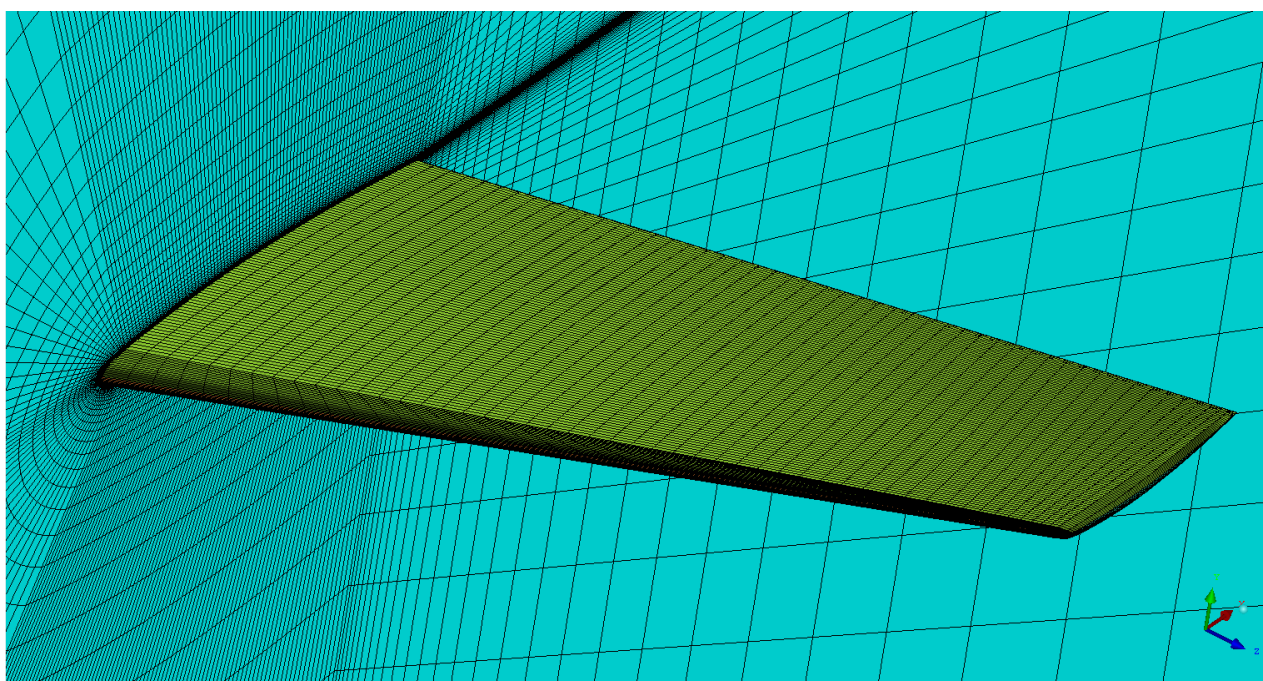


Рис. 4. Сетка вблизи крыла ONERAM6.

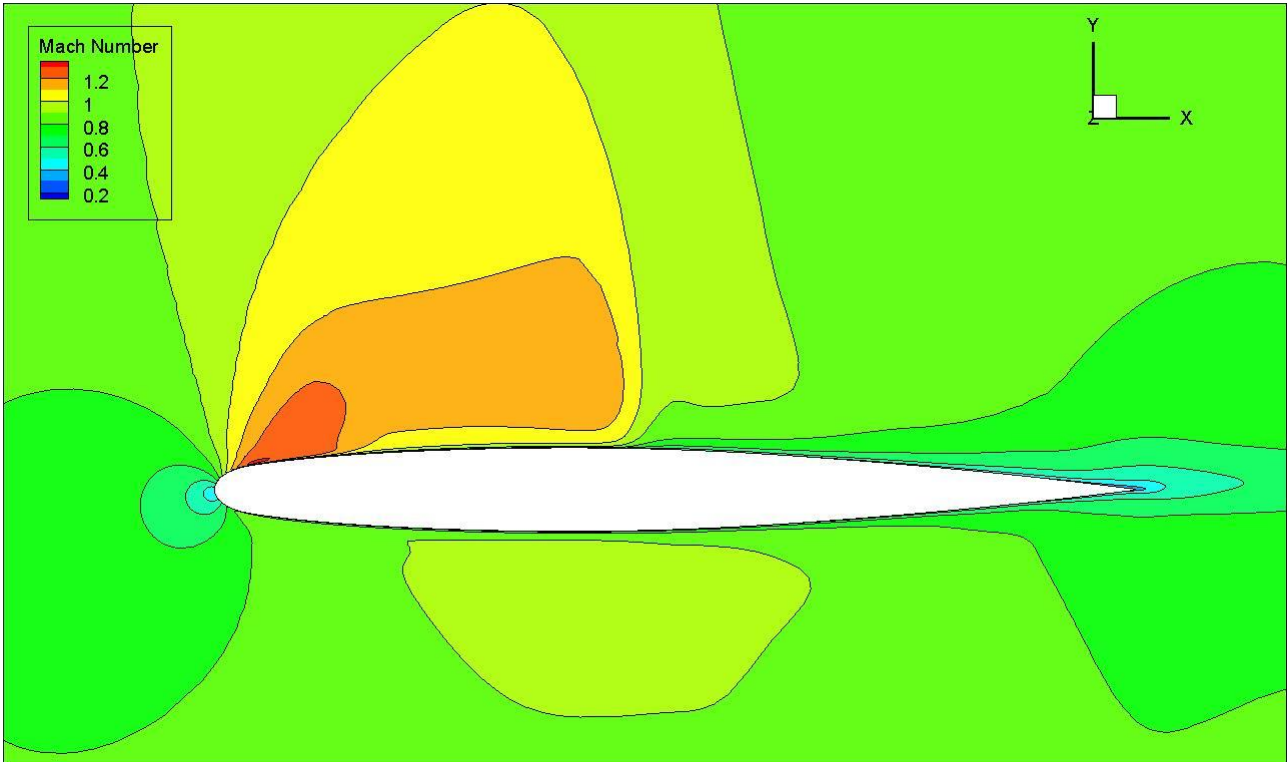


Рис. 5. Число Маха вокруг крыла ONERAM6, сечение $Z=0.65b$.

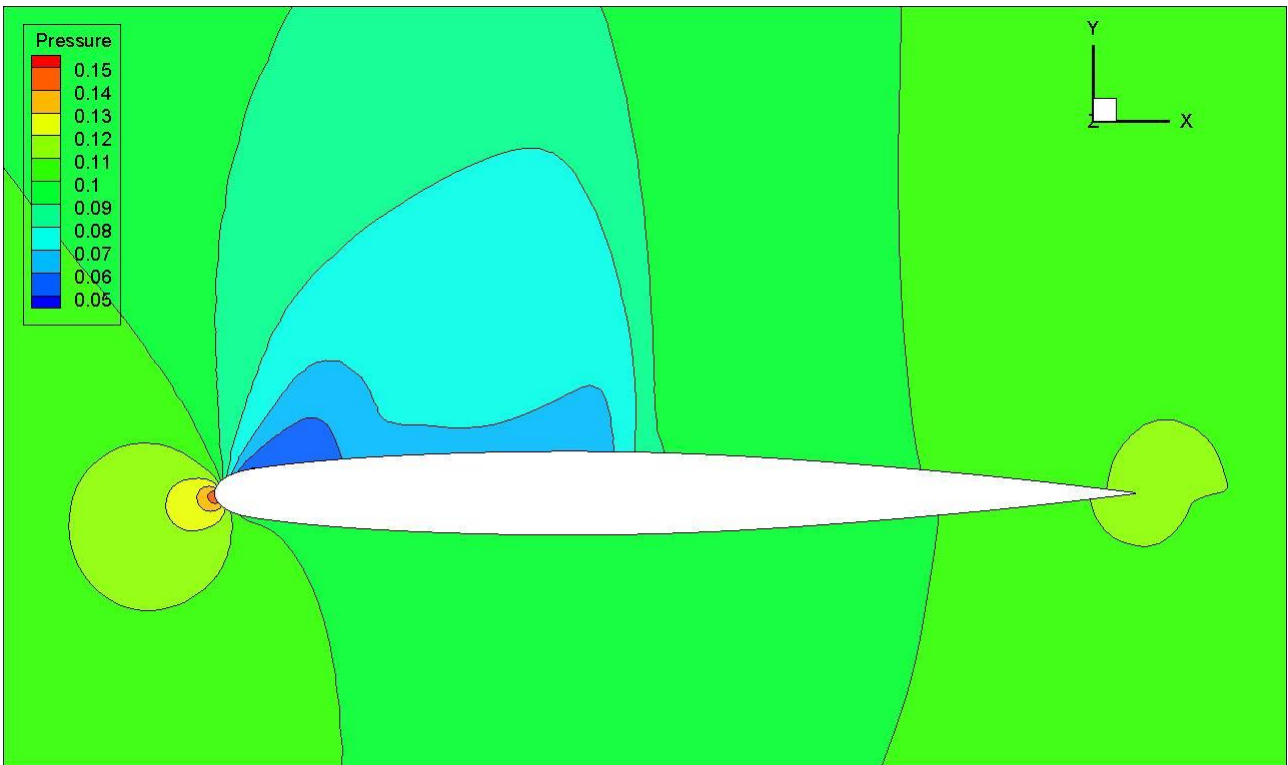


Рис. 6. Давление (МПа) вокруг крыла ONERAM6, сечение $Z=0.65b$.

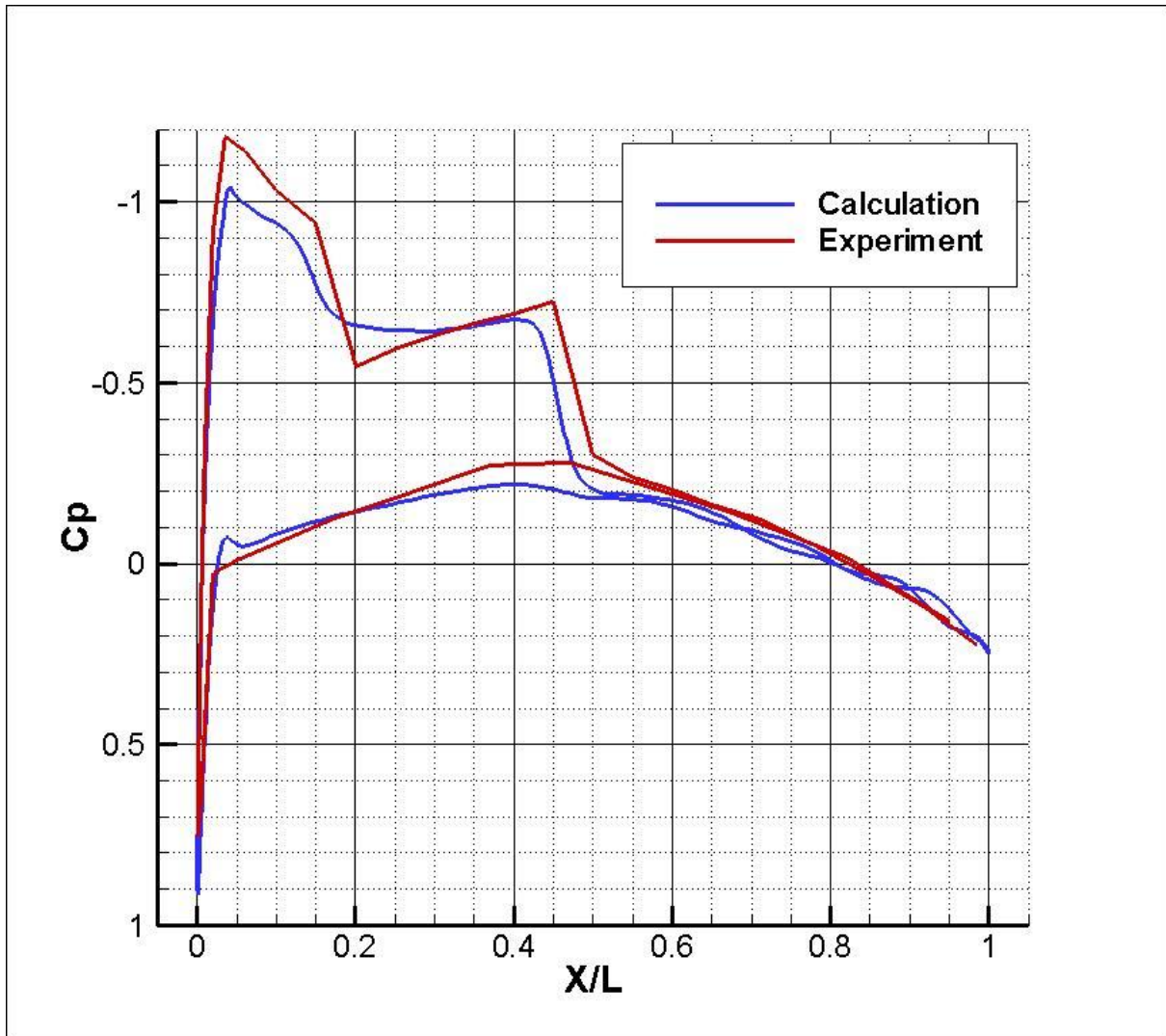


Рис. 7. Коэффициент давления на поверхности крыла ONERAM6, сечение $Z=0.65b$.

Литература

1. А.Л. Афонди́ков, А.Е. Луцкий, А.В. Плёнкин, Вейвлетный анализ локализованных структур в идеальной и вязкой моделях. // *Матем. моделирование*, **23**:1 (2011), С. 41–50.
2. А.Л. Афонди́ков, А.Е. Луцкий, А.В. Плёнкин, Локализация особенностей газодинамических полей и адаптация расчетной сетки к положению разрывов. // *Матем. моделирование*, **24**:12 (2012), С. 49–54.
3. Официальная страница стандарта MPI. Режим доступа: URL: <http://www.mpi-forum.org>
4. Официальная страница стандарта CGNS. Режим доступа: URL: <http://cgns.sourceforge.net>
5. Геометрия и результаты продувки крыла ONERAM6 на сайте NASA. Режим доступа: URL: <http://www.grc.nasa.gov/WWW/wind/valid/m6wing/m6wing.html>
6. I. Men'shov, Y. Nakamura, Hybrid Explicit–Implicit, Unconditionally Stable Scheme for Unsteady Compressible Flows // *AIAA Journal*, Vol. 42, No. 3, pp. 551-559, 2004.
7. Joseph Smagorinsky. General Circulation Experiments with the Primitive Equations. *Monthly Weather Review*, 1963. Vol. 91, pp. 99-164.
8. T. Knopp. On grid-independence of RANS predictions for aerodynamic flows using model-consistent universal wall-functions // *European Conference on Computational Fluid Dynamics ECCOMAS CFD*, 2006.