

Исследование эффективности распараллеливания расчета движения подвижных тел и свободных поверхностей в FlowVision на компьютерах с распределенной памятью

А.А.Аксенов, А.А.Дядькин, С.А.Харченко

Современные задачи вычислительной гидродинамики часто включают необходимость моделирования нестационарного движения жидкости в расчетных областях, имеющих подвижные границы, а также моделирование контактных границ раздела фаз (свободная поверхность). В программном комплексе FlowVision используется наиболее универсальный метод расчета таких границ с использованием эйлеровых расчетных сеток. При этом подходе движение твердых границ и свободных поверхностей во времени приводит к образованию и исчезновению новых расчетных ячеек, причем общее количество ячеек может существенно меняться. В программном комплексе FlowVision используется автоматическая динамическая адаптация расчетной сетки в процессе решения, что также приводит к изменению как топологической структуры сетки так и ее размера. По этим причинам возникает задача о балансировке параллельных вычислений для расчетной сетки, последовательно меняющейся в процессе численного моделирования. Статическая декомпозиция задачи по процессорам (т.е. один раз в начале расчета) может приводить к нарушению межпроцессорного баланса вычислений и потере параллельной эффективности. С другой стороны, полное перевычисление декомпозиции может приводить к большой межпроцессорной миграции данных, поэтому проведение таких вычислений на каждом временном шаге вычислительного алгоритма недопустимо дорого. В данной работе описывается подход, позволяющий эффективно обойти основные трудности параллельной реализации технологий подвижных тел и свободных поверхностей. Приводятся результаты численных экспериментов по масштабируемости предложенных алгоритмов на примере моделирования задачи вычислительной гидродинамики при помощи FlowVision.

1. Введение

Современные задачи вычислительной гидродинамики часто включают необходимость моделирования нестационарного движения жидкости в расчетных областях, имеющих подвижные границы, а также моделирование контактных границ раздела фаз (свободная поверхность) сложной формы. Для адекватного воспроизведения тонких физических эффектов в геометрически сложных трехмерных областях требуются подробные расчетные сетки, содержащие от сотен тысяч до десятков и даже сотен миллионов расчетных ячеек. Подобные задачи требуют огромных вычислительных ресурсов и могут быть решены только на самой современной мощной параллельной вычислительной технике.

В программном комплексе **FlowVision** [1-4] используется наиболее универсальный метод расчета задач вычислительной гидродинамики с использованием эйлеровых адаптивных расчетных сеток для областей произвольно сложной формы. Моделирование свободной поверхности осуществляется во **FlowVision** с использованием подхода VOF (volume of fluid), впервые предложенного в работе [5]. Моделирование течений при наличии в расчетной области подвижных границ требует специальной методологии расчета. Во **FlowVision** используется методология, подробно описанная в работе [2]; аналогичная методика рассматривается в работе [6].

Вопросы организации параллельных вычислений во **FlowVision** при неподвижных границах расчетной области представлены в работах [7-11]. В работе [12] показана параллельная эффективность **FlowVision** при решении на большом числе процессоров задач вычислительной гидродинамики с огромным числом расчетных ячеек.

В работе подробно рассматриваются вопросы организации параллельных вычислений в условиях наличия в расчетной области подвижных границ и свободных поверхностей. Сложность задачи состоит в том, что движение твердых границ и свободных поверхностей во времени приводит к образованию и исчезновению новых расчетных ячеек, причем общее количество ячеек может существенно меняться. В программном комплексе **FlowVision** используется автоматическая динамическая адаптация расчетной сетки в процессе решения, что также приводит

к изменению как топологической структуры сетки, так и ее размера. По этим причинам возникает задача о балансировке параллельных вычислений для расчетной сетки, меняющейся в процессе численного моделирования. Статическая декомпозиция задачи по процессорам (т.е. один раз в начале расчета) может приводить к нарушению межпроцессорного баланса вычислений и потере параллельной эффективности. С другой стороны, полное перевычисление декомпозиции может приводить к большой межпроцессорной миграции данных, поэтому проведение таких вычислений на каждом временном шаге вычислительного алгоритма недопустимо дорого.

В работе [13] авторами был предложен некоторый подход к решению данной задачи. Этот подход основан на единой нумерации всех ячеек сетки с помощью непрерывной кривой, соединяющей центры всех расчетных ячеек, и называемой SFC (Space Filling Curve). Идея построения этой кривой состоит в рекурсивном биении расчетной области на $8=2 \times 2 \times 2$ подобластей тремя координатными плоскостями, и рекурсивной нумерации этих подобластей “N” или “U” способом. Единая нумерация позволяет авторам легко решить проблему сбалансированной декомпозиции задачи по процессорам для любой подобласти простым рекуррентным биением кривой SFC на кусочно непрерывные части с близким к одинаковому числу ячеек в каждом непрерывном кусочке кривой. Такая декомпозиция задачи может быть быстро вычислена, обладает по построению некоторыми свойствами локальности, и во многих случаях дает близкую к оптимальной декомпозиции задачи в общем случае нерегулярной локально адаптированной сетки в сложной трехмерной расчетной области. Эта же декомпозиция задачи и нумерация ячеек сетки может быть использована в решателе систем уравнений, поскольку по структуре она напоминает упорядочивание вложенных сечений ND (Nested Dissection).

На основе подхода SFC можно организовать параллельные вычисления при наличии в расчетной области подвижных элементов [13]. Действительно, при небольших локальных изменениях расчетной сетки можно более-менее локально изменять части кривой SFC, более-менее локально меняя таким образом декомпозицию задачи по процессорам и алгоритмы решения систем уравнений.

Главный недостаток подхода, основанного на SFC, состоит в том, что для такой декомпозиции нельзя во всех случаях гарантировать ее геометрическую локальность и связность получающихся подобластей. Так происходит потому, что на кривой SFC есть участки, которые соединяют две подряд нумеруемые ячейки, расположенные в геометрически сильно разделенных подобластях. Эти участки соответствуют линиям соединения типов “N” или “U”, отвечающим верхним уровням рекурсии разбиения расчетной области. Если при этом граница межпроцессорной декомпозиции пройдет не между этими двумя ячейками, то на некотором процессоре могут одновременно оказаться геометрически очень удаленные друг от друга ячейки. Наличие несвязных подобластей на одном процессоре означает в том числе и неоптимальность размера поверхностной границы с другими процессорами, а значит избыточные обмены данными по этим границам.

В данной работе описывается подход, позволяющий эффективно обойти основные трудности параллельной реализации технологий подвижных тел и свободных поверхностей. Этот подход существенно отличается от подхода, основанного на SFC и описанного в работе [13]. В нем по построению обеспечивается локальность и связность подобластей каждого процессора при декомпозиции задачи. Кроме того, при изменении расчетной сетки также гарантируется минимальное управляемое изменение декомпозиции задачи по процессорам. Достигается это за счет фиксации декомпозиции для внутренних ячеек расчетной области, и возможности перераспределения на другие процессоры ячеек, близких к границам исходных процессорных подобластей. Приводятся результаты численных экспериментов по масштабируемости предложенных алгоритмов на примере моделирования индустриальной задачи вычислительной гидродинамики при помощи **FlowVision**.

2. Автоматическая генерация адаптивной расчетной сетки

Расчетная сетка, используемая во **FlowVision**, является прямоугольной, адаптивной локально измельченной (АЛИС). Адаптация сетки позволяет разрешать малые детали геометрии расчетной области и высокие градиенты рассчитываемых величин с подсеточным разрешением геометрии, которое позволяет аппроксимировать криволинейные границы на прямоугольной

сетке. Задание расчетной сетки во **FlowVision** разделяется на задание начальной сетки (или сетки начального уровня) и адаптацию сетки (как по поверхности, так и в объеме).

Сущность технологии АЛИС заключается в следующем. Во всей расчетной области вводится прямоугольная сетка. Выделяются подобласти с особенностями геометрии или течения, в которых необходимо провести расчет на более мелкой, чем исходная, сетке. При этом расчетная ячейка, в которую попала выделяемая особенность, делится на 8 равных ячеек. Далее, если необходимо, ячейки делятся еще раз, и так до достижения необходимой точности. Ячейки начальной сетки называются ячейками уровня 0, ячейки, получаемые измельчением уровня 0, называются ячейками уровня 1 и т.д. При генерации АЛИС накладывается условие, что гранями и ребрами могут граничить друг с другом только ячейки с номерами уровней, отличающимися не более, чем на единицу.

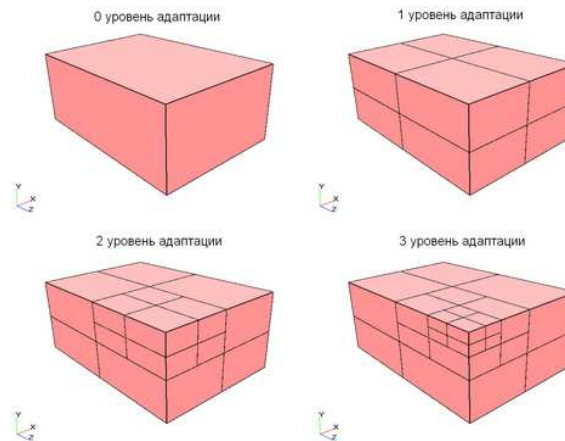


Рис. 1 – Адаптивная сетка с локальным измельчением

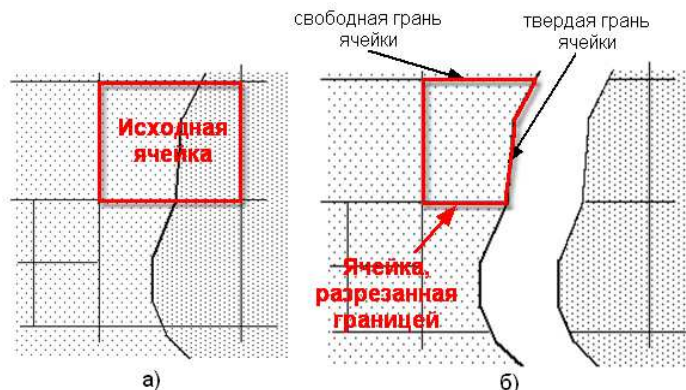


Рис.2 – Метод подсеточного разрешения геометрии: а) поверхность проходит через расчетные ячейки, б) расщепление расчетных ячеек границей.

Метод подсеточного разрешения геометрии, который используется во **FlowVision**, предназначен для аппроксимации криволинейных границ на прямоугольной сетке. Суть этого метода состоит в следующем. Ячейки, через которые проходит граница, расщепляются на 2, 3 и т.д. ячеек. При этом они теряют свою первоначальную форму параллелепипеда и превращаются в многогранники произвольной формы. Уравнения математической модели аппроксимируются для этих многогранников без каких-либо упрощений. Такой подход позволяет с достаточной степенью точности рассчитывать течения даже на грубой расчетной сетке.

3. Организация параллельных вычислений при неподвижной геометрии задачи

Основная идея распараллеливания вычислений во **FlowVision** в случае неподвижной геометрии состоит в *геометрической декомпозиции* задачи [7-9]. Этот подход традиционно используется при распараллеливании вычислений и состоит в расщеплении задачи на некоторое число подобластей, каждая из которых связана и имеет минимальную возможную поверхностную границу с другими подобластями. Предполагается, что каждая такая подобласть распределяется на соответствующий процессор. Для большинства типов вычислений для их полного

проведения с локальными данными процессора достаточно получить некоторые данные от соседних по поверхностной границе подобластей других процессоров. При таком подходе к декомпозиции и распараллеливанию вычислений минимизируются обмены данными с другими процессорами. Для того, чтобы объемы вычислений на каждом процессоре были по возможности близки, необходимо также добиваться чтобы количества расчетных ячеек в подобластях были близки.

При решении систем уравнений распараллеливание вычислений является более сложной задачей. В эффективных однопроцессорных алгоритмах решения систем уравнений существенно используется подходящая нумерация неизвестных. Кроме этого, имеется некоторая геометрически нелокальная зависимость по данным одних вычислений от других. Тем не менее, оказывается, что объемная геометрическая декомпозиция, описанная выше, может снова являться базисом к распараллеливанию вычислений. Изменения с точки зрения распределения данных по процессорам в основном касаются распараллеливания вычислений с поверхностными границами. Некоторые детали организации параллельных вычислений в этом случае подробно рассмотрены в работе [10].

4. Моделирование при наличии подвижных элементов геометрии и свободных поверхностей

Моделирование множества задач вычислительной гидродинамики включает в себя необходимость проведения расчетов в присутствии подвижных элементов геометрии. В качестве примеров можно привести полет вращающегося мяча для гольфа, движение поршня в двигателе внутреннего сгорания под действием продуктов сгорания топлива, моделирование течения в газовой турбине, маневры самолета в полете при изменении положения закрылков, и др. В этих задачах требуется с одной стороны высокоточное моделирование турбулентного движения жидкости/газа, а с другой - необходимо аккуратно учитывать изменение геометрии задачи с течением времени и моделировать влияние этого изменения на исследуемые параметры. Часто возникают также и задачи, в которых присутствуют одновременно и жидкость, и газ, а значит возникает поверхность раздела между этими фазами. В качестве примера можно привести задачи: о движении судна по морю, о вытекании воды из воронки под действием силы тяжести, о вращении шестеренки в поддоне с машинным маслом, и др. Во всех подобных задачах с течением времени меняется расчетная область, а значит методика моделирования должна включать в себя периодическое перестроение расчетной сетки.

Программный комплекс **FlowVision** основан на автоматической генерации адаптивной расчетной сетки с подсеточным разрешением геометрии задачи (см. Раздел 2). Эта технология является основным фундаментом, позволяющим моделирование во **FlowVision** таких приложений вычислительной аэро- гидродинамики. В этом разделе описываются некоторые детали алгоритмов работы **FlowVision** при решении задач с подвижными элементами геометрии и при наличии свободных поверхностей.

4.1 Моделирование при наличии подвижных элементов геометрии

Во **FlowVision** реализовано моделирование течений жидкости/газа при наличии в расчетной области подвижных элементов геометрии. Изменение положения твердого тела приводит к тому (см. Рисунок 3), что часть ячеек, являвшихся расчетными, перестает быть таковыми и попадает внутрь подвижного тела, и наоборот; часть ячеек меняет свою форму, часть ячеек адаптируется, измельчается или укрупняется в соответствии с критериями адаптации, и т.д. Все эти изменения расчетной сетки меняют геометрию ячеек и топологию связей между ними. Кроме того, используются специальные методики [2], по которым данные из исходных расчетных ячеек преобразовываются в данные новых расчетных ячеек.

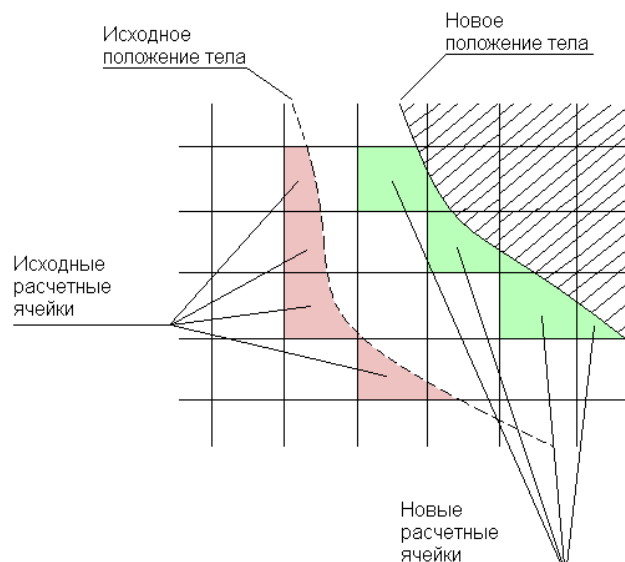


Рис. 3 – Изменение геометрии ячеек сетки и топологии связей между ними при сдвиге подвижного тела

4.2 Моделирование свободной поверхности

Во **FlowVision** реализовано моделирование так называемой *свободной поверхности* жидкости на границе раздела сред жидкость/газ.

Идея реализации моделирования свободной поверхности состоит в ведении для каждой расчетной ячейки параметра VOF (Volume Of Fluid) [5]. Этот параметр представляет собой долю жидкости в заполнении объема текущей ячейки, поэтому $0 \leq VOF \leq 1$. Соответственно, если $VOF = 1$, то текущая ячейка содержит только жидкость. Если $VOF = 0$, то текущая ячейка содержит только газ. Если $0 < VOF < 1$, то в текущей ячейке имеется и жидкость, и газ, и между ними в текущей ячейке имеется граница раздела, именуемая свободной поверхностью. Во **FlowVision** предполагается, что граница раздела в текущей ячейке всегда имеет простейший возможный вид – это плоскость, рассекающая ячейку на две части. Имея значения параметра VOF для всех расчетных ячеек, можно в области расчета восстановить форму поверхности, разделяющую жидкую и газообразную фазы.

Во **FlowVision** для вычисления геометрии ячеек, через которые проходит свободная поверхность, используется методика подсеточного разрешения геометрии. Это позволяет осуществлять перенос VOF с использованием высокоточного метода. По сравнению с подобными методиками, реализованными в других программных комплексах, во **FlowVision** уравнения решаются и в ячейках со свободной поверхностью. В других программных комплексах в ячейках со свободной поверхностью уравнения не решаются, осуществляется только экстраполяция значений переменных, что снижает точность расчетов.

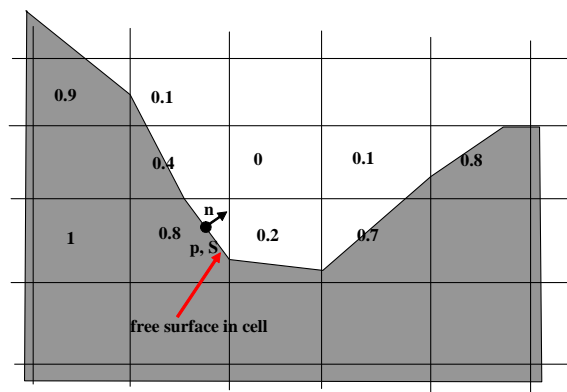


Рис. 4 – Параметр VOF для набора ячеек и свободная поверхность

5. Организация параллельных вычислений при наличии подвижных элементов геометрии и свободных поверхностей

Наличие в расчетной области подвижных элементов геометрии или свободной поверхности приводит к необходимости периодически перестраивать расчетную сетку. Это значит, что число ячеек в фиксированной области пространства может меняться с течением времени. С точки зрения организации параллельных вычислений, это означает необходимость периодически корректировать распределение данных по процессорам для поддержания эффективности параллельных вычислений. В данном разделе описывается реализованная во **FlowVision** организация параллельных вычислений для этого случая.

5.1 Двойная декомпозиция сетки

Подвижный элемент геометрии в задачах аэро- гидродинамики часто представляет собой некоторое твердое или деформируемое трехмерное тело (обозначаемое ниже как подвижное тело), которое движется в расчетной области по заданному закону или под действием гидродинамических и/или других сил. С точки зрения вычисления гидродинамических величин, таких как давление жидкости, скорость потока жидкости, и т.д., во внутренней части подвижного тела расчетные ячейки отсутствуют. При этом собственно ячейки геометрии внутри подвижного тела во **FlowVision** всегда строятся и должны сохраняться, поскольку при достаточно сильном сдвиге подвижного тела эти ячейки могут стать расчетными ячейками сетки.

С точки зрения организации параллельных вычислений все расчетные ячейки должны быть сбалансированным образом распределены по процессорам с учетом геометрической локальности данных (см. геометрическую декомпозицию задачи, описанную в Разделе 3). С другой стороны, нерасчетные ячейки внутри подвижного тела также должны быть распределены по процессорам с учетом условий геометрической локальности данных и их балансировки. По этим причинам во **FlowVision** при наличии подвижных тел производится *двойная декомпозиция сетки*.

Пусть m , $m \geq 1$ - есть некоторое целое число. Для текущего положения подвижных тел и текущей расчетной сетки рассмотрим все расчетные ячейки области, не находящиеся целиком внутри подвижных тел, и обозначим это множество ячеек через g . Обозначим через g_B множество всех ячеек сетки, находящихся целиком внутри подвижного тела. По определению, множество $g \cup g_B$ содержит всю расчетную область, и при этом множества g и g_B не имеют общих элементов.

Присоединим к множеству g все соседние по стороне ячейки из множества g_B . Это один шаг цикла расширения множества расчетных ячеек, обозначим полученное множество g_1^+ . Прделаем это расширение еще $m-1$ раз. Получим множество g_m^+ , состоящее из всех расчетных ячеек вне подвижных тел, плюс m слоев ячеек внутри подвижных тел. По построению, имеет место цепочка вложений $g \subseteq g_1^+ \subseteq \dots \subseteq g_m^+$. Аналогично, присоединим к множеству g_B все соседние по стороне ячейки из множества g . Это один шаг цикла расширения множества ячеек внутри подвижного тела, обозначим полученное множество g_{B1}^+ . Прделаем это расширение еще $m-1$ раз. Получим множество g_{Bm}^+ , состоящее из всех ячеек внутри подвижных тел, плюс m слоев ячеек на границе или вне подвижных тел. По построению, имеет место аналогичная цепочка вложений $g_B \subseteq g_{B1}^+ \subseteq \dots \subseteq g_{Bm}^+$. Обозначим $g_j^- = g \cup g_B / g_{Bj}^+$, $j = 1, \dots, m$, очевидно имеют место соотношения $g_m^- \subseteq \dots \subseteq g_1^- \subseteq g \subseteq g_1^+ \subseteq \dots \subseteq g_m^+$. Множество ячеек $g_m^+ \cap g_{Bm}^+$ образует $2 * m$ слоев ячеек вокруг границ подвижных тел.

Пусть G есть множество всех начальных ячеек сетки, не находящихся целиком внутри подвижных тел, соответственно пусть G_B есть множество всех начальных ячеек сетки, находящихся целиком внутри подвижных тел. По определению множество $G \cup G_B$ содержит всю

расчетную область, и при этом множества G и G_B не имеют общих элементов. Рассмотрим теперь множество G_m^+ тех начальных ячеек, в которых есть хотя бы одна ячейка из описанного выше расширенного множества расчетных ячеек g_m^+ . Обозначим $G_{Bm}^- = G \cup G_B / G_m^+$ - множество начальных ячеек сетки, находящихся в расчетной области и не содержащих ячеек множества g_m^+ . Проводим независимые геометрические декомпозиции подобластей G_m^+ и G_{Bm}^- так, как это описано в Разделе 3. По построению, множества G_m^+ и G_{Bm}^- не имеют общих элементов и покрывают всю расчетную область. На Рисунке 5 показан пример двойной декомпозиции расширенной расчетной области и внутренности подвижного тела на примере задачи о мяче для гольфа.

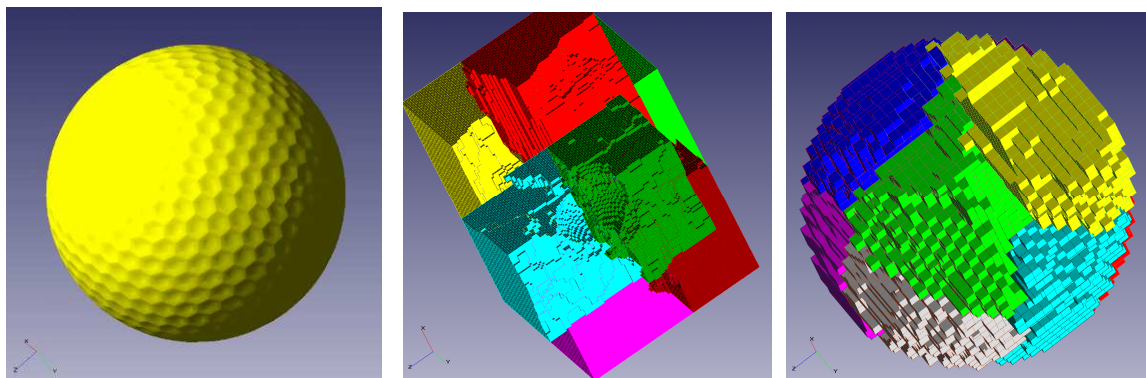


Рис. 5 – Геометрия мяча для гольфа, декомпозиция расширенной области расчетных ячеек сетки G_m^+ (слева) и внутренности подвижного тела G_{Bm}^- (справа) на 8 процессоров.

5.2 Критерии перестроения декомпозиции

При небольших движениях подвижного тела имеет смысл в основном сохранить декомпозицию задачи по процессорам даже в условиях некоторого перестроения расчетной сетки. Этого можно добиться сохраняя декомпозицию по процессорам начальных ячеек. Тем не менее, при относительно больших движениях подвижного тела наоборот необходимо перестроить декомпозицию задачи по процессорам и перенести все необходимые данные на соответствующие процессоры. Иначе возможен сильный дисбаланс вычислений на разных процессорах. С учетом введенных в Разделе 5.1 обозначений, декомпозицию естественно перестраивать если граница подвижного тела начинает проходить через ячейки, не входящие в множество $g_m^+ \cap g_{Bm}^+$ из $2 * m$ слоев ячеек вокруг границы подвижного тела в том его положении, при котором вычислялась текущая декомпозиция данных по процессорам. Другими словами, декомпозиция должна перевычисляться если появляются нерасчетные ячейки вне множества g_{Bm}^+ , либо если появляются расчетные ячейки вне множества g_m^+ .

При таком подходе к перестроению декомпозиции, параметр m становится естественным регулятором частоты ее перестроения по отношению к движению подвижного тела. А именно, чем больше параметр m , тем “реже” перестраивается декомпозиция. С другой стороны, чем больше величина параметра m , тем большее число нерасчетных ячеек (m слоев ячеек внутри подвижного тела около его границы) участвует в декомпозиции, нарушая баланс вычислений. По этим причинам необходимо выбирать параметр m таким образом, чтобы с одной стороны, декомпозиция не перевычислялась слишком часто, а с другой, чтобы не сильно увеличивать дисбаланс вычислений из-за нерасчетных ячеек.

Декомпозицию необходимо перестраивать также и при сильном изменении числа расчетных ячеек на любом из процессоров из-за адаптации сетки.

5.3 Декомпозиция области с минимальным сдвигом по отношению к предыдущей декомпозиции

Передвижение подвижного тела в расчетной области рано или поздно приводит к необходимости перевычисления декомпозиции области по процессорам. Во **FlowVision** предполагается, что шаг по времени достаточно мал, а потому передвижение подвижного тела за один шаг по времени не приводит к слишком значительному перестроению расчетной сетки. По этой причине представляется естественным при построении новой декомпозиции задачи по процессорам осуществить ее явную привязку к предыдущей декомпозиции.

Пусть имеется предыдущая декомпозиция некоторой подобласти G_m расчетной области по процессорам. Под подобластью G_m здесь имеется в виду любая из двух подобластей Раздела 5.1, либо подобласть G_m^+ расчетных ячеек, либо подобласть G_{Bm}^- внутренности подвижных тел. Пусть G_m^{new} - новая подобласть, возникшая в результате движения подвижного тела, и для которой необходимо построить декомпозицию задачи по процессорам, связанную с исходной.

Идея алгоритма построения новой декомпозиции состоит в том, чтобы изменить распределение по процессорам некоторой части начальных ячеек, оставив при этом совпадающим с исходным распределение по процессорам большей части начальных ячеек. Для этого рассмотрим поверхности, разделяющие G_m на подобласти для разных процессоров, см. Рисунок 6. Пусть k , $k \geq 1$ - некоторый целый параметр. Рассмотрим множество всех начальных ячеек G_m , которые лежат на расстоянии не более k ячеек от поверхностей раздела G_m на подобласти для разных процессоров. На Рисунке 6 все такие ячейки около поверхностей раздела обозначены пунктиром. Все внутренние части подобластей каждого процессора остаются данными этого процессора. Только для k приповерхностных слоев ячеек разрешается изменить их процессорную принадлежность. Если в подобласти G_m^{new} имеются ячейки, которых не было в подобласти G_m , то они рассматриваются как ячейки, для которых разрешается вычислить их процессорную принадлежность.

Вычисление новой декомпозиции для числа процессоров – степени двойки можно осуществлять последовательным биением подобласти пополам. А именно, на каждом этапе декомпозиции строим декомпозицию части области G_m^{new} на две группы процессоров. Для этого объединяем в одну огромную суперячейку все начальные ячейки каждой из двух групп процессоров, за исключением тех начальных ячеек, что примыкают к границе между этими группами процессоров в предыдущей декомпозиции. Проводим декомпозицию этих начальных ячеек и двух суперячеек на два набора ячеек с учетом балансировки числа реальных ячеек, и с учетом минимизации связей между наборами. Две огромные суперячейки при этом должны попасть каждая в свой набор. Таким образом, с каждым получившимся новым набором начальных ячеек ассоциируется некоторая подгруппа процессоров. И т.д.

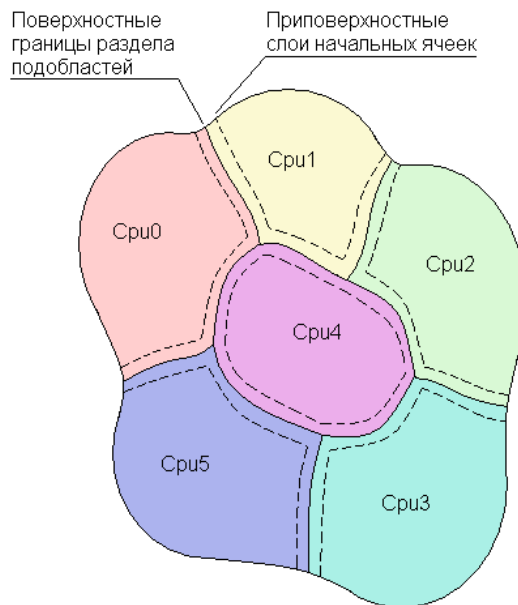


Рис. 6 – Поверхностные границы раздела по процессорам и приповерхностные слои начальных ячеек

6. Результаты численных экспериментов

Для проведения численных экспериментов была выбрана задача о всплытии подводной лодки модельной геометрии (Рис. 7). Лодка движется с горизонтальной скоростью 4 м/с, и всплывает со скоростью 0.1 м/с. При решении данной задачи требуется определить силу сопротивления воды, а также структуру гидродинамического течения вокруг лодки.

Моделирование тестовой задачи проводилось на расчетной сетке с адаптацией к поверхности лодки. Адаптивная расчетная сетка в двух положениях лодки показана на Рисунке 8. Полное число расчетных ячеек в задаче при начальном положении лодки было равно 1.522.292 ячеек и не существенно изменялось при всплытии лодки. Декомпозиция расчетной области на 8 процессоров для последовательности положений лодки приведена на Рисунке 9. На Рисунке 9 хорошо видна связь между декомпозициями задачи при сдвиге положения лодки. На Рисунке 10 показана масштабируемость при решении тестовой задачи вплоть до 16 процессоров. Результаты по масштабируемости при решении тестовой задачи получены на суперкомпьютере СКИФ МГУ “Чебышев” в режиме расчета 2 процессора на узел.

7. Заключение

В работе представлен параллельный алгоритм, позволяющий эффективно организовать вычисления в случае если в расчетной области присутствуют подвижные тела и свободные поверхности. На примере тестовой гидродинамической задачи показана высокая эффективность предложенных в работе алгоритмов.

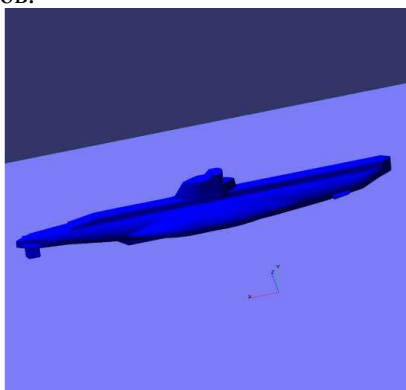


Рис. 7 – Геометрия задачи о подводной лодке

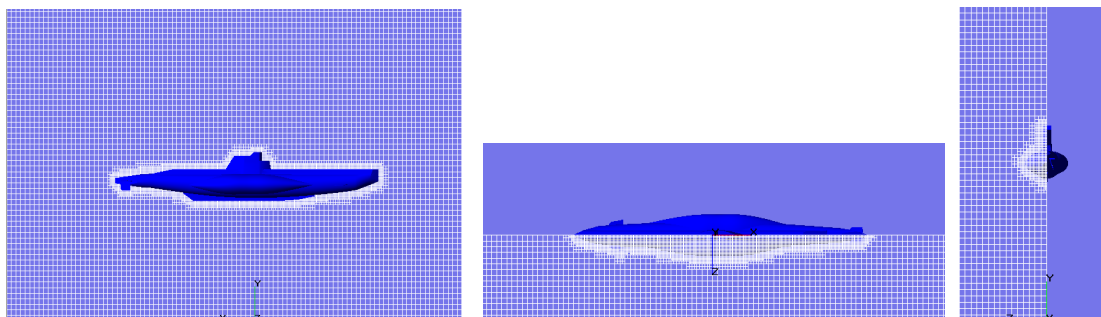


Рис. 8 – Адаптивная сетка в различных ракурсах

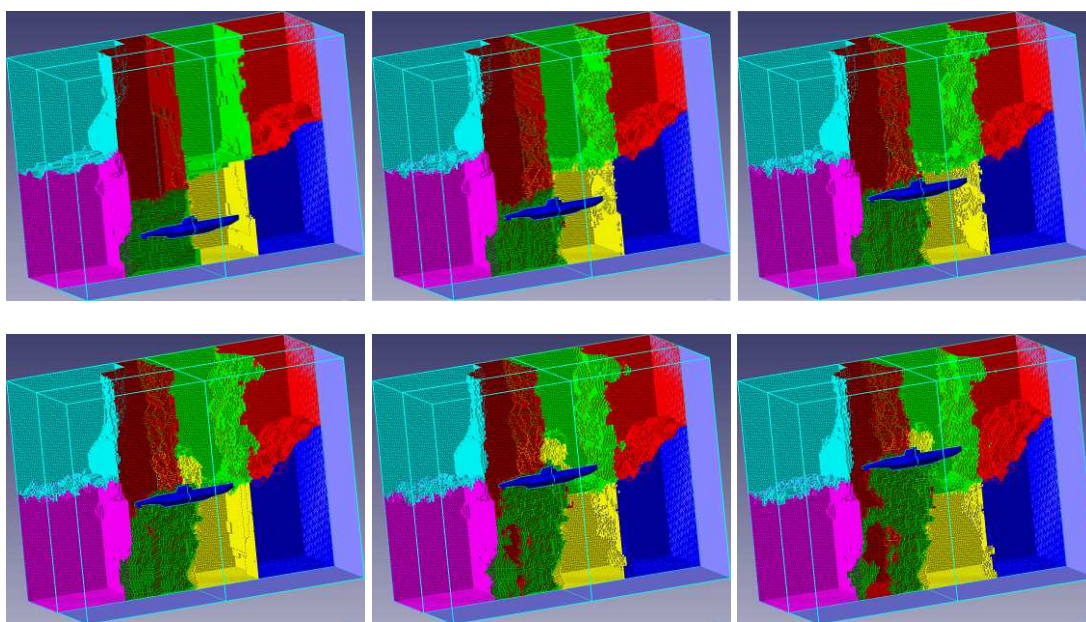


Рис. 9 – Декомпозиция на 8 процессоров для последовательности положений подводной лодки

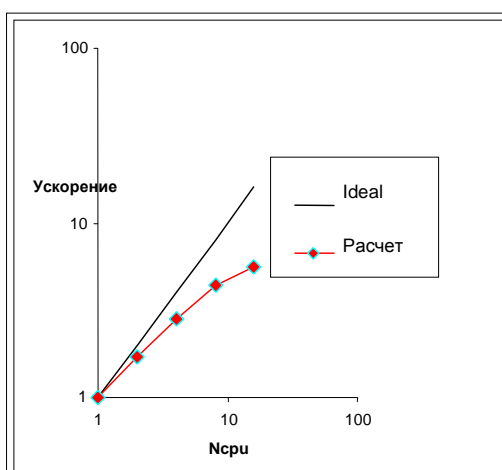


Рис. 10. Ускорение при решении задачи о подводной лодке

Литература

1. Aksenov A., Dyadkin A., Pokhilko V. "Overcoming of Barrier between CAD and CFD by Modified Finite Volume Method", Proc. 1998 ASME Pressure Vessels and Piping Division Conference, San Diego, ASME PVP-Vol 377-1, 1998.
2. Aksenov A., Pokhilko V., Dyadkin A. "Numerical simulation of water flow around ship with screw propeller", Proc. Computational technology (CFD) for fluid/thermal/chemical/stress systems and industrial applications, ASME PVP, July 22-26, 2001 Hyatt Regency, Atlanta, Georgia, USA.
3. A. Aksenov, A. Dyadkin, T. Luniewski, V. Pokhilko. "Fluid Structure Interaction analysis using Abaqus and FlowVision", Proc. Abaqus User Conference, 2004, Boston, USA, 2004.
4. A. Aksenov, K. Iliine, T. Luniewski, T. McCarthy, F. Popielas, R. Ramkumar. "Oil Leakage Through a Valve Stem Seal", Proc. Abaqus User Conference, 2006, Boston, USA, 2006.
5. Hirt C., Nichols B., "Volume of Fluid (VOF) method for the dynamics of free boundaries", J.Comput. Phys., 1981, v. 39, pp 201-225.
6. Murman S.M., Aftosmith M.J., Berger M.J., "Implicit Approaches for Moving Boundaries in a 3-D Cartesian Method", AIAA Paper 2003-1119, Jan. 2003.
7. Aksenov A.A., Kharchenko S.A., Konshin V.N., Pokhilko V.I. "FlowVision software: numerical simulation of industrial CFD applications on parallel computer systems", Parallel CFD 2003 conference, Book of abstracts, p. 280-284, 2003.
8. С.А. Харченко "Параллельная реализация алгоритма решения систем линейных уравнений в пакете FlowVision", в сборнике "Прикладные исследования в механике", труды конференции "Инженерные системы -2007", 135-144, 2007.
9. А.А. Дядькин, С.А. Харченко. "Алгоритмы декомпозиции области и нумерации ячеек с учетом локальных адаптаций расчетной сетки при параллельном решении систем уравнений в пакете FlowVision". Труды Всероссийской научной конференции "Научный сервис в сети Internet: многоядерный компьютерный мир", 201-206, 2007.
10. С.А.Харченко. "Влияние распараллеливания вычислений с поверхностными межпроцессорными границами на масштабируемость параллельного итерационного алгоритма решения систем линейных уравнений на примере уравнений вычислительной гидродинамики", Труды международной научной конференции Параллельные вычислительные технологии (ПаВТ'2008), Санкт-Петербург, 28 января – 1 февраля 2008 г. Челябинск, Изд. ЮУрГУ, 2008, с.494-499.
11. Г.Б. Сушко, С.А. Харченко. "Многопоточная параллельная реализация итерационного алгоритма решения систем линейных уравнений с динамическим распределением нагрузки по нитям вычислений", Труды международной научной конференции Параллельные вычислительные технологии (ПаВТ'2008), Санкт-Петербург, 28 января – 1 февраля 2008 г. Челябинск, Изд. ЮУрГУ, 2008, с.452-457.
12. А.А.Аксенов, А.А.Дядькин, В.А.Кутин, И.В.Москалёв, Г.Б.Сушко, С.А.Харченко. "Решение больших задач вычислительной гидродинамики на СКИФ МГУ с помощью FlowVision", Материалы Всероссийской научной конференции "Научный сервис в сети ИНТЕРНЕТ: решение больших задач", Новороссийск, 22-27 сентября 2008 г., Изд-во Московского Университета, 2008, с. 69-73.
13. Aftosmith M.J., Berger M.J., Murman S.M., "Applications of Space-Filling Curves to Cartesian Methods for CFD", AIAA Paper 2004-1232, Jan. 2004.