УДК 532.5

Параллельная реализация метода SIMPLE на основе многосеточного метода^{*}

А.С. Козелков^{1,2}, С.В. Лашкин¹, А.А. Куркин², А.В. Корнев³, А.М. Вялых¹

¹Российский федеральный ядерный центр. Всероссийский научно-исследовательский институт экспериментальной физики, просп. Мира, 37, Саров, Нижегородская обл., 607188

²Нижегородский государственный технический университет им. Р.Е. Алексеева, ул. Минина, 24, Нижний Новгород, 603950

³Московский авиационный институт (национальный исследовательский университет), Волоколамское шоссе, 4, Москва, 125993

E-mails: askozelkov@mail.ru (Козелков A.C.), S.V.Lashkin@itmf.vniief.ru (Лашкин C.B.), aakurkin@gmail.com (Куркин A.A.), avkornev@mai.ru (Корнев A.B.), dmitv@inbox.ru (Вялых А.М.)

Английская версия этой статьи печатается в журнале "Numerical Analysis and Applications" No 1, Vol. 13, 2020.

Козелков А.С., Лашкин С.В., Куркин А.А., Корнев А.В., Вялых А.М. Параллельная реализация метода SIMPLE на основе многосеточного метода // Сиб. журн. вычисл. математики / РАН. Сиб. отд-ние. — Новосибирск, 2020. — Т. 23, № 1. — С. 1–22.

В работе рассматривается алгоритм параллельной реализации метода SIMPLE (Semi-Implicit Method for Pressure Linked Equations) для численного решения системы уравнений Навье–Стокса для вязких несжимаемых течений. Описан механизм межпроцессных обменов при условии декомпозиции сеточной модели с использованием фиктивных ячеек и применения алгебраического многосеточного метода. Представлено описание хранения распределенных матриц и алгоритм реализации матрично-векторных операций, позволяющий уменьшить число межпроцессных обменов. Приводятся результаты серии численных экспериментов на структурированных и неструктурированных сеточных моделях (включая задачу внешней аэродинамики), на основе которых проводится анализ влияния настроек многосеточного решателя СЛАУ на общую эффективность алгоритма. Показано, что предложенный алгоритм параллельной реализации метода SIMPLE на основе алгебраического многосеточного метода позволяет достаточно эффективно считать задачи на сотнях процессоров.

DOI: 10.15372/SJNM20200101

Ключевые слова: вычислительная гидродинамика, алгоритм SIMPLE, многосеточный решатель, моделирование.

Kozelkov A.S., Lashkin S.V., Kurkin A.A., Kornev A.V., Vyalykh A.M. Effective implementation of the parallel SIMPLE algorithm based on multigrid method // Siberian J. Num. Math. / Sib. Branch of Russ. Acad. of Sci. – Novosibirsk, 2020. – Vol. 23, N \cong 1. – P. 1–22.

This paper deals with the investigation of parallel SIMPLE (Semi-Implicit Method for Pressure Linked Equations) algorithm for the numerical solution of the Navier–Stokes system of equations for viscous incompressible flows. The interprocessor exchange mechanism with mesh decomposition with virtual cells and algebraic multigrid method is presented. The method of distributed matrix storage and the algorithm for matrix-vector operations reducing the number of interprocessor exchanges are presented. The results of a

^{*}Работа выполнена в рамках государственного задания в сфере научной деятельности (задания № 5.4568.2017/6.7, № 5.1246.2017/4.6), при финансовой поддержке грантов Президента Российской Федерации по государственной поддержке научных исследований ведущих научных школ Российской Федерации (проект № НШ-2685.2018.5) и молодых российских ученых-докторов наук (проект № МД-4874.2018.9), при поддержке РФФИ (проект № 16-01-00267).

[©] А.С. Козелков, С.В. Лашкин, А.А. Куркин, А.В. Корнев, А.М. Вялых, 2020

series of the numerical experiments on structured and unstructured grids (including the external aerodynamics problem) are presented. Based on the results obtained, the analysis of the influence of multigrid solver settings on the total algorithm efficiency was made. It was shown that the parallel algorithm for the SIMPLE method based on the algebraic multigrid technique proposed makes possible to efficiently calculate problems on hundreds of processors.

Keywords: computational fluid dynamics (CFD), SIMPLE algorithm, multigrid solver, modeling.

1. Введение

Течения жидкостей и газов описываются краевыми задачами для уравнений Навье– Стокса, представляющих собой систему дифференциальных уравнений в частных производных [1]. Проблема дискретизации уравнений Навье–Стокса, а также проблема их численного решения составляют один из ключевых этапов математического моделирования. Одним из широко известных способов численного решения уравнений Навье– Стокса является алгоритм SIMPLE, основанный на использовании полунеявной схемы с расщеплением по физическим процессам и имеющий несколько модификаций [2]. Описанию данного алгоритма посвящено много статей и монографий, в которых подробно приводятся принципы его реализации на произвольных неструктурированных сетках с применением различных схем и способов дискретизации на неортогональных (в основном тетраэдральных) сетках, позволяющих получать результаты с достаточной степенью соответствия экспериментальным данным [3, 4]. Большинство опубликованных работ основываются на исследовании и описании последовательной версии алгоритма SIMPLE, а его параллельная составляющая отражена недостаточно или вовсе опускается.

Тем не менее, в последние годы наряду с проблемами численной реализации, касающихся точности и расчета на неортогональных сетках, все более важной становится проблема эффективной реализации параллельных вычислений на многопроцессорных суперЭВМ, содержащих десятки тысяч процессорных ядер. Необходимость расчетов на подробных сетках порядка сотен миллионов ячеек (например, при исследовании различных энергетических установок со сложными физическими процессами, при использовании LES-моделей (Large Eddy Simulation) [5, 6]) требует использования мощных ЭВМ [7], поскольку решение задач на малопроцесорных ЭВМ может длиться десятки и сотни часов. Сокращение времени счета можно достичь двумя способами. Первый это увеличение производительности процессора, применяя сопроцессоры фирм NVIDIA и Intel [8, 9], что требует больших затрат на эффективную адаптацию неявных алгоритмов на нерегулярных сетках к их архитектуре с негарантированным положительным результатом. Другим, не менее популярным способом, могут служить методы, предложенные в работах [10, 11]. Суть данных методов сводится к разбиению расчетной области на домены с разной топологией сетки, что в конечном итоге позволяет повысить эффективность и точность вычислений. Достаточно большой объем работ посвящен методам динамической балансировки [12, 13], которые наиболее актуальны для сглаженных частиц. Можно также отметить проблематику ускорения решателей СЛАУ, которую можно выделить в отдельную группу [14, 15].

Понятно, что повышение эффективности какого-либо алгоритма или метода является индивидуальной задачей и не имеет универсального решения. В случае проведения численных экспериментов с использованием SIMPLE-подобных алгоритмов одним из ключевых факторов повышения эффективности счета, как показано в работе [16], является выбор решателя СЛАУ. Как показано в [17–19] самым эффективным способом решения СЛАУ является многосеточный метод AMG (Algebraic Multigrid Method). Однако большое число настроечных параметров (тип цикла, тип сглаживателя, количество итераций сглаживателя, глобальный уровень и так далее) приводит к вопросам его оптимального (эффективного) использования. В большинстве случаев подбор оптимальных параметров многосеточного решателя является индивидуальной задачей, но как показывает вычислительная практика существуют универсальные настройки для конкретных классов задач, например, несжимаемые или сжимаемые течения. В случае неудовлетворительного подбора настроек многосеточного решателя возможно замедление решения СЛАУ и, соответственно, вычислительного алгоритма до нескольких раз [7, 16, 20].

Помимо этого можно получить ускорение путем декомпозиции расчетной модели с использованием фиктивных ячеек, что позволит минимизировать число межпроцессных обменов, в том числе и при матрично-векторных операциях.

В данной статье рассматривается несколько аспектов повышения производительности вычислительного алгоритма SIMPLE для высокопараллельных ЭВМ, а именно: декомпозиция расчетной модели с использованием фиктивных ячеек; параллельный многосеточный решатель СЛАУ и его параметры, структура и операции с матрицей. Дана оценка общей эффективности алгоритма SIMPLE, рассчитанная как аналитически, так и с помощью специальных программ. В конце статьи представлены результаты исследования алгоритма SIMPLE на примере решения задач о движении жидкости в квадратной каверне с движущейся верхней стенкой, турбулентного обтекания плоской пластины и турбулентного обтекания модельного крыла ONERA M6.

2. Итерационная схема алгоритма SIMPLE

Если при разработке численного метода ориентироваться на решение задач на произвольных неструктурированных сетках с ячейками произвольной формы, то оптимальным выбором для дискретизации является метод конечных объемов [1]. Именно этот метод выбран в качестве основного для решения задач вычислительной аэрогидродинамики в пакете программ ЛОГОС [5, 16, 21–23], предназначенном также и для решения связанных и сопряженных задач тепломассопереноса и прочности на параллельных ЭВМ. Пакет программ ЛОГОС успешно прошел верификацию и показал достаточно хорошие результаты на серии гидродинамических задач, включая расчеты турбулентных и нестационарных течений [5, 21, 22], а также геофизических явлений [24, 25].

Одним из алгоритмов, реализованных в пакете программ ЛОГОС, является итерационный алгоритм SIMPLE, который предназначен для решения задач несжимаемых и слабосжимаемых течений.

Несжимаемые течения (течения с постоянной плотностью) описываются системой уравнений Навье–Стокса:

$$\begin{cases} \rho \nabla \cdot \vec{u} = 0, \\ \rho \frac{\partial(\vec{u})}{\partial t} + \rho \nabla \cdot (\vec{u} \otimes \vec{u}) - \nabla \cdot \vec{\tau} = -\nabla p. \end{cases}$$
(1)

В данной системе используются общепринятые обозначения: ρ — плотность моделируемой среды, t — время, $\vec{u} = \{u, v, w\}$ — вектор скорости осредненного течения в трехмерной декартовой системе координат, $\vec{\tau} = \vec{\tau_{\mu}} + \vec{\tau_t}$ — сумма молекулярной и турбулентной (тензор рейнольдсовских напряжений) составляющих вязкой части тензора напряжений, p — давление. Компоненты вязкой части тензора напряжений с учетом гипотезы Буссинеска рассчитываются по следующим формулам:

$$\begin{aligned} \vec{\tau_{\mu}} &= \mu_l \left(\nabla \vec{u} + \left(\nabla \vec{u} \right)^\top - \frac{2}{3} \vec{I} \left(\nabla \cdot \vec{u} \right) \right), \\ \vec{\tau_t} &= \mu_t \left(\nabla \vec{u} + \left(\nabla \vec{u} \right)^\top - \frac{2}{3} \vec{I} \left(\nabla \cdot \vec{u} \right) \right) - \frac{2}{3} \rho k \vec{I} \end{aligned}$$

где μ_l — молекулярная вязкость, μ_t — турбулентная вязкость, вычисляемая согласно используемой RANS-модели (Reynolds-averaged Navier-–Stokes) турбулентности, k — кинетическая энергия турбулентности, \vec{I} — единичный тензор второго ранга, $\nabla \vec{u}$ в трехмерной декартовой системе координат представляет собой тензор второго ранга, состоящий из градиентов трех компонент скоростей:

$$\nabla \vec{u} = \left(\begin{array}{cc} \frac{\partial u}{\partial x}, & \frac{\partial u}{\partial y}, & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x}, & \frac{\partial v}{\partial y}, & \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial x}, & \frac{\partial w}{\partial y}, & \frac{\partial w}{\partial z} \end{array} \right)$$

верхний индекс \top — транспонированный тензор.

Для вычисления турбулентной вязкости используются RANS-модели турбулентности, обеспечивающие приемлемые показатели ресурсоемкости, сроки получения результатов, а также экспериментальную и эмпирическую достоверность результатов. Среди многообразия RANS-моделей хорошо зарекомендовала себя на практике модель SST (Shear Stress Transport) Ментера [26], используемая в данной статье для численных экспериментов.

В случае моделирования несжимаемых сред численное решение системы уравнений (1) затруднительно, так как давление не входит в уравнение неразрывности (в отличие от сжимаемых сред, где давление учитывается в законе идеального газа). Изначально давление не входило в уравнение неразрывности. Без привлечения специальных алгоритмов попытка одновременного решения двух уравнений приводит к неразрешимым трудностям. На сегодняшний день алгоритм SIMPLE является одним из самых универсальных подходов, который обеспечивает связь полей скорости и давления. Его суть заключается в расщеплении исходных величин, согласовывая их в дальнейшем на каждом итерационном шаге:

$$\vec{u}^n = \vec{u}^* + \vec{u}', \qquad p^n = p^{n-1} + p',$$

где \vec{u}' и p' — поля приращений скорости и давления соответственно, n — решение, получаемое на новой итерации, n-1 — решение, получаемое на предыдущей итерации, \vec{u}^* — предварительное поле скорости. Как будет показано далее, такое расщепление позволяет решать уравнение неразрывности относительно приращения давления.

Запишем систему дифференциальных уравнений (1) в дискретном виде:

$$\begin{cases} \rho \nabla \cdot \vec{u}^{n+1} = 0, \\ \rho \frac{\vec{u}^{n+1} - \vec{u}^n}{\Delta t} + \rho \nabla \cdot (\vec{u}^n \otimes \vec{u}^{n+1}) - \nabla \cdot \vec{\tau}^{n+1} = -\nabla p^{n+1}. \end{cases}$$
(2)

Итерационная схема алгоритма SIMPLE подразумевает введение нескольких этапов. На первом этапе — этапе предиктора — решается линейное алгебраическое уравнение сохранения количества движения и находятся предварительные значения компонент вектора скорости $\vec{u}^* = [u_x^*, v_y^*, w_z^*]$ при поле давления с предыдущего итерационного шага:

$$\rho \frac{\vec{u}^* - \vec{u}^n}{\Delta t} + \rho \nabla \cdot (\vec{u}^n \otimes \vec{u}^*) - \nabla \cdot \vec{\tau}^n = -\nabla p^{n-1}.$$
(3)

Найденные таким способом предварительные значения компонент вектора скорости не удовлетворяют уравнению неразрывности. Поэтому на втором этапе вычисляется значение приращения скорости, выражение для которого получается вычитанием из второго уравнения системы (2) уравнения (3) и отбрасыванием второго и третьего слагаемых:

$$\vec{u}' = -\frac{\Delta t}{\rho} \nabla p'. \tag{4}$$

Видно, что поправка для скорости зависит от величины градиента приращения давления. При подстановке равенства (4) в уравнение неразрывности получается уравнение для поправки давления

$$\nabla \cdot \vec{u}^* = \frac{\Delta t}{\rho} \nabla \cdot (\nabla p').$$
(5)

В результате реализацию представленных выше шагов можно записать в виде следующей "полунеявной" итерационной процедуры алгоритма SIMPLE:

- 1) вычисление градиентов скорости и давления;
- 2) решение уравнения сохранения количества движения (3) для вычисления предварительного поля скорости \vec{u}^* , не удовлетворяющего уравнению неразрывности;
- вычисление массовых потоков на гранях контрольного объема: m^{*}_f = ρS_f(ũ^{*}_f n
 f_f), где f — грань контрольного объема, S_f — площадь грани и n
 f_f — единичный вектор нормали;
- 4) вычисление поля приращения давления на основании уравнения (5);
- 5) установление градиента приращения давления и корректировка поля скорости по уравнению (4) и перерасчет массового потока на гранях: $m_f^* = \rho S_f(\vec{u}_f^{n+1} \vec{n}_f)$, который удовлетворяет исходному уравнению неразрывности;
- 6) проверка условия выхода (например, по величине невязки расчетных полей) и при необходимости возврат к шагу 1.

В результате применения данной процедуры получаются четыре системы линейных алгебраических уравнений: три системы для трех компонент скоростей и одна для приращения давления. Решение дополнительных уравнений (например уравнений, моделирующих турбулентность) соответственно увеличивает число СЛАУ.

3. Реализация параллельного алгоритма SIMPLE

Использование метода конечных объемов подразумевает этап восстановления всех физических величин, участвующих в расчете, а также массового потока на гранях, разделяющих контрольные объемы. Для каждого контрольного объема необходимо вычислить потоки импульса и массы через образующие грани f. Для вычисления потока (например потока массы), проходящего через грань f, необходимо знать значения величин скоростей в контрольных объемах P и N (рисунок 1). При совпадении грани с границей раздела MPI-процессов значение потока вычислить невозможно, поскольку образующие грань ячейки физически принадлежат разным процессам. Существует два варианта решения данной проблемы. Первый — ввести понятие "обменных" граней f' и их "специальную" обработку в расчетных алгоритмах, что вводит дополнительные сложности при разработке численных схем, так как требуется написание отдельных (по сравнению с последовательным вариантом) циклов обработки. Подобный вариант используется, например, в [27]. Второй — ввести фиктивные ячейки P' и N' таким образом, что они становятся прообразами счетных ячеек на соседнем MPI-процессе, и организовать MPI-обмены между фиктивными и счетными ячейками (рис. 2). При этом расчетные алгоритмы остаются неизменными, так как грань f считается внутренней и обрабатывается таким же образом, как и все остальные внутренние грани. Преимуществом данного варианта является возможность расширения слоя фиктивных ячеек путем создания следующего слоя ячеек P''и N'', причем количество дополнительных слоев может быть произвольным. Это позволяет расширить сеточный шаблон и применять схемы произвольного порядка точности, ограниченные лишь числом слоев фиктивных ячеек. Именно этот вариант перехода между MPI-процессами используется при параллельном счете задач, приведенных в данной статье, и он полностью адаптирован к произвольным неструктурированным сеткам.



Рис. 1. P и N — соседние контрольные объемы, f — смежная грань, \mathbf{d}_{PN} — вектор, соединяющий центры контрольных объемов, \mathbf{d}_{Pf} и \mathbf{d}_{Nf} — вектора, соединяющие центры контрольных объемов P и N с центром грани f, \mathbf{n}_f — вектор нормали к смежной грани



Рис. 2. Фиктивные ячейки (выделены пунктиром) на соседних процессах

При разбиении расчетной области на домены (один домен для каждого MPI-процесса) расчетные ячейки доменов нумеруются независимо с учетом слоя фиктивных ячеек (рис. 3a), которым ставятся в соответствие счетные ячейки соседних доменов. Вся информация о фиктивных ячейках и соседних процессах запоминается каждым процессом и используется в MPI-обменах.

Для обмена расчетными величинами на этапах итерационного цикла вызываются асинхронные MPI-функции приема и передачи сообщений¹, которые реализованы в процедуре, производящей обмен массивами данных со всеми соседними MPI-процессами одновременно. Если номер текущего процесса больше, чем номер соседнего процесса, то

¹По нашему опыту использование **синхронных** функций обмена приводит к общему замедлению счета на 5–10%.

происходит сначала посылка данных, а потом прием, иначе наоборот. Процедура может обмениваться целочисленными, вещественными и векторными данными, ассоциированными с ячеечными массивами. Данная процедура рассылает информацию из граничных расчетных ячеек каждого процесса в фиктивные ячейки всех соседних MPI-процессов (рис. 3a). Механизм обмена ячеечными массивами одинаков вне зависимости от числа процессоров.



Рис. 3. Пример декомпозиции сеточной модели (а, б) и портреты матриц СЛАУ в однопроцессорном (в) и двухпроцессорном (г) вариантах; элементы фиктивной матрицы выделены пунктиром, символы означают: × — наличие связи между ячейками (см. номера строк и столбцов), • — отсутствие связи; элементы матрицы на 1-м и 2-м процессах, взятые в кружки и квадратики, равны между собой соответственно

Параллельная версия вычислительной процедуры алгоритма SIMPLE отличается от исходной последовательной версии только включением межпроцессных обменов и содержит аналогичные шаги:

- 1) вычисление градиентов трех компонент скоростей и давления с межпроцессными обменами векторными градиентами величин;
- решение уравнения сохранения количества движения для счетных и фиктивных ячеек для вычисления предварительного поля вектора скорости, распределенного по процессам. Межпроцессные обмены массивами скоростей выполняются на каждой итерации линейного решателя до сходимости;
- вычисление массовых потоков на гранях контрольного объема. Данные для вычисления потока на гранях, разделяющих счетную и фиктивную ячейки, берутся из фиктивных ячеек, информация в которых появилась после межпроцессных обменов на шаге 1;
- вычисление поля приращения давления на основании уравнения (5) для счетных и фиктивных ячеек. Межпроцессные обмены производятся аналогично шагу 2 на каждой итерации линейного решателя;
- 5) вычисление градиента приращения давления и корректировка поля скорости по уравнению (4), перерасчет массового потока на гранях, удовлетворяющего исходному уравнению неразрывности. Осуществление межпроцессных обменов данными о градиенте приращения давления и подправленным полем скорости;

6) проверка условия выхода (например, по величине невязки расчетных полей) и при необходимости возврат к шагу 1. Проведение коллективных межпроцессных операций суммирования.

В зависимости от физики моделируемого процесса возможно решение дополнительных уравнений переноса для турбулентности, энергии, компонент вещества и т. д. Межпроцессные обмены ячеечными массивами дополнительных полей осуществляются на каждом итерационном шаге алгоритма SIMPLE и при решении соответствующих СЛАУ в количестве, равном количеству внутренних итераций, необходимых для сходимости.

4. Организация решения линейных уравнений

Важным фактором эффективной реализации параллельного неявного алгоритма является этап решения СЛАУ. В зависимости от задачи и способа решения СЛАУ количество внутренних линейных итераций может варьироваться от одной до десятков тысяч. В алгоритме SIMPLE матрица СЛАУ для давления часто не имеет строгого диагонального преобладания, и ее число обусловленности может достигать величины $10^7 \div 10^9$, что затрудняет решение СЛАУ итерационными методами [16, 28] в подпространствах Крылова, на которое может затрачиваться более 90% времени расчетного шага.

В программном комплексе ЛОГОС для решения наиболее сложной СЛАУ давления используется алгебраический многосеточный метод (AMG), детали реализации которого подробно изложены в [16, 20, 28], а для решения СЛАУ для скоростей и турбулентных параметров — симметричный решатель Гаусса–Зейделя.

Основная идея многосеточного метода заключается в иерархическом построении и сохранении последовательности вложенных грубых матриц СЛАУ и операторов произвольного перехода от одной матрицы СЛАУ к другой. В зависимости от параметров огрубления количество этих грубых матриц (уровней) может быть произвольным. Процесс решения начинается с исходной СЛАУ (с 0-го уровня) и последующей интерполяции найденного решения и невязки на более грубые уровни. При достижении самого грубого уровня процесс интерполяции решения и невязки начинается в обратном направлении. Такая итерационная процедура позволяет значительно сократить количество внутренних итераций до момента получения необходимой точности решения.

Важным фактором эффективной реализации многосеточного решателя является сглаживатель. В данной работе роль основного сглаживателя выполняет симметричный метод Гаусса–Зейделя [18]:

$$x_{i}^{\text{new}} = \frac{\tilde{b}_{i} - \sum_{j=i+1}^{N} a_{ij} x_{i}^{\text{old}}}{a_{ii}}, \\ \tilde{b}_{j} = \tilde{b}_{j} - a_{ji} x_{i}^{\text{old}}, \quad j \in \{i-1, \dots, N\}, \end{cases}$$
 $i \in \{1, \dots, N\}$

Итерационная процедура многосеточного решателя совместно со сглаживателем Гаусса–Зейделя позволяют значительно сокращать время решения СЛАУ давления в итерационном алгоритме SIMPLE, что продемонстрированно в [21].

Основная счетная нагрузка в последовательном и параллельном вариантах многосеточного решателя ложится на создание грубых уровней и на итерационный вызов вычислительной процедуры сглаживателя. Если параллельное построение грубых уровней обсуждено в [21], то осуществление параллельных итераций сглаживателем освещено достаточно слабо. При введении фиктивных ячеек существенным плюсом является возможность избавиться от дополнительных обменов при матрично-векторных операциях. Покажем это на примере.

Для этого рассмотрим более подробно распределенное хранение матрицы. Для хранения матрицы используется формат LDU. Отдельно хранятся диагональ D, верхняя треугольная U и нижняя треугольная L части матрицы. Соответственно, матрица Aпредставляется в виде A = L + D + U. Три массива являются одномерными и содержат значения элементов матрицы. Массив, хранящий диагональ, индексируется номером соответствующей ячейки. Два других массива индексируются номерами ячеек, определенных в дополнительных массивах lP и uP, хранящие строки и столбцы матрицы соответственно. Заметим, что, как и для гранево-ячеечного представления, номер ячейки lPвсегда меньше номера uP. Упорядочение массивов представлено на рис. 4.

Прямое сходство массивов lP, uP с гранево-ячеечным представлением сеточной модели не требует дополнительного выделения памяти для массивов и обеспечивает прямую связь форматов хранения сеточной модели и матрицы СЛАУ.



Рис. 4. Представление формата *LDU*

Эффективная параллельная реализация на представленной структуре данных предполагает построение матрицы каждым MPI-процессом как для счетных, так и фиктивных ячеек. На рис. 3 показаны портреты матрицы СЛАУ, созданные для уравнения неразрывности на восьми ячейках расчетной области при двух процессных вариантах. Межпроцессная граница разделяет при этом ячейки под номерами 2 и 3, 6 и 7. Найденные при линеаризации уравнений недиагональные коэффициенты матрицы, отвечающие за связь между ячейками, выделены в виде одинаковых фигур. Операция умножения локальных матриц на локальные части распределенного вектора приводит к глобальному вектору, в точности такому же, как на одном процессе.

Предложенный подход открывает путь, например в многосеточном решателе СЛАУ, имеющем в основе сглаживатель Гаусса–Зейделя, к использованию только одного межпроцессного обмена для каждой внутренней итерации, т.е. при вычислении значений x^{n+1} и b в прямом и обратном ходах, а также при определении невязки r = b - Axмежпроцессные обмены отсутствуют. В итоге, сглаживателю методом Гаусса–Зейделя потребуется только один межпроцессный обмен информацией о фиктивном фрагменте вектора неизвестных x^{n+1} в конце единичной внутренней итерации.

Также можно утверждать, что данный подход к параллельному решению СЛАУ позволяет избавиться от дополнительного межпроцессного обмена данными о фиктивном фрагменте вектора неизвестных при "получении" решения из решателя СЛАУ, так как этот обмен уже произошел непосредственно в линейном решателе.

Следующим шагом на пути к эффективной реализации решателя СЛАУ является настройка решателя AMG таким образом, чтобы минимизировать количество внутренних итераций многосеточного решателя СЛАУ. Чем меньше внутренних итераций решателя СЛАУ, тем меньше межпроцессных обменов, увеличивающих время расчета (одна внутренняя итерация решателя — один межпроцессный обмен). Основными настройками, влияющими на количество внутренних итераций многосеточного решателя, являются:

- 1) тип используемого цикла (V-, W- или F-цикл);
- 2) количество итераций сглаживателя на каждом уровне AMG;
- 3) количество ячеек для огрубления (2, 4, 8).

Для обоснования выбора оптимальных настроек многосеточного решателя СЛАУ, обеспечивающих минимальное количество итераций, приведем результаты численных экспериментов следующих задач: течения в замкнутой каверне с движущейся верхней крышкой (Задача 1) [29], турбулентное обтекание плоской пластины (Задача 2) [1] и течение в канале с обратным уступом (Задача 3) [30]. В качестве сглаживателя будем использовать симметричный алгоритм Гаусса–Зейделя как наименее затратный по числу машинных операций на одной итерации [31, 32], и рекурсивное огрубление производить до тех пор, пока на самом грубом уровне останется максимум 5 счетных ячеек. Все полученные результаты по каждой изменяемой настройке сведены в три нижеприведенные таблицы.

Таблица 1 содержит полное количество внутренних итераций для каждого V-, W- и F-цикла, необходимых сглаживателю в многосеточном решателе до полной сходимости задачи, решаемой с использованием алгоритма SIMPLE. Также в таблице приведено общее время счета задачи в стационарной постановке до сходимости по массе порядка 10^{-6} , вычисляемой по формуле

$$\operatorname{res}_{m} = \sum_{i=1}^{N} \left(\sum_{f=nb(P)} \rho_{f} S_{f}(\vec{u}_{f} \cdot \vec{n}_{f}) \right), \tag{10}$$

где суммирование осуществляется по всем N счетным ячейкам модели.

| Номер задачи | Полное в | количество | итераций | Время счета, с | | | |
|--------------|----------|------------|----------------|----------------|--------|----------------|--|
| | V-цикл | W-цикл | F-цик л | V-цикл | W-цикл | F-цик л | |
| 1 | 15633 | 492093 | 45090 | 8.9 | 27.4 | 12.3 | |
| 2 | 48384 | 1034775 | 107730 | 17.4 | 48.5 | 20.0 | |
| 3 | 61416 | 1354076 | 198378 | 28.3 | 89.2 | 38.8 | |

Таблица 1. Результаты расчетов по применению V-, W- и F-циклов

Данная таблица однозначно показывает, что применение V-цикла в многосеточном решателе СЛАУ существенно сокращает общее количество итераций и время решения данных задач. По сравнению с V-циклом количество итераций для F-цикла выше в 2 или 3 раза, а для W-цикла — в 20–30 раз. Общее время счета задачи также минимально при использовании V-цикла.

Далее будем рассматривать только V-цикл. Общее количество V-циклов в решении задачи зависит от количества итераций сглаживателя на каждом уровне многосеточного метода. Чем больше их число, тем меньше V-циклов требуется для полного решения задачи. Однако с увеличением количества итераций многосеточный решатель "вырождается" в решатель Гаусса–Зейделя, что приводит к замедлению расчета. В случае высокопараллельных вычислений замедление становится более существенным, так количество межпроцессных обменов увеличивается. Таблица 2 содержит данные по влиянию изменения количества итераций сглаживателя на каждом уровне на время решения представленных задач. Анализ данной таблицы показывает, что применение 2-х или 3-х итераций обеспечивает минимальное физическое время счета хотя бы одной задачи. Использование одной и более четырех итераций сглаживателя приводит к замедлению счета задачи, и, в случае шести итераций, замедление достигает 30%.

Таблица 2. Зависимость количества V-циклов от количества итераций сглаживателя на иерархических уровнях многосеточного метода

| | Количество итераций сглаживателя | | | | | | | |
|--------------|----------------------------------|-----------|-----------|-----------|-----------|-----------|--|--|
| Номер задачи | 1 | 2 | 3 | 4 | 5 | 6 | | |
| | о V-циклов | | | | | | | |
| 1 | 10.7/1553 | 8.6/769 | 8.8/579 | 9.2/501 | 9.9/472 | 11.2/470 | | |
| 2 | 28.9/9773 | 25.6/5104 | 22.2/3270 | 22.3/2431 | 22.6/2056 | 23.3/1807 | | |
| 3 | 27.1/10120 | 24.9/5011 | 24.4/3448 | 24.7/2796 | 25.4/2477 | 26.8/2225 | | |

Дальнейшая задача состоит в определении степени огрубления (оптимального количества ячеек, образующих грубую ячейку), влияющей на количество грубых уровней и суммарное число грубых ячеек. Чем выше данный параметр, тем меньше генерируется уровней и грубых ячеек, и наоборот. Следовательно, требуется меньше времени на построение грубых уровней и выполнение одной итерации AMG (одного V-цикла), но тем больше итераций сглаживателя и итераций AMG необходимо для решения задачи, т. к. при уменьшении числа грубых уровней многосеточный решатель постепенно вырождается в обычный решатель Гаусса–Зейделя. Таблица 3 содержит данные о влиянии количества ячеек для огрубления на общее количество итераций сглаживателя и физическое время счета задачи.

| Номер задачи | Количеств | во итераций | сглаживателя | Время счета, с | | | |
|--------------|-----------|-------------|--------------|----------------|----------|---------|--|
| | 2 ячейки | 4 ячейки | 8 ячеек | 2 ячейки | 4 ячейки | 8 ячеек | |
| 1 | 13842 | 13296 | 14430 | 9.2 | 8.0 | 8.6 | |
| 2 | 41472 | 41832 | 48656 | 24.7 | 21.3 | 24.8 | |
| 3 | 51052 | 50110 | 51370 | 30.7 | 25.3 | 25.7 | |

Таблица 3. Результаты расчетов по изменению количества ячеек для огрубления

Необходимо отметить, что при увеличении количества ячеек для огрубления общее количество уровней уменьшается, и, как показано в таблице, количество итераций и время счета задачи начинают расти при использовании 8 ячеек. В данных тестовых случаях оптимальным вариантом будет огрубление по 4 ячейки для генерации следующих грубых уровней.

Общий анализ полученных результатов показывает, что для оптимального использования многосеточного решателя необходимо в настройках задавать V-цикл, 3 итерации сглаживателя на каждом уровне и огрублять по 4-м ячейкам. Опыт расчета производственных задач показывает, что данные настройки не являются универсальными², и для некоторых задач оптимальными могут являться другие типы циклов и количество ячеек для огрубления. Стоит отметить, что самые устойчивые и, соответственно, достаточно медленные настройки решателя СЛАУ получаются при использовании F-цикла, 3-х итераций сглаживателя и 2-х ячеек для огрубления.

²Например, при счете сжимаемых задач целесообразней использовать F-цикл по причине его большей устойчивости. Данные настройки по умолчанию используются в пакете программ ЛОГОС.

Данные настройки совместно с решением системы до относительной точности порядка 0.1 и ограничением максимального числа итераций до 30 [16, 20] позволяют получать минимальное количество внутренних циклов и, соответственно, минимальное время решения большинства задач гидродинамики с использованием алгоритма SIMPLE. Анализ применения различных методов ускорения газодинамических расчетов детально обсуждается в [20].

5. Способы оценки эффективности

В данной статье эффективность работы расчетного алгоритма SIMPLE измерялась двумя различными способами — прямым и косвенным. Прямая (реальная) оценка эффективности основана на сравнении времени расчета задачи на одном процессоре со временем расчета этой же задачи на нескольких процессорах.

Так, ускорение параллельной реализации на *p* процессорах вычисляется по формуле

$$S_p = \frac{T_1}{T_p},$$

где T_1 — это время выполнения расчета задачи на одном процессоре, T_p — время решения задачи на p процессорах.

Для оценки масштабируемости параллельного алгоритма используется понятие расчетной эффективности

$$E_p = \frac{S_p}{p} 100\%.$$

Понятно, что из-за наличия межпроцессных обменов эффективность $E_p = 100\%$ никогда недостижима в реальных расчетах.

В настоящее время существует большое разнообразие программ косвенной (программной) оценки эффективности параллельной реализации алгоритмов (см., например, [33]) на используемом числе процессоров без проведения экспериментов на одном процессоре. Здесь воспользуемся программным средством STK (Statistics Tool Kit) [33], позволяющим оценить эффективность параллельных алгоритмов, которые используют MPIинтерфейс передачи сообщений [34] и OpenMP-интерфейс [35] многопоточного распараллеливания на вычислительных узлах с общей памятью. Программа STK позволяет проводить мониторинг и анализ эффективности использования вычислительных ресурсов многопроцессорных ЭВМ. Помимо этого в STK имеются средства для определения причин неэффективной работы как программы в целом, так и отдельных ее фрагментов.

В STK эффективность выполнения задачи основана на подсчете отношения времени вычислений ко всему времени счета. Для каждого MPI-процесса параллельной задачи измеряется время выполнения вызовов функций MPI и ввода-вывода ($T_{\rm change}$) и общее время ($T_{\rm total}$) выполнения программы. Используя эти данные, можно вычислить показатель эффективности выполнения параллельной программы отдельным MPI-процессом по формуле

$$E_i^{\rm STK} = \frac{T_a}{T_a + T_{\rm change}} 100\%,$$

где $T_a = T_{\text{total}} - T_{\text{change}}$ — время арифметической работы, i — номер процесса. Для оценки эффективности выполнения параллельной программы вычисляется средняя эффективность выполнения всех MPI-процессов по формуле

$$E_{\rm avr}^{\rm STK} = \sum_{i=1}^{N} E_i^{\rm STK} 100\%,$$

где N — количество процессов, $E_{\rm avr}^{\rm STK}$ — показатель средней программной эффективности выполнения параллельной задачи.

В эффективном параллельном алгоритме при условии равномерной декомпозиции (количество ячеек на каждом MPI-процессе равно) все MPI-процессы должны затрачивать одно и то же время на решение своего участка задачи. Однако в реальных условиях это труднодостижимо (наличие разномасштабных ячеек, ввод-вывод, сопряженный теплообмен и т.д.). Поэтому для каждого MPI-процесса вычисляется свой показатель эффективности E_i^{STK} , среди которых по всем MPI-процессам вычисляются минимальный E_{\min}^{STK} , максимальный E_{\max}^{STK} и средний $E_{\text{avr}}^{\text{STK}}$ показатели программной эффективности. Вместе они дают представление о дисбалансе вычислиений параллельной задачи. Близость $E_{\text{avr}}^{\text{STK}}$ к E_{\max}^{STK} свидетельствует о слабом вычислительном дисбалансе (небольшое число MPI-процессов выполняют меньше вычислительной работы, чем остальные), близость $E_{\text{avr}}^{\text{STK}}$ к E_{\max}^{STK} при $E_{\max}^{\text{STK}} \ll E_{\max}^{\text{STK}}$ свидетельствует о сильном вычислительном дисбалансе нислительном дисбалансе (небольшое число MPI-процессов выполняют меньше вычислительной работы, чем остальные), близость E_{\max}^{STK} к E_{\min}^{STK} при $E_{\max}^{\text{STK}} \ll E_{\max}^{\text{STK}}$ свидетельствует о сильном вычислительном дисбалансе.

6. Численные эксперименты

Для определения эффективности представленной параллельной реализации алгоритма SIMPLE декомпозиция представленных тестовых задач проводится на разное число процессов, заданное формулой

$$N = 2^i$$

где i варьируется в зависимости от задачи и ограничивается числом ячеек на одном MPI-процессе не меньше 10000.

Первая задача — это внутренние течения, которые широко распространены практически во всех отраслях промышленности (автомобильная, атомная и т. д.). Вторая и третья задачи — внешнее обтекание тела — типичные задачи автомобильной и авиационной промышленности.

Течения во всех нижеприведенных численных экспериментах считаются турбулентными и для расчетов используется модель турбулентности SST с автоматическим определением зоны пограничного слоя [26]. Разрешение сетки в пристеночной области выбиралось исходя из безразмерного расстояния до стенки: $y^+ \simeq 1$. Сравнение с экспериментальными или аналитическими данными для представленных задач в данной работе не приводится, но результаты моделирования достоверны. Подробную информацию о точности моделирования различных течений в пакете программ ЛОГОС можно найти в работах [16, 21–23].

6.1. Движение жидкости в квадратной каверне с движущейся верхней стенкой

Рассмотрим квадратную каверну с подвижной верхней стенкой. Длина стороны квадрата h = 1 м. Верхняя стенка движется в своей собственной плоскости с постоянной скоростью U = 1 м/с, боковые и нижняя стенки неподвижны.

Параметры вязкой несжимаемой жидкости, находящейся в каверне: плотность $\rho = 100 \,\mathrm{kr/m^3}$; молекулярная вязкость $\mu = 0.1 \,\mathrm{Ia} \cdot \mathrm{c}$. Число Рейнольдса $\mathrm{Re} = \frac{\rho U h}{\mu} = 1000$. Задача рассматривается в двумерной постановке. В качестве граничных условий применяются жесткие стенки с прилипанием, верхняя стенка имеет постоянную скорость $U = 1 \,\mathrm{m/c}$, боковые и нижняя стенки неподвижны.

Для данной задачи проведем оценку эффективности рассмотренного метода в зависимости от сеточного параметра *h*. Для этого будем использовать 3 расчетные сетки с характерным размером ячеек 0.002 м, 0.001 м и 0.0005 м. Характеристики расчетных моделей приведены в табл. 4. Одна из моделей расчетной области представлена на рис. 5а. В расчетах используется трехмерная сеточная модель толщиной в одну ячейку (исходная постановка задачи — трехмерная).

| Номер сетки | Характерный размер ячеек, м | Общее количество ячеек |
|-------------|-----------------------------|------------------------|
| 1 | 0.002 | 250000 |
| 2 | 0.001 | 1000000 |
| 3 | 0.0005 | 4000000 |

| Габлица 4. Ха | рактеристика | расчетных | моделей |
|---------------|--------------|-----------|---------|
|---------------|--------------|-----------|---------|



Рис. 5. Сеточные модели каверны (а), плоской пластины (б) и крыла ONERA M6 (в, г)

Данные о времени решения задач на различном количестве MPI-процессов с приведением оценок ускорения по двум методикам для сеток с номерами 1, 2 и 3 приведены в таблицах 5, 6, 7 соответственно.

Идеальное ускорение недостижимо в реальных расчетах ввиду выполнения необходимых межпроцессных обменов, нелинейно увеличивающихся с ростом числа расчетных MPI-процессов и числа фиктивных ячеек, расчет в которых отчасти дублируется³, что в свою очередь приводит к замедлению расчета задачи в целом. Найти баланс между ускорением и эффективностью использования вычислительных ресурсов — достаточно трудоемкая задача и зачастую требует индивидуального подхода в каждом конкретном случае.

 $^{^{3}}$ Например, для подобласти из 10000 ячеек слой фиктивных ячеек (размер границы) составляет $\sim 30\%$ от числа всех ячеек подобласти одного MPI-процесса, что в значительной степени определяет увеличение отличий между реальной эффективностью и эффективностью по STK с уменьшением размера подобластей, т. к. по STK вычисления в фиктивных ячейках и нефиктивных ячейках неотличимы.

| Varunaarna | V. a managemente | | | | | | |
|------------|------------------|----------|-------|---------------|----------------------|----------------------|-----------------------------|
| количество | количество | D | a | F (64) | DSTK (07) | DSTK (07) | DSTK (M) |
| MPI- | ячеек на один | Время, с | S_p | $E_p(\%)$ | $E_{\min}^{STR}(\%)$ | $E_{\max}^{STR}(\%)$ | $E_{\rm avr}^{\rm STR}(\%)$ |
| процессов | МРІ-процесс | | | | | | |
| 1 | 250000 | 525.77 | 1.00 | 100.00 | 100.00 | 100.00 | 100.00 |
| 2 | 125000 | 297.05 | 1.77 | 88.50 | 98.17 | 98.68 | 98.48 |
| 4 | 62500 | 172.38 | 3.05 | 76.30 | 92.97 | 94.80 | 94.43 |
| 8 | 31250 | 101.06 | 5.20 | 65.00 | 67.16 | 94.89 | 87.75 |
| 16 | 15625 | 52.50 | 10.02 | 62.64 | 73.16 | 91.53 | 85.99 |
| 32 | 7812.5 | 27.50 | 19.11 | 59.70 | 52.47 | 81.88 | 75.87 |
| 64 | 3906.25 | 16.13 | 32.60 | 50.90 | 50.49 | 66.65 | 60.12 |
| 128 | 1953.125 | 8.75 | 60.10 | 46.95 | 25.49 | 37.27 | 32.72 |

Таблица 5. Эффективность параллельной реализации для сетки №1

Таблица 6. Эффективность параллельной реализации для сетки №2

| Количество MPI- процессов | Количество ячеек на один MPI-пропесс | Время, с | S_p | $E_p\left(\%\right)$ | E_{\min}^{STK} (%) | $E_{\max}^{\text{STK}}\left(\% ight)$ | E_{avr}^{STK} (%) |
|---------------------------------|--|----------|-------|----------------------|-----------------------------|---------------------------------------|------------------------------|
| 1 | 1000000 | 2087.60 | 1.00 | 100.00 | 100.00 | 100.00 | 100.00 |
| 2 | 500000 | 1160.00 | 1.80 | 90.00 | 97.80 | 98.40 | 98.16 |
| 4 | 250000 | 707.66 | 2.95 | 73.75 | 92.14 | 94.18 | 94.14 |
| 8 | 125000 | 407.70 | 5.12 | 64.00 | 66.50 | 94.78 | 87.35 |
| 16 | 62500 | 217.30 | 9.60 | 60.00 | 71.90 | 91.17 | 85.42 |
| 32 | 31250 | 111.20 | 18.77 | 58.67 | 51.50 | 81.47 | 75.23 |
| 64 | 15625 | 63.30 | 32.98 | 51.53 | 49.80 | 66.24 | 60.01 |
| 128 | 7812.5 | 33.78 | 61.80 | 48.34 | 24.70 | 37.13 | 32.12 |

Таблица 7. Эффективность параллельной реализации для сетки №3

| Количество МРІ- процессов | Количество ячеек на один MPI-процесс | Время, с | S_p | E_p (%) | $E_{\min}^{\text{STK}}\left(\% ight)$ | E_{\max}^{STK} (%) | $E_{\mathrm{avr}}^{\mathrm{STK}}\left(\% ight)$ |
|---------------------------------|--|----------|-------|-----------|---------------------------------------|-----------------------------|---|
| 4 | 1000000 | 3616.7 | 2.88 | 72.11 | 92.45 | 94.05 | 94.04 |
| 8 | 500000 | 1890.4 | 5.51 | 68.92 | 64.76 | 94.42 | 87.18 |
| 16 | 250000 | 982.6 | 10.6 | 66.25 | 72.20 | 91.24 | 85.10 |
| 32 | 125000 | 569.2 | 18.3 | 57.11 | 51.18 | 81.14 | 75.07 |
| 64 | 62500 | 322.9 | 32.26 | 50.4 | 49.09 | 66.27 | 60.02 |
| 128 | 31250 | 107.64 | 75.6 | 45.34 | 24.87 | 37.01 | 32.10 |

Для данной задачи анализ результатов, полученных на трех сеточных моделях, показывает, что реальная эффективность обеспечивается при декомпозиции на 32 MPIпроцесса. Например, для первой сеточной модели реальная эффективность $E_p = 59.7\%$, при этом средняя программная эффективность $E_{\rm avr}^{\rm STK}$ составляет 75.87%. Дальнейшее увеличение числа процессоров приводит к заметному снижению эффективности. Так, при декомпозиции на 64 процесса реальная эффективность $E_p = 50.9\%$, программная эффективность $E_{\rm avr}^{\rm STK} = 60.12\%$. Отличия связаны с обработкой фиктивных ячеек, увеличивающей вычислительную нагрузку на процессоры. В случае отсутствия фиктивных ячеек показатель программной эффективности был бы ниже. Таким образом, для данной задачи минимальное время счета достигается при количестве ячеек порядка $10^4 - 2 \cdot 10^4$ в каждом MPI-процессе. В дополнение можно отметить отсутствие дисбаланса в вычислениях ($E_{\min}^{\text{STK}} \approx E_{\max}^{\text{STK}}$), свидетельствующее о равномерности вычислительной нагрузки на все MPI-процессы.

6.2. Турбулентное обтекание плоской пластины

Рассмотрим стационарное турбулентное течение вязкой несжимаемой жидкости вблизи плоской пластины длиной L = 1 м и толщиной h/2 = 0.05 мм с закруглением на носике. Сеточная модель приведена на рис. 5б. Задача рассматривается в двумерной постановке, т. е. по третьему направлению одна ячейка, на соответствующих поверхностях задается граничное условие симметрии (SYMMETRY). На входной границе задан постоянный расход (INLET), на внешней стенке пластины задано условие прилипания (WALL NO SLIP). Выходная граница описывается границей с заданным статическим давлением p = 0. Толщина пристеночной ячейки составляет 10^{-6} м. Характерный размер ячеек — 10^{-4} м. Сетка содержит 2 миллиона ячеек, на рис. 56 — часть расчетной сетки в зоне передней кромки пластины.

Используются следующие характеристики среды:

- плотность $\rho = 1.214187 \, \mathrm{kg/m^3},$
- молекулярная вязкость $\mu = 1.85 \cdot 10^{-5} \, \mathrm{kr} / (\mathrm{m \cdot c}).$

Результаты расчета сравнения времени решения задачи на различном количестве MPI-процессов с приведением оценок ускорения по двум методикам приведены в табл. 8.

| Количество | Количество | | | | | |
|------------|---------------|------------|--------|----------------------|----------------------|-----------------------------|
| MPI- | ячеек на один | Время, с | S_p | $E_{\min}^{STK}(\%)$ | $E_{\max}^{STK}(\%)$ | $E_{\rm avr}^{\rm STK}(\%)$ |
| процессов | MPI-процесс | | | | | |
| 4 | 512591.3 | 1365.34295 | 3.80 | 95.55 | 96.49 | 96.01 |
| 8 | 256295.6 | 719.894556 | 7.59 | 92.24 | 93.57 | 92.94 |
| 16 | 128147.8 | 368.763345 | 14.81 | 88.72 | 90.66 | 89.66 |
| 32 | 64073.91 | 217.674312 | 25.09 | 73.83 | 74.73 | 74.24 |
| 64 | 32036.95 | 123.851372 | 44.10 | 61.82 | 64.33 | 63.01 |
| 128 | 16018.48 | 73.8174404 | 73.98 | 52.80 | 55.32 | 53.98 |
| 256 | 8009.238 | 45.4923423 | 120.05 | 43.71 | 45.60 | 44.29 |
| 512 | 4004.619 | 39.2388725 | 139.18 | 30.09 | 33.27 | 32.19 |

Таблица 8. Эффективность параллельной реализации

Как видно из таблицы минимальное время счета обеспечивается при расчете в каждом MPI-процессе ~ 14000 ячеек — примерно столько же, как в предыдущих экспериментах.

Анализ результатов вышеприведенных численных экспериментов показал, что минимальное время счета достигается для разных задач при ~ 10^4 счетных ячейках в каждом MPI-процессе. При этом показатели реальной и программной эффективности находятся на уровне $E_p \approx 30\%$ и $E_{\rm avr}^{\rm STK} \approx 35\%$ и превышают 50% при более $2 \cdot 10^4$ ячеек в каждом MPI-процессе. Разница в показателях эффективности связана в первую очередь с наличием фиктивных ячеек и возрастающей вычислительной нагрузкой на каждый MPI-процесс при увеличении числа процессорных ядер.

6.3. Обтекание модельного крыла ONERA M6

В данной задаче рассматривается задача обтекания тестового крыла ONERA M6 [23]. Обтекание моделировалось при угле атаки $\alpha = 3.06^{\circ}$, числе Маха M = 0.8395 и числе Рейнольдса $\text{Re} = 1.172 \cdot 10^{6}$. В эксперименте использовалась трехмерная сеточная модель. Расчетная сетка имеет призматические слои вдоль стенок крыла с коэффициентом роста, равным 1.2 (см. рисунки 5в и 5г). Толщина первой расчетной призматической ячейки тетраэдральной сетки составляет $1 \cdot 10^{-3}$ м. Характерный размер ячеек 10^{-4} м. Общая расчетная область представляет из себя куб с размерами $40 \text{ м} \times 40 \text{ м} \times 20 \text{ м}$. Геометрия расчетной области и внешний вид расчетных сеток представлены на рис. 5в. Общее число ячеек составляет 6117769.

Результаты сравнения времени решения задачи на различном количестве MPI-процессов приведены в табл. 9. Они основаны на измерении программной эффективности, начиная с 8-ми процессоров. Это связано с невозможностью посчитать данную задачу на одном процессе ввиду ограничения по объему доступной оперативной памяти. В данном случае S_p воспроизводится из $E_{\rm avr}^{\rm STK}$ и является приближенной величиной с погрешность ~ 12%, судя по результатам из табл. 9.

| Количество МРІ- процессов | Количество ячеек на один MPI-процесс | Время, с | S_p | $E_{\min}^{\text{STK}}(\%)$ | $E_{\max}^{STK}(\%)$ | $E_{\rm avr}^{\rm STK}(\%)$ |
|---------------------------------|--|----------|--------|-----------------------------|----------------------|-----------------------------|
| 8 | 764721.13 | 6110.76 | 7.60 | 92.95 | 97.76 | 95.10 |
| 16 | 382360.56 | 3221.97 | 14.90 | 90.51 | 97.58 | 93.00 |
| 32 | 191180.28 | 1650.44 | 28.10 | 86.17 | 91.97 | 87.70 |
| 64 | 95590.14 | 974.23 | 52.60 | 79.98 | 89.67 | 82.20 |
| 128 | 47795.07 | 554.31 | 104.40 | 77.96 | 85.23 | 81.50 |
| 256 | 23897.54 | 330.38 | 175.10 | 65.40 | 71.55 | 68.39 |
| 512 | 11948.77 | 203.61 | 284.10 | 53.32 | 58.02 | 55.52 |
| 1024 | 5974.38 | 175.62 | 329.30 | 29.80 | 33.68 | 32.15 |

Таблица 9. Эффективность параллельной реализации

Как видно из таблицы минимальное время счета обеспечивается при расчете в каждом MPI-процессе ~ 14000 ячеек — примерно столько же, как в предыдущих экспериментах.

Анализ результатов вышеприведенных численных экспериментов показал, что минимальное время счета достигается для разных задач при ~ 10^4 счетных ячейках в каждом MPI-процессе. При этом показатели реальной и программной эффективности находятся на уровне $E_p \approx 30\%$ и $E_{\rm avr}^{\rm STK} \approx 35\%$ и превышают 50% при более $2 \cdot 10^4$ ячеек в каждом MPI-процессе. Разница в показателях эффективности связана в первую очередь с наличием фиктивных ячеек и возрастающей вычислительной нагрузкой на каждый MPI-процесс при увеличении числа процессорных ядер.

7. Заключение

В данной работе рассмотрен алгоритм параллельной реализации метода SIMPLE. Приведено подробное описание шагов итерационного алгоритма SIMPLE в последовательном и параллельном режимах. Особое внимание уделено особенностям эффективного распараллеливания многосеточного решателя СЛАУ. Предложенный способ построения локальных матриц с учетом фиктивных ячеек позволил уменьшить число межпроцессных обменов в векторно-матричных операциях. В серии численных экспериментов с вариацией настроек алгоритма определены оптимальные параметры для эффективного расчета на сотнях процессоров. При этом предложенный алгоритм реализации системы уравнений Навье–Стокса методом SIMPLE позволяет проводить эффективное численное моделирование трансзвуковых течений сжимаемого газа.

Литература

- 1. Лойцянский Л.Г. Механика жидкости и газа. М.-Л.: Гостехиздат, 1950.
- 2. Moukalled F., Darwish M. A Unified formulation of the segregated class of algorithms for fluid flow at all speeds // Numerical Heat Transfer. 2000. Vol. 37, Nº 1. P. 103–139.
- 3. Shterev K.S., Stefanov S.K. Pressure based finite volume method for calculation of compressible viscous gas flows // J. Computational Physics. -2010. Vol. 229, № 2. P. 461-480.
- Katz A., Sankaran V. High aspect ratio grid effects on the accuracy of Navier–Stokes solutions on unstructured meshes // Computers & Fluids. - 2012. - Vol. 65. - P. 66-79.
- 5. Козелков А.С., Курулин В.В., Тятюшкина Е.С., Пучкова О.Л. Моделирование течений вязкой несжимаемой жидкости на неструктурированных сетках методом отсоединенных вихрей // Математическое моделирование. — 2014. — Т. 26, № 8. — С. 81–96.
- 6. Козелков А.С., Курулин В.В. Численная схема для моделирования турбулентных течений несжимаемой жидкости с использованием вихреразрешающих подходов // Журн. вычисл. матем. и мат. физики. 2015. Т. 55, № 7. С. 1255–1266. Перевод: Kozelkov A.S., Kurulin V.V. Eddy-resolving numerical scheme for simulation of turbulent incompressible flows // USSR Comput. Math. and Math. Phys. 2015. Vol. 55, № 7. Р. 1232–1241.
- 7. Козелков А.С., Шагалиев Р.М., Курулин В.В., Ялозо А.В., Лашкин С.В. Исследование потенциала суперкомпьютеров для масштабируемого численного моделирования задач гидродинамики в индустриальных приложениях // Журн. вычисл. матем. и мат. физики. 2016. Т. 56, № 8. С. 1524–1535. Перевод: Kozelkov A.S., Shagaliev R.M., Kurulin V.V., Yalozo A.V., Lashkin S.V. Investigation of supercomputer capabilities for the scalable numerical simulation of computational fluid dynamics problems in industrial applications // USSR Comput. Math. and Math. Phys. 2016. Vol. 56, № 8. Р. 1506–1516.
- Barth M., Byckling M., Ilieva N., Saarinen S., and Schliephake M. Best Practice Guide – Intel Xeon Phi v1.1. / V. Weinberg. – 2014. – URL: https://www.researchgate.net/ publication/260369376.
- 9. Lindholm E., Nickolls J., Oberman S., and Montrym J. NVIDIA Tesla: a unified graphics and computing architecture // IEEE MICRO. 2008. Vol. 28, iss. 2. P. 39–55.
- Исаев С.А., Баранов П.А., Усачов А.Е. Многоблочные вычислительные технологии в пакете VP2/3 по аэротермодинамике. — Саарбрюкен: LAP LAMBERT Academic Publishing, 2013.
- Gicquel L.Y.M, Gourdain N., Boussuge J.-F., Deniau H., Staffelbach G., and Wolf P. High performance parallel computing of flows in complex geometries // Comptes Rendus Mecanique. - 2011. - Vol. 339, iss. 2-3. - P. 104–124.
- Guo X., Rogers B.D., Lind S., and Stansby P.K. New massively parallel scheme for Incompressible Smoothed Particle Hydrodynamics (ISPH) for highly nonlinear and distorted flow // Computer Physics Communications. - 2018. - Vol. 233. - P. 16-28.
- 13. Chow A.D., Rogers B.D., Lind S.J., and Stansby P.K. Incompressible SPH (ISPH) with fast Poisson solver on a GPU // Computer Physics Communications. 2018. Vol. 226. P. 81-103.

- 14. Notay Y., Napov A. A massively parallel solver for discrete Poisson-like problems // J. Computational Physics. 2015. Vol. 281. P. 237-250.
- 15. Emans M. Performance of parallel AMG-preconditioners in CFD-codes for weakly compressible flows // Parallel Computing. 2010. Vol. 36, Nº 5-6. P. 326-338.
- 16. Козелков А.С., Дерюгин Ю.Н., Лашкин С.В. и др. Реализация метода расчета вязкой несжимаемой жидкости с использованием многосеточного метода на основе алгоритма SIMPLE в пакете программ ЛОГОС // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. — 2013. — № 4. — С. 44–56. Перевод: Kozelkov A.S., Deryugin Yu.N., Lashkin S.V. et. al. Implementation in Logos software of a computational scheme for a viscous incompressible fluid using the multigrid method based on algorithm SIMPLE // VANT. Ser.: Mat. Mod. Fiz. Proc. — 2013. — № 4. — Р. 31–43.
- Brandt A. Guide to multigrid development // Multigrid Methods. Proc. Conf. Held at Koln-Porz, November 23–27, 1981 / W. Hackbusch, U. Trottenberg. Springer, 1982. P. 220–312. (Lect. Notes in Mathematics; 960).
- 18. Saad Y. Iterative Methods for Sparse Linear Systems. Second ed. Minneapolis: SIAM, 2003
- 19. Vanek P., Mandel J., and Brezina M. Algebraic multigrid by smoothed aggregation for second and fourth order problems // Computing. 1996. Vol. 56, iss. 3. P. 179-196.
- Волков К.Н., Дерюгин Ю.Н., Емельянов В.Н., Козелков А.С., Тетерина И.В. Алгебраический многосеточный метод в задачах вычислительной гидродинамики // Вычислительные методы и программирование. — 2014. — Т. 15. — С. 183–200.
- 21. Козелков А.С., Курулин В.В., Пучкова О.Л., Лашкин С.В. Моделирование турбулентных течений с использованием алгебраической модели рейнольдсовых напряжений с универсальными пристеночными функциями // Вычислительная механика сплошных сред. — 2014. — Т. 7, № 1. — С. 40–51.
- 22. Бойко А.В., Нечепуренко Ю.М., Жучков Р.Н., Козелков А.С. Блок расчета положения ламинарно-турбулентного перехода для пакета программ ЛОГОС // Теплофизика и аэромеханика. — 2014. — Т. 21, № 2. — С. 201–220.
- 23. Сафронов А.В., Дерюгин Ю.Н., Жучков Р.Н. и др. Результаты валидации многофункционального пакета программ ЛОГОС при решении задач аэрогазодинамики старта и полета ракет-носителей // Математическое моделирование. — 2014. — Т. 26, № 9. — С. 83–95.
- 24. Козелков А.С., Куркин А.А., Пелиновский Е.Н., Курулин В.В., Тятюшкина Е.С. Моделирование возмущений в озере Чебаркуль при падении метеорита в 2013 году // Известия РАН. Механика жидкости и газа. — 2015. — № 6. — С. 134–149. Перевод: Kozelkov A.S., Kurkin A.A., Pelinovskii E.N., Kurulin V.V., Tyatyushkina E.S. Modeling the disturbances in the lake Chebarkul caused by the fall of the meteorite in 2013 // Fluid Dynamics. — 2015. — Vol. 50, № 6. — Р. 828–840.
- 25. Козелков А.С., Куркин А.А., Пелиновский Е.Н. Влияние угла входа тела в воду на высоты генерируемых волн // Известия РАН. Механика жидкости и газа. — 2016. — № 2. — С. 166–176. Перевод: Kozelkov A.S., Kurkin A.A., Pelinovskii E.N. Effect of the angle of water entry of a body on the generated wave heights // Fluid Dynamics. — 2016. — Vol. 51, № 2. — P. 288–298.
- Menter F.R. Two-equation eddy-viscosity turbulence models for engineering applications // AIAA J. - 1994. - Vol. 32, № 6. - P. 1598–1605.
- 27. OpenFOAM. URL: http://openfoam.org/.
- 28. Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы.—М: Лаборатория Базовых Знаний, 2002.
- 29. Ghia U., Ghia K.N., and Shin C.T. High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method // J. Computational Physics. -- 1982. -- Vol. 48, iss. 3. P. 387-411.

- 30. Vogel J.C., Eaton J.K. Combined heat transfer and fluid dynamic measurements downstream of a backward-facing step // J. Heat Transfer. 1985. Vol. 107, iss. 4. P. 922-929.
- 31. Уоткинс Д.С. Основы матричных вычислений. М.: БИНОМ, 2009.
- 32. Woodward C.S. SUNDIALS: Suite of Nonlinear and Differential/Algebraic Equation Solvers.— Livermore: Lawrence Livermore National Laboratory.—(UCRL-PRES-213978).
- 33. Новаев Д.А., Бартенев Ю.Г., Липов Д.И. и др. Программные средства STK для исследования эффективности выполнения параллельных приложений // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. — 2011. — Вып. 4. — С. 72–81.
- 34. Snir M., Otto S., Huss-Lederman S., Walker D., and Dongarra J. MPI: The Complete Reference. MIT Press, 1996.
- 35. Антонов А.С. Параллельное программирование с использованием технологии OpenMP. Учебное пособие. М.: Изд-во МГУ, 2009.

Поступила в редакцию 12 июля 2018 г. После исправления 7 ноября 2018 г. Принята к печати 15 октября 2019 г.

Литература в транслитерации

- 1. Loicyanskii L.G. Mekhanika zhidkosti i gaza. M.-L.: Gostekhizdat, 1950.
- 2. Moukalled F., Darwish M. A Unified formulation of the segregated class of algorithms for fluid flow at all speeds // Numerical Heat Transfer. 2000. Vol. 37, Nº 1. P. 103–139.
- 3. Shterev K.S., Stefanov S.K. Pressure based finite volume method for calculation of compressible viscous gas flows // J. Computational Physics. −2010. − Vol. 229, Nº 2. − P. 461–480.
- 4. Katz A., Sankaran V. High aspect ratio grid effects on the accuracy of Navier–Stokes solutions on unstructured meshes // Computers & Fluids. 2012. Vol. 65. P. 66-79.
- Kozelkov A.S., Kurulin V.V., Tyatyushkina E.S., Puchkova O.L. Modelirovanie techenii vyazkoi neszhimaemoi zhidkosti na nestrukturirovannyh setkah metodom otsoedinennyh vihrei // Matematicheskoe modelirovanie. - 2014. - T. 26, N^Q 8. - S. 81-96.
- 6. Kozelkov A.S., Kurulin V.V. Chislennaya skhema dlya modelirovaniya turbulentnyh techenii neszhimaemoi zhidkosti s ispol'zovaniem vihrerazreshayuschih podhodov // Zhurn. vychisl. matem. i mat. fiziki. 2015. T. 55, № 7. S. 1255–1266. Perevod: Kozelkov A.S., Kurulin V.V. Eddy-resolving numerical scheme for simulation of turbulent incompressible flows // USSR Comput. Math. and Math. Phys. 2015. Vol. 55, № 7. P. 1232–1241.
- 7. Kozelkov A.S., Shagaliev R.M., Kurulin V.V., Yalozo A.V., Lashkin S.V. Issledovanie potenciala superkomp'yuterov dlya masshtabiruemogo chislennogo modelirovaniya zadach gidrodinamiki v industrial'nyh prilozheniyah // Zhurn. vychisl. matem. i mat. fiziki. 2016. T. 56, Nº 8.—S. 1524–1535. Perevod: Kozelkov A.S., Shagaliev R.M., Kurulin V.V., Yalozo A.V., Lashkin S.V. Investigation of supercomputer capabilities for the scalable numerical simulation of computational fluid dynamics problems in industrial applications // USSR Comput. Math. and Math. Phys.—2016.—Vol. 56, Nº 8.—P. 1506–1516.
- Barth M., Byckling M., Ilieva N., Saarinen S., and Schliephake M. Best Practice Guide – Intel Xeon Phi v1.1. / V. Weinberg. – 2014. – URL: https://www.researchgate.net/ publication/260369376.
- Lindholm E., Nickolls J., Oberman S., and Montrym J. NVIDIA Tesla: a unified graphics and computing architecture // IEEE MICRO. – 2008. – Vol. 28, iss. 2. – P. 39–55.

- 10. Isaev S.A., Baranov P.A., Usachov A.E. Mnogoblochnye vychislitel'nye tekhnologii v pakete VP2/3 po aerotermodinamike.—Saarbryuken: LAP LAMBERT Academic Publishing, 2013.
- 11. Gicquel L.Y.M, Gourdain N., Boussuge J.-F., Deniau H., Staffelbach G., and Wolf P. High performance parallel computing of flows in complex geometries // Comptes Rendus Mecanique. 2011. Vol. 339, iss. 2-3. P. 104-124.
- Guo X., Rogers B.D., Lind S., and Stansby P.K. New massively parallel scheme for Incompressible Smoothed Particle Hydrodynamics (ISPH) for highly nonlinear and distorted flow // Computer Physics Communications. - 2018. - Vol. 233. - P. 16-28.
- 13. Chow A.D., Rogers B.D., Lind S.J., and Stansby P.K. Incompressible SPH (ISPH) with fast Poisson solver on a GPU // Computer Physics Communications. 2018. Vol. 226. P. 81-103.
- 14. Notay Y., Napov A. A massively parallel solver for discrete Poisson-like problems // J. Computational Physics. 2015. Vol. 281. P. 237-250.
- 15. Emans M. Performance of parallel AMG-preconditioners in CFD-codes for weakly compressible flows // Parallel Computing. 2010. Vol. 36, Nº 5-6. P. 326-338.
- 16. Kozelkov A.S., Deryugin Yu.N., Lashkin S.V. i dr. Realizaciya metoda rascheta vyazkoi neszhimaemoi zhidkosti s ispol'zovaniem mnogosetochnogo metoda na osnove algoritma SIMPLE v pakete programm LOGOS // Voprosy atomnoi nauki i tekhniki. Ser. Matematicheskoe modelirovanie fizicheskih processov. 2013. Nº 4. S. 44-56. Perevod: Kozelkov A.S., Deryugin Yu.N., Lashkin S.V. et. al. Implementation in Logos software of a computational scheme for a viscous incompressible fluid using the multigrid method based on algorithm SIMPLE // VANT. Ser.: Mat. Mod. Fiz. Proc. 2013. Nº 4. P. 31-43.
- Brandt A. Guide to multigrid development // Multigrid Methods. Proc. Conf. Held at Koln-Porz, November 23–27, 1981 / W. Hackbusch, U. Trottenberg. Springer, 1982. P. 220–312. (Lect. Notes in Mathematics; 960).
- 18. Saad Y. Iterative Methods for Sparse Linear Systems. Second ed. Minneapolis: SIAM, 2003
- 19. Vanek P., Mandel J., and Brezina M. Algebraic multigrid by smoothed aggregation for second and fourth order problems // Computing. 1996. Vol. 56, iss. 3. P. 179-196.
- 20. Volkov K.N., Deryugin Yu.N., Emel'yanov V.N., Kozelkov A.S., Teterina I.V. Algebraicheskii mnogosetochnyi metod v zadachah vychislitel'noi gidrodinamiki // Vychislitel'nye metody i programmirovanie. 2014. T. 15. S. 183–200.
- 21. Kozelkov A.S., Kurulin V.V., Puchkova O.L., Lashkin S.V. Modelirovanie turbulentnyh techenii s ispol'zovaniem algebraicheskoi modeli reinol'dsovyh napryazhenii s universal'nymi pristenochnymi funkciyami // Vychislitel'naya mekhanika sploshnyh sred. 2014. T. 7, № 1. S. 40-51.
- 22. Boiko A.V., Nechepurenko Yu.M., Zhuchkov R.N., Kozelkov A.S. Blok rascheta polozheniya laminarno-turbulentnogo perekhoda dlya paketa programm LOGOS // Teplofizika i aeromekhanika. −2014. −T. 21, Nº 2. −S. 201–220.
- 23. Safronov A.V., Deryugin Yu.N., Zhuchkov R.N. i dr. Rezul'taty validacii mnogofunkcional'nogo paketa programm LOGOS pri reshenii zadach aerogazodinamiki starta i poleta raket-nositelei // Matematicheskoe modelirovanie. 2014. T. 26, № 9. S. 83–95.
- 24. Kozelkov A.S., Kurkin A.A., Pelinovskii E.N., Kurulin V.V., Tyatyushkina E.S. Modelirovanie vozmuschenii v ozere Chebarkul' pri padenii meteorita v 2013 godu // Izvestiya RAN. Mekhanika zhidkosti i gaza. 2015. № 6. S. 134–149. Perevod: Kozelkov A.S., Kurkin A.A., Pelinovskii E.N., Kurulin V.V., Tyatyushkina E.S. Modeling the disturbances in the lake Chebarkul caused by the fall of the meteorite in 2013 // Fluid Dynamics. 2015. Vol. 50, № 6. P. 828–840.

- 25. Kozelkov A.S., Kurkin A.A., Pelinovskii E.N. Vliyanie ugla vhoda tela v vodu na vysoty generiruemyh voln // Izvestiya RAN. Mekhanika zhidkosti i gaza. 2016. Nº 2. S. 166–176. Perevod: Kozelkov A.S., Kurkin A.A., Pelinovskii E.N. Effect of the angle of water entry of a body on the generated wave heights // Fluid Dynamics. 2016. Vol. 51, Nº 2. P. 288–298.
- Menter F.R. Two-equation eddy-viscosity turbulence models for engineering applications // AIAA J. - 1994. - Vol. 32, № 6. - P. 1598–1605.
- 27. OpenFOAM. URL: http://openfoam.org/.
- 28. Bahvalov N.S., Zhidkov N.P., Kobel'kov G.M. Chislennye metody. M: Laboratoriya Bazovyh Znanii, 2002.
- 29. Ghia U., Ghia K.N., and Shin C.T. High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method // J. Computational Physics. -- 1982. -- Vol. 48, iss. 3. P. 387-411.
- 30. Vogel J.C., Eaton J.K. Combined heat transfer and fluid dynamic measurements downstream of a backward-facing step // J. Heat Transfer. 1985. Vol. 107, iss. 4. P. 922-929.
- 31. Uotkins D.S. Osnovy matrichnyh vychislenii. M.: BINOM, 2009.
- 32. Woodward C.S. SUNDIALS: Suite of Nonlinear and Differential/Algebraic Equation Solvers.— Livermore: Lawrence Livermore National Laboratory.—(UCRL-PRES-213978).
- Novaev D.A., Bartenev Yu.G., Lipov D.I. i dr. Programmnye sredstva STK dlya issledovaniya effektivnosti vypolneniya parallel'nyh prilozhenii // Voprosy atomnoi nauki i tekhniki. Ser. Matematicheskoe modelirovanie fizicheskih processov. — 2011. — Vyp. 4. — S. 72– 81.
- 34. Snir M., Otto S., Huss-Lederman S., Walker D., and Dongarra J. MPI: The Complete Reference. MIT Press, 1996.
- 35. Antonov A.S. Parallel'noe programmirovanie s ispol'zovaniem tekhnologii OpenMP. Uchebnoe posobie. M.: Izd-vo MGU, 2009.