



DOI: 10.15514/ISPRAS-2023-35(5)-19

Использование метода декомпозиции области для распараллеливания моделирования течения вязкой несжимаемой среды методом LS-STAG и дополнительного предобуславливания

¹ И.К. Марчевский, ORCID: 0000-0003-4899-4828 <iliamarchevsky@mail.ru>

² В.В. Пузикова, ORCID: 0000-0003-0712-4519 <v.puzikova@yadro.com>

¹ Московский государственный технический университет им. Н.Э. Баумана,
105005, Россия, г. Москва, ул. 2-я Бауманская, д. 5.

² ООО YADRO,
123022, Россия, г. Москва, ул. Рочдельская, д. 15с15.

Аннотация. В ходе численного решения задач механики сплошной среды основная часть вычислительных затрат, как правило, приходится на решение больших разреженных систем линейных алгебраических уравнений. По этой причине эффективное распараллеливание именно этой процедуры может значительно ускорить моделирование. Наиболее простой подход к решению этой задачи, заключающийся в распараллеливании матрично-векторных операций в обычном итерационном решателе, требует нескольких точек синхронизации и обменов коэффициентами на каждой итерации метода, что не позволяет значительно ускорить расчет в целом. Поэтому предпочтительнее оказываются методы декомпозиции области, которые подразумевают разбиение расчетной области на подобласти, построение и решение отдельных задач в них, а также некоторую процедуру согласования решения между подобластями для обеспечения глобальной сходимости. Подобласти могут перекрываться, как в методе Шварца, используемом в OpenFOAM, или разделяться интерфейсными участками, для которых решается своя собственная интерфейсная задача, как в методе дополнения Шура. Последний метод используется в данной работе для построения параллельного алгоритма моделирования течений вязкой несжимаемой среды методом погруженных границ LS-STAG. Полученная матрица интерфейсной системы имеет блочную трехдиагональную структуру. Для ускорения прототипирования в программной реализации разработанного алгоритма использована технология параллельного программирования OpenMP, поэтому вычислительные эксперименты проводятся только на системах с общей памятью, в частности на отдельных узлах учебно-экспериментального кластера кафедры «Прикладная математика» МГТУ им. Н. Э. Баумана. Для верификации и оценки эффективности разработанного алгоритма рассмотрена хорошо исследованная тестовая задача о моделировании плоского обтекания неподвижного кругового профиля. Расчеты на последовательности сеток при их разделении на разное количество подобластей показывают, что параллельный алгоритм сходится к тому же решению, что и исходный алгоритм, а рассчитанные значения числа Струхала и коэффициента лобового сопротивления хорошо согласуются с известными в литературе экспериментальными и расчетными данными. Эксперименты демонстрируют, что разработанный алгоритм с декомпозицией области позволяет ускорить моделирование даже в последовательном режиме за счет уменьшения количества итераций, то есть метод декомпозиции области действует как дополнительный предобуславливатель. Благодаря этому свойству при расчетах в параллельном режиме ускорение оказывается сверхлинейным до некоторого числа подобластей, зависящего от размера задачи.

Ключевые слова: метод декомпозиции области; метод погруженных границ; метод LS-STAG; вязкая несжимаемая среда; система линейных алгебраических уравнений; предобуславливание; метод FGMRES; параллельные вычисления; OpenMP; OpenFOAM.

Для цитирования: Марчевский И.К., Пузикова В.В. Использование метода декомпозиции области для распараллеливания моделирования течения вязкой несжимаемой среды методом LS-STAG и дополнительного преобуславливания. Труды ИСП РАН, том 35, вып. 5, 2023 г., стр. 287–302. DOI: 10.15514/ISPRAS–2023–35(5)–19.

Domain Decomposition Method Usage for Parallelization and Extra Preconditioning of Viscous Incompressible Flow Simulation by Using the LS-STAG Method

¹ I. K. Marchevsky, ORCID: 0000-0003-4899-4828 <iliamarchevsky@mail.ru>

² V.V. Puzikova, ORCID: 0000-0003-0712-4519 <v.puzikova@yadro.com>

¹ *Bauman Moscow State Technical University,
5, 2nd Baumanskaya st., Moscow, 105005, Russia.*

² *YADRO,
15 bld.15, Rochdelskaya st., Moscow, 123022, Russia.*

Abstract. As a rule, the main part of the computational costs in the numerical solution of problems in continuum mechanics consists in the solving of large sparse systems of linear algebraic equations. For this reason, efficient parallelization of this particular procedure can significantly speed up the simulation. To solve this problem, two main approaches can be used. The simplest approach consists in parallelizing of matrix-vector operations in a usual iterative solver. It requires several synchronization points and exchanges of coefficients at each iteration of the solver, which does not significantly speed up the simulation process as a whole. Therefore, domain decomposition methods are preferable. These methods involve dividing the computational domain into subdomains, constructing and solving separate problems in them, as well as some procedure to coordinate the solution between subdomains to ensure global convergence. Subdomains can overlap, as in the Schwartz method used in OpenFOAM, or they can be separated by interface sections, on which their own interface task is built, as in the Schur complement method. The latter method is used in this research to construct a parallel algorithm for viscous incompressible flow simulation by using the immersed boundary method LS-STAG with cut-cells and level-set functions. The resulting matrix of the interface system has a block tridiagonal structure. To speed up prototyping, OpenMP parallel programming technology is used in the software implementation of the developed algorithm, so computational experiments are carried out only on systems with shared memory, in particular on individual nodes of the educational and experimental cluster of the Applied Mathematics Department, Bauman Moscow State Technical University. To verify and evaluate the effectiveness of the developed parallel algorithm, a well-studied test problem about simulation of two-dimensional flow around a stationary circular airfoil is considered. Computations on a sequence of meshes with different numbers of subdomains show that the parallel algorithm allows one to obtain the same numerical solution as the original algorithm of the LS-STAG method, and the computed values of the Strouhal number and drag coefficient are in good agreement with the experimental and computational data known in the literature. Experiments demonstrate that the developed algorithm with domain decomposition allows to accelerate simulation even in sequential mode by reducing the number of solver iterations, i.e. the domain decomposition method acts as an additional preconditioner. Due to this property, the acceleration is superlinear when simulating in parallel mode with developed algorithm. This effect persists up to a certain number of subdomains, which depends on the size of the problem.

Keywords: domain decomposition method; immersed boundary method; the LS-STAG method; viscous incompressible flow; system of linear equations; preconditioner; FGMRES method; parallel computations; OpenMP; OpenFOAM.

For citation: Marchevsky I.K., Puzikova V.V. Domain Decomposition Method Usage for Parallelization and Extra Preconditioning of Viscous Incompressible Flow Simulation by Using the LS-STAG Method. *Trudy ISP RAN/Proc. ISP RAS*, vol. 35, issue 5, 2023. pp. 287-302 (in Russian). DOI: 10.15514/ISPRAS-2023-35(5)-19.

1. Введение

В ходе численного решения задач механики сплошной среды вычислительные ресурсы могут стать ограничивающим фактором при использовании больших и сложных моделей. Распараллеливание решения разреженных систем линейных алгебраических уравнений (СЛАУ), на долю которого, как правило, приходится основная часть вычислительных затрат при проведении моделирования, является одним из основных способов ускорения таких расчетов. Кроме того, использование параллельных методов решения СЛАУ становится обязательным при решении задач на распределенных системах.

Существует несколько подходов к параллельному решению СЛАУ. Во-первых, может использоваться обычная итерационная процедура (например, такие крыловские методы [1] как CG, BiCG, BiCGStab и т.д.) с распределенными вычислениями матрично-векторных произведений или операций над независимыми частями векторов и межпроцессорными обменами в точках вычисления необходимых коэффициентов алгоритма с использованием скалярных произведений [2]. В этом случае, каждый процессор работает только со своим участком расчетной области, но при этом на каждой итерации решения СЛАУ требуется делать несколько точек синхронизации и обменов коэффициентами. С точки зрения реализации этот подход наиболее прост, но требует много коммуникационных затрат. Во-вторых, могут использоваться методы декомпозиции области [3], которые подразумевают решение на каждом процессоре отдельных специальным образом построенных задач в подобластях и некоторую межпроцессорную процедуру согласования решения между этими подобластями с целью обеспечения глобальной сходимости.

Методы декомпозиции области можно разделить на две группы [1]: подходы, основанные на методе дополнения Шура, и подходы, основанные на методе Шварца. В первом случае область разделяется на подобласти без перекрытий, в которых формируются локальные СЛАУ, и интерфейсные границы между ними, для которых составляется собственная СЛАУ. Решение последней позволяет вычислить значения неизвестных на интерфейсных границах и использовать их как условия первого рода для задач в подобластях. Во втором случае (для методов Шварца) область разделяется на подобласти с перекрытиями, решения пересылаются между процессорами, обрабатывающими соседние подобласти. Этот подход требует больше коммуникационных затрат, но является более робастным. Похожий подход используется в пакете OpenFOAM [4] для проведения гидродинамических расчетов с использованием MPI, декомпозиция области при этом строится при помощи таких библиотек, как Scotch [5] и METIS [6], а решение задач в подобластях осуществляется при помощи обычных решателей СЛАУ.

Целью данной работы является разработка, реализация и верификация параллельного алгоритма для моделирования течений вязкой несжимаемой среды методом погруженных границ [7] с усеченными ячейками и функциями уровня [8] LS-STAG [9]. Этот метод развивается с 2010 г. и еще не реализован в крупных библиотеках для решения задач вычислительной гидродинамики, хотя обладает рядом привлекательных особенностей:

- использование прямоугольных разнесенных сеток, не связанных с границами обтекаемых тел, в т.ч. в случае их движения;
- сохранение пятиточечного шаблона дискретизации в 2D и семиточечного в 3D [10];
- корректная аппроксимация уравнений и граничных условий на усеченных ячейках;
- наличие модификаций для моделирования турбулентных течений с использованием подходов RANS/LES/DES [11];
- наличие модификаций для моделирования течений неньютоновских жидкостей [12].

2. Постановка тестовой задачи

Для верификации разработанного алгоритма используется хорошо изученная тестовая задача о моделировании двумерного обтекания неподвижного кругового профиля. Постановка задачи включает в себя уравнение неразрывности и уравнения Навье-Стокса, а также необходимые начальные и граничные условия. Задача представляется в безразмерном виде; в качестве характерной скорости используется скорость набегающего потока, а в качестве характерного размера – диаметр обтекаемого кругового профиля.

Рассматривается расчетная область, представленная на рис. 1. На левой границе расчетной области для скорости стоит граничное условие первого рода – набегающий поток со скоростью $V_\infty = 1$, на остальных границах – стандартное для практики условие выхода потока (равенство нулю нормальной производной скорости).

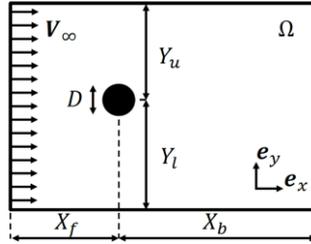


Рис. 1. Расчетная область.
Fig. 1. Computational domain.

В данную расчетную область помещен неподвижный круговой профиль диаметром $D = 1$. Центр профиля находится на расстоянии $X_f = 8$ от левой границы расчетной области и $X_b = 15$ от правой. Верхняя и нижняя границы расчетной области равноотстоят от центра профиля на расстояние $Y_u = Y_l = 12$.

3. Основные идеи метода LS-STAG

В двумерной расчетной области вводится прямоугольная сетка, в общем случае неравномерная, называемая «основной». Относительно нее строятся еще две сетки: «x-сетка», смещенная относительно основной на половину ячейки по горизонтали, и «y-сетка», смещенная относительно основной на половину ячейки по вертикали.

В центрах ячеек основной сетки вычисляются давление и нормальные напряжения, в углах – касательные напряжения. В серединах жидких частей граней ячеек – компоненты скорости, причем горизонтальная составляющая скорости вычисляется в середине восточной грани, а вертикальная – в середине северной. Таким образом, контрольные объемы для построения дискретного аналога уравнения неразрывности совпадают с ячейками основной сетки, ячейки x-сетки являются контрольными объемами для построения дискретного аналога уравнения импульса в проекции на ось Ox , а ячейки y-сетки – контрольными объемами для построения дискретного аналога уравнения импульса в проекции на ось Oy [9].

Для описания положения погруженной границы вводится знакопеременная функция уровня [8], ее значения вычисляются в углах ячеек основной сетки. Функция уровня равна нулю на границе профиля, отрицательна снаружи профиля и положительна внутри него. По четырем значениям функции уровня для каждой ячейки можно вычислить два коэффициента заполнения ячейки – доли восточной и северной грани ячейки, заполненные жидкостью. Эти коэффициенты позволяют однозначно определить тип усеченной ячейки: в 2D случае возможны трапециевидные, треугольные и пятиугольные ячейки; названия отражают форму твердой части ячейки. Коэффициенты заполнения ячеек активно используются при построении дискретных аналогов определяющих соотношений и позволяют добиться

единообразной формы из записи на усеченных ячейках и ячейках, полностью заполненных жидкостью.

Для интегрирования по времени в случае неподвижных погруженных границ используется схема предиктор-корректор второго порядка точности. Шаг предиктора приводит к решению разностного аналога уравнения Гельмгольца для прогноза скоростей, а шаг корректора – к решению разностного аналога уравнения Пуассона для поправки давления. Затем определяются новые значения скоростей и давления.

Таким образом, на каждом шаге по времени решается три СЛАУ:

- для горизонтальных составляющих скоростей на x -сетке;
- для вертикальных составляющих скоростей на y -сетке;
- для функции давления на основной сетке.

Отметим, что третья СЛАУ является плохообусловленной, поскольку возникает при построении дискретного аналога уравнения Пуассона. По этой причине время счета может сильно различаться в зависимости от выбора предобуславливателя.

4. Распараллеливание алгоритма

В работе [10] имеется упоминание о распараллеливании метода LS-STAG, однако никаких деталей алгоритма не представлено. Кроме того, в указанной работе не было выполнено никакого исследования эффективности параллельного алгоритма: рассматривалась только точность решения при проведении расчетов с использованием двух и четырех потоков.

Других работ по построению параллельных алгоритмов для методов усеченных ячеек, в которых построение дискретных аналогов и решение уравнений происходят в том числе и на погруженной границе (на усеченных ячейках), найти не удалось. Однако для классических методов погруженных границ, в которых решение на погруженной границе лишь интерполируется, известен, например, алгоритм [13], основанный на расщеплении системы уравнений по направлениям, сведении к трехдиагональным СЛАУ и их параллельном решении с помощью дополнения Шура. Кроме того, для этих классических методов погруженных границ также известна работа [14], в которой использован алгоритм Эйткена-Шварца.

В данной работе в качестве метода декомпозиции области рассматривается метод дополнения Шура [1, 3]. Он позволяет заменять решение задачи в исходной расчетной области на решение серии независимых задач в подобластях, которые не имеют перекрытий. Интерфейсные условия (граничные условия на границах между подобластями) при этом учитываются неявно. На каждом шаге по времени при использовании данного метода решаются по две задачи для каждой подобласти и одна задача для вычисления неизвестных, соответствующих интерфейсам между соседними подобластями.

4.1 Декомпозиция области

Рассмотрим СЛАУ

$$Ax = b, \quad \det A \neq 0 \quad (1)$$

Значения компонент вектора неизвестных x вычисляются в центрах ячеек введенной в расчетной области Ω сетки. Разделим расчетную область на непересекающиеся подобласти, разделенные интерфейсными рядами контрольных объемов:

$$\Omega = \cup_{p=0}^{P-1} \Omega_p \cup_{i=0}^{P-2} \Gamma_i.$$

Здесь Ω_p – непересекающиеся подобласти, $p = \overline{0, P-1}$; Γ_i – интерфейсные ряды между подобластями, $i = \overline{0, P-2}$. Обозначим число контрольных объемов в интерфейсном ряду как N . Предположим для определенности, что подобласти представляют собой горизонтальные

полосы. Для случая разбиения расчетной области на вертикальные полосы выкладки могут быть проведены аналогично.

В результате такого разбиения вектор неизвестных также разбивается на непересекающиеся подвекторы – векторы неизвестных из подобластей и вектор неизвестных со всех интерфейсных границ. Введем следующие обозначения: x_p – вектор неизвестных из p -й подобласти, $p = \overline{0, P-1}$; x_s – вектор неизвестных со всех интерфейсных границ, пронумерованных подряд слева направо снизу вверх. Теперь перенумеруем контрольные объемы еще раз – сначала во всех подобластях подряд сверху вниз, потом на всех интерфейсных рядах. После такой перенумерации СЛАУ (1) примет следующий блочный вид:

$$\begin{bmatrix} A_{0,0} & \Theta & \cdots & \Theta & A_{0,s} \\ \Theta & A_{1,1} & \cdots & \Theta & A_{1,s} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \Theta & \Theta & \cdots & A_{P-1,P-1} & A_{P-1,s} \\ A_{s,0} & A_{s,1} & \cdots & A_{s,P-1} & A_{s,s} \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{P-1} \\ x_s \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{P-1} \\ b_s \end{bmatrix}. \quad (2)$$

Применяя к СЛАУ (2) блочный метод Гаусса, получим СЛАУ только для интерфейсных неизвестных:

$$\tilde{A}_{s,s} x_s = \tilde{b}_s. \quad (3)$$

Таким образом, решив СЛАУ (3), можно вычислить неизвестные на интерфейсных границах до решения задач в подобластях и использовать полученные значения интерфейсных неизвестных как граничные условия первого рода для задач в подобластях. В результате задачи в подобластях становятся полностью независимыми и могут решаться параллельно. Для расчета правой части \tilde{b}_s интерфейсной задачи (3) также необходимо решить независимые задачи в подобластях. Таким образом, на каждой итерации построенного метода декомпозиции области необходимо решить по две задачи в каждой подобласти и интерфейсную систему (3):

- 1: **for** $p = \overline{0, P-1}$ **do**
- 2: $A_{p,p} t = b_p$
- 3: $\tilde{b}_s^p = A_{s,p} t$
- 4: **end for**
- 5: $\tilde{b}_s = b_s - \sum_{p=0}^{P-1} \tilde{b}_s^p$ (4)
- 6: $\tilde{A}_{s,s} x_s = \tilde{b}_s$
- 7: **for** $p = \overline{0, P-1}$ **do**
- 8: $A_{p,p} x_p = b_p - A_{p,s} x_s$
- 9: **end for**

Для расчета матрицы $\tilde{A}_{s,s}$ интерфейсной задачи (3) необходимо решить задачи в подобластях для каждого столбца матрицы:

- 1: **for** $p = \overline{0, P-1}$ **do**
- 2: **for** $c = \overline{0, N-1}$ **do**
- 3: $A_{p,p} t = [A_{p,s}]_c$
- 4: $[\tilde{A}_{s,s}^p]_c = A_{s,p} t$ (5)
- 5: **end for**
- 6: **end for**
- 7: $\tilde{A}_{s,s} = A_{s,s} - \sum_{p=0}^{P-1} \tilde{A}_{s,s}^p$

Для задач, в которых необходимо перестраивать сетку на каждом шаге расчета, перестроение вместе с ней матрицы $\tilde{A}_{s,s}$ по алгоритму (5) может оказывать серьезное влияние на время

счета. Но для задач с неподвижными погруженными границами процедуру (5) достаточно провести один раз – на этапе построения сетки.

4.2 Вычисление матриц для задач в подобластях и на интерфейсах

Теперь учтем, что в методе LS-STAG в 2D случае используется пятиточечный шаблон (в обозначениях сторон света – S, W, P, E, N), поэтому матрицы $A_{s,s}$, $A_{s,p}$ и $A_{p,s}$ формируются достаточно просто – ненулевые элементы в них появляются только благодаря вкладам от интерфейсных рядов контрольных объемов. В итоге получаем, что элементы W, P, E с интерфейсного ряда Γ_p идут в p -й диагональный блок $A_{s,s}^p$ матрицы $A_{s,s}$; элемент S идет в блоки $A_{s,p}^S$ и $A_{p,s}^N$ матриц $A_{s,p}$ и $A_{p,s}$; элемент N – в блоки $A_{s,p+1}^N$ и $A_{p+1,s}^S$ матриц $A_{s,p+1}$ и $A_{p+1,s}$. Отметим, что для первой и последней подобластей, имеющих интерфейсную границу только с одной стороны, половины перечисленных блоков не существует:

$$A_{s,0}^N = A_{s,P-1}^S = A_{0,s}^S = A_{P-1,s}^N = 0; \quad x_s^{-1} = x_s^{P-1} = \tilde{b}_s^{-1} = \tilde{b}_s^{P-1} = 0;$$

$$\tilde{A}_{s,s}^{-1,k} = \tilde{A}_{s,s}^{P-1,k} = 0; \quad k = \overline{0, P-2}.$$

С учетом описанного вида матриц $A_{s,p}$ и $A_{p,s}$ алгоритм (4) принимает следующий вид:

- 1: $\tilde{b}_s = b_s$
- 2: **for** $p = \overline{0, P-1}$ **do**
- 3: $A_{p,p}t = b_p$
- 4: $\tilde{b}_s^{p-1} -= A_{p,s}^N t^0$
- 5: $\tilde{b}_s^p -= A_{p,s}^S t^{M_p-1}$
- 6: **end for**
- 7: $\tilde{A}_{s,s}x_s = \tilde{b}_s$
- 8: **for** $p = \overline{0, P-1}$ **do**
- 9: $A_{p,p}x_p = b_p - A_{p,s}^S x_s^{p-1} - A_{p,s}^N x_s^p$
- 10: **end for**

Алгоритм (5) расчета матрицы $\tilde{A}_{s,s}$ интерфейсной задачи (3) после подстановки матриц $A_{s,p}$ и $A_{p,s}$ для метода LS-STAG может быть представлен следующим образом:

- 1: $\tilde{A}_{s,s} = A_{s,s}$
- 2: **for** $p = \overline{1, P-1}$ **do**
- 3: **for** $c = \overline{0, N-1}$ **do**
- 4: $A_{p,p}t = [A_{p,s}]_c$
- 5: $[\tilde{A}_{s,s}^{p-1,p-1}]_c -= A_{s,p}^N t^0$
- 6: $[\tilde{A}_{s,s}^{p,p-1}]_c -= A_{s,p}^S t^{M_p-1}$
- 7: **end for**
- 8: **end for**
- 9: **for** $p = \overline{0, P-2}$ **do**
- 10: **for** $c = \overline{0, N-1}$ **do**
- 11: $A_{p,p}t = [A_{p,s}]_{N_d-N+c}$
- 12: $[\tilde{A}_{s,s}^{p-1,p}]_c -= A_{s,p}^N t^0$
- 13: $[\tilde{A}_{s,s}^{p,p}]_c -= A_{s,p}^S t^{M_p-1}$
- 14: **end for**
- 15: **end for**

Таким образом, хотя матрица $A_{s,s}$ в случае метода LS-STAG имеет блочно-диагональную структуру, причем каждый блок является трехдиагональной матрицей (ни одна пара интерфейсных рядов не является соседними и от каждого контрольного объема с интерфейсных рядов в эту матрицу идут только три ненулевых элемента), матрица $\tilde{A}_{s,s}$ интерфейсной задачи (3) имеет более сложную, блочно-трехдиагональную, структуру, причем каждый блок из нее – полностью заполненная матрица.

4.3 Метод решения интерфейсной задачи

Итак, в построенном параллельном алгоритме (6)-(7) для моделирования методом LS-STAG с описанной декомпозицией расчетной области, матрица $\tilde{A}_{s,s}$ интерфейсной СЛАУ (3) имеет блочно-трехдиагональную структуру. Соответственно, в программной реализации для ее хранения используется блочный формат, а не CSR как для всех остальных рассматриваемых матриц.

Заметим, что эту трехдиагональную блочную матрицу $\tilde{A}_{s,s}$ можно представить в виде произведения нижнетреугольной и верхнетреугольной блочных матриц. Такую LU-факторизацию матрицы интерфейсной СЛАУ можно хранить компактно в виде одной трехдиагональной блочной матрицы, так как блоки на главной блочной диагонали нижнетреугольной матрицы из разложения являются единичными матрицами и хранить их нет смысла.

Поскольку для решения интерфейсной СЛАУ (3) на каждом шаге расчета требуется только наличие LU-факторизации матрицы $\tilde{A}_{s,s}$, а сама матрица интерфейсной СЛАУ после построения этой факторизации в расчетах больше не требуется, выделенную изначально для хранения матрицы $\tilde{A}_{s,s}$ область памяти можно переиспользовать для хранения ее LU-факторизации.

Отметим, что при построении LU-факторизации требуется вычисление обратных матриц для блоков, составляющих главную диагональ верхнетреугольной матрицы разложения. Вместо непосредственного обращения этих матриц так же строим их LU-факторизации и сохраняем вместо исходных блоков на главной блочной диагонали.

5. Вычислительные эксперименты

Предложенный параллельный алгоритм реализован в разработанном программном комплексе "LS-STAG_DDM" [15] на языке C++. Для ускорения прототипирования в качестве технологии распараллеливания вычислений использовалась библиотека OpenMP, поэтому в рамках данной работы представлены вычислительные эксперименты, проведенные только на системах с общей памятью. В дальнейшем будет разработана реализация с использованием технологии параллельного программирования MPI, что позволит моделировать течения на системах с распределенной памятью.

В качестве решателя СЛАУ, соответствующих задачам в подобластях, используется итерационный метод FGMRES [1] с ILU-предобуславливателем для уравнения Гельмгольца и с многосеточным [16] предобуславливателем для уравнения Пуассона. Для решения СЛАУ, соответствующей задаче на интерфейсных границах между подобластями, реализован оптимизированный выше блочный прямой решатель.

Верификация и оценка эффективности разработанного параллельного алгоритма проводятся на описанной выше тестовой задаче о моделировании двумерного обтекания неподвижного кругового профиля при значении числа Рейнольдса, равном 200, в течение 100 безразмерных единиц времени. Эксперименты проводились на последовательности из четырех неравномерных сеток (табл. 1). Каждая сетка содержит равномерный блок в окрестности рассматриваемого профиля, причем на диаметр профиля приходится ND ячеек. В качестве примера на рис. 2 показана сетка G_1 .

Табл. 1 Основные характеристики используемых сеток
 Table 1. Main characteristics of used meshes

Сетка	ND	$N \times M$	Δt
G_1	16	120×148	0.0100
G_2	32	240×296	0.0050
G_3	64	480×592	0.0010
G_4	128	960×1184	0.0005

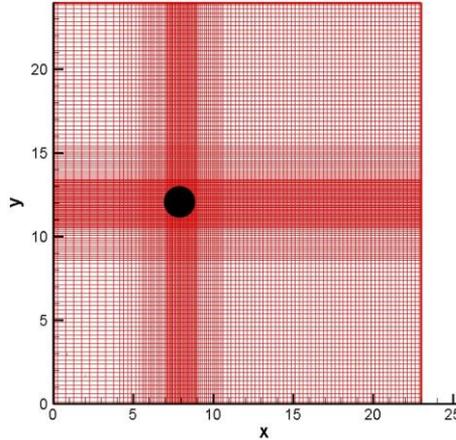


Рис. 2. Сетка G_1 .
 Fig. 2. G_1 mesh.

5.1 Валидация алгоритма

Эксперименты, проведенные на сетках из табл. 1 при разном количестве подобластей, показали, что построенный и реализованный параллельный алгоритм позволяет получить такое же численное решение, что и исходный алгоритм без декомпозиции. При моделировании были рассчитаны нестационарные нагрузки $C_{xa}(t)$ и $C_{ya}(t)$. На рис. 3 и 4 представлены полученные зависимости для рассматриваемой последовательности сеток.

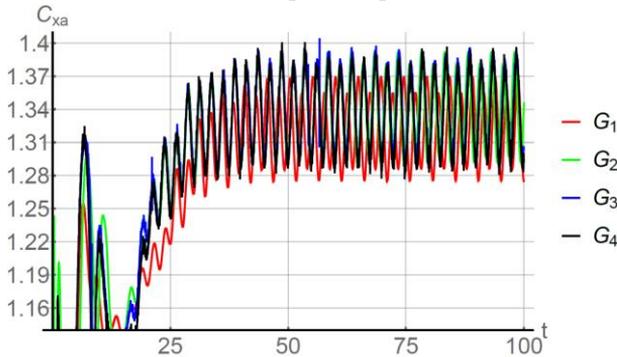


Рис. 3. Нестационарные нагрузки $C_{xa}(t)$, рассчитанные на сетках G_1 и G_2 без деления на подобласти и на сетках G_3 и G_4 с разбиением на 18 подобластей.
 Fig. 3. Unsteady loads $C_{xa}(t)$, computed on G_1 and G_2 meshes without division into subdomains and on G_3 and G_4 meshes with division into 18 subdomains.

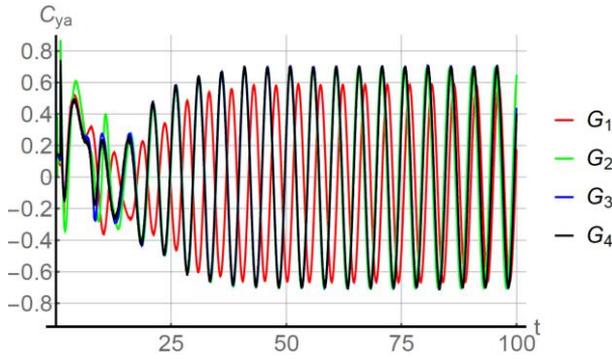


Рис. 4. Нестационарные нагрузки $C_{ya}(t)$, рассчитанные на сетках G_1 и G_2 без деления на подобласти и на сетках G_3 и G_4 с разбиением на 18 подобластей.

Fig. 4. Unsteady loads $C_{ya}(t)$, computed on G_1 and G_2 meshes without division into subdomains and on G_3 and G_4 meshes with division into 18 subdomains.

Видно, что по мере сгущения сетки результаты начинают сходиться несмотря на то, что представленные зависимости для подробных сеток получены при использовании декомпозиции расчетной области на 18 подобластей.

Кроме того, для каждого вычислительного эксперимента было рассчитано значение безразмерного стационарного аэродинамического коэффициента лобового сопротивления C_{xa} (табл. 2). Для этого смоделированные нестационарные нагрузки $C_{xa}(t)$ осреднялись по промежутку от 75 до 100 безразмерных единиц. Кроме того, в этом же временном диапазоне были проанализированы рассчитанные нестационарные зависимости $C_{ya}(t)$ с целью определения числа Струхала Sh .

Табл. 2 Сравнение рассчитанных значений коэффициента лобового сопротивления C_{xa} и числа Струхала Sh с известными в литературе данными

Table 2. Comparison of computed values of the drag coefficient C_{xa} and the Strouhal number Sh with data known in the literature

Источник	C_{xa}	Sh
Расчет на G_1 без декомпозиции	1.322	0.191
Расчет на G_1 , 8 подобластей	1.320	0.194
Расчет на G_2 без декомпозиции	1.341	0.208
Расчет на G_2 , 8 подобластей	1.340	0.208
Расчет на G_3 без декомпозиции	1.342	0.202
Расчет на G_3 , 18 подобластей	1.343	0.202
Расчет на G_4 без декомпозиции	1.336	0.201
Расчет на G_4 , 18 подобластей	1.336	0.201
Cheny & Botella [9], LS-STAG на сетке 550×350	1.327	0.200
Rajani и др. [17], расчет на сетке 386×322	1.316	0.196
Qu, Norberg, Davidson, Peng & Wang [18], расчет	1.337	0.196
Williamson [19], эксперимент	–	0.195
Wieselberger [20], эксперимент	1.330	–

В качестве примера в табл. 2 представлены некоторые из полученных значений $C_{ха}$ и Sh для расчетов на разных сетках с разным числом подобластей. Видно, что по мере сгущения сетки рассчитанные значения коэффициентов перестают отличаться в зависимости от используемого числа подобластей. Хотя даже на самой грубой из используемых сеток эти значения отличаются не более чем на 1.5%, что также можно считать незначительным.

Также в табл. 2 приведены для сравнения известные в литературе экспериментальные и расчетные данные других исследователей. Заметим, что рассчитанные на сетке G_4 при разном числе подобластей значения коэффициента лобового сопротивления отличаются от экспериментально определенного на 0.45%, а значения числа Струхала – на 3.08%. Таким образом, можно сделать вывод о хорошем согласовании смоделированных значений с экспериментальными, как при использовании метода LS-STAG без декомпозиции расчетной области, так и проведении расчетов с разбиением области на подобласти.

5.2 Исследование свойств алгоритма на грубых сетках

Расчеты на сетках G_1 и G_2 проводились на ПК с 8-ядерным процессором AMD Ryzen 7 5800H (Zen 3 с SMT, 16 потоков, 3,2 ГГц, кэш L3 16 Мб, 16 ГБ оперативной памяти DDR4-3200 и 512 ГБ SSD NVMe PCIe). Из графиков на рис. 5 можно сделать вывод, что на сетке G_1 задача недостаточно велика (около 14 рядов контрольных объемов в подобласти при делении на 8 подобластей). Ускорение, полученное на сетке G_2 сравнимо с результатами, представленными в работе [14] для задачи сопоставимого размера (400 × 100 ячеек, 1..8 узлов).

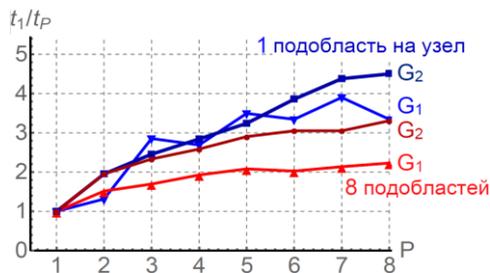


Рис. 5. Ускорение вычислений на грубых сетках.
Fig. 5. Acceleration of computations on coarse meshes.

Необходимо отметить, что разработанный алгоритм с декомпозицией области позволяет ускорить моделирование даже на 1 узле/потоке за счет уменьшения количества итераций, т.е. метод декомпозиции области с дополнением Шура работает как дополнительный предобуславливатель. В случае уравнения Гельмгольца для прогноза скоростей задача во всей расчетной области решается за 1 итерацию для любого числа подобластей больше 1, в то время как для решения задачи без подобластей требуется 2 итерации метода FGMRES с ILU-предобуславливателем. Решение плохообусловленной СЛАУ, возникающей из уравнения Пуассона для функции давления, без подобластей требует около 8-12 итераций метода FGMRES с многосеточным предобуславливанием. Это количество можно сократить всего до 1-2 итераций при оптимальном выборе числа подобластей.

Как показали численные эксперименты на рассматриваемых грубых сетках G_1 и G_2 , в случае расчетов в последовательном режиме наилучшее ускорение достигается при количестве уравнений для задач в подобластях в 2 раза большем, чем для интерфейсной системы. Таким образом, лучше использовать 8 подобластей при расчетах в один поток на сетке G_1 и 16 подобластей на сетке G_2 . Получаемое при этом ускорение составляет 1.55 раз и 2.32 раза соответственно.

5.2 Исследование свойств алгоритма на подробных сетках

Расчеты на сетках G_3 и G_4 проводились на учебно-экспериментальном кластере кафедры «Прикладная математика» МГТУ им. Н.Э. Баумана. Кластер состоит из управляющего узла и семи вычислительных. Управляющий узел кластера имеет два четырехядерных процессора Intel Xeon E5620 и оборудован 48 Гб ОЗУ. Он оснащен жестким диском 2 Тб (для собственных нужд) и RAID-массивом объемом 12 Тб для данных пользователей, тремя сетевыми контроллерами (Gigabit Ethernet – для связи с внешним миром; 10-Гб Ethernet – для связи с узлами кластера; Mellanox InfiniBand FDR, 2x54 Гбит/с – основной интерконнект), а также видеокартой Nvidia Geforce GTX 970. Каждый вычислительный узел имеет 18-ядерный процессор Intel Core i9-10980XE, 128 Гб ОЗУ, видеоускоритель Nvidia Titan V и два сетевых интерфейса: 10-Гб Ethernet и Mellanox InfiniBand (FDR, 2x54 Гбит/с). На тесте Linpack данный кластер показал для CPU максимальную производительность 8.82 Тфлопс, составляющую около 97 % от пиковой.

Для каждой рассматриваемой в этом разделе сетки расчеты проводились на одном вычислительном узле кластера с использованием числа ядер (потоков OpenMP), равным числу подобластей. Серия экспериментов для каждой сетки состояла из 18 расчетов: расчета методом LS-STAG без декомпозиции в последовательном режиме и расчетов с декомпозицией на 2..18 подобластей. Полученное благодаря декомпозиции ускорение представлено на рис. 6.

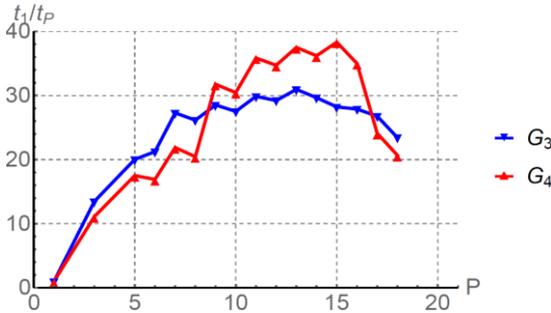


Рис. 6. Ускорение вычислений на подробных сетках.
Fig. 6. Acceleration of computations on fine meshes.

В первую очередь, из рис. 6 видно, что при любом рассматриваемом числе подобластей получаемое ускорение превышает число используемых для распараллеливания потоков OpenMP. Такое сверхлинейное ускорение объясняется хорошими предобуславливающими свойствами метода декомпозиции области с дополнением Шура, которые уже обсуждались выше при рассмотрении вычислительных экспериментов на грубых сетках. Наилучшее ускорение на сетке G_3 составляет 30.9 раз и достигается при использовании 13 подобластей (и потоков), а на сетке G_4 – 38.3 раза при использовании 15 подобластей.

Кроме того, видно, что при многих четных значениях числа подобластей ускорение меньше, чем при соседнем нечетном. Это объясняется тем, что при выбранном способе разбиения на подобласти одна из интерфейсных границ в первом случае будет проходить через ячейки, полностью занятые твердым телом, как показано на рис. 7 на примере разбиения расчетной области на две подобласти. Это делает интерфейсную СЛАУ плохообусловленной. Когда подобластей и, соответственно, интерфейсных границ, немного, численные ошибки при решении такой плохообусловленной системы могут быть особенно критичны – так, например, для случая двух и четырех подобластей на сетках G_3 и G_4 итерационный процесс в итоге перестал сходиться. Вероятно, данная проблема может быть решена заменой описанного выше прямого решателя интерфейсной СЛАУ на более сложный вариант, лучше

подходящий для решения плохообусловленных систем. Этот вопрос требует дальнейшего исследования.

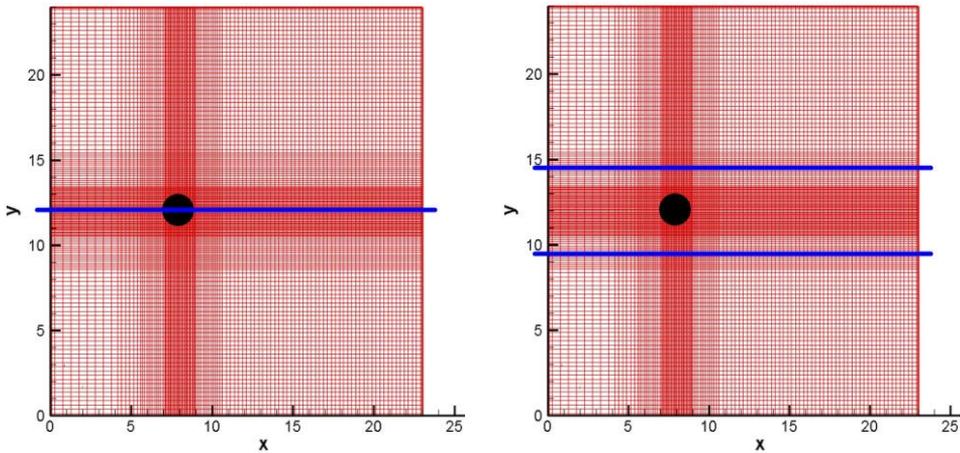


Рис. 7. Разделение сетки G_1 на 2 (слева) и 3 (справа) подобласти.

Fig. 7. The G_1 mesh division into 2 (left) and 3 (right) subdomains.

6. Заключение

С использованием метода декомпозиции области, в частности метода дополнения Шура, построен параллельный алгоритм для моделирования течений вязкой несжимаемой среды методом погруженных границ LS-STAG с усеченными ячейками и функциями уровня. Представлен алгоритм построения задач в подобластях и задачи на интерфейсных границах. Для ускорения прототипирования программная реализация описанного метода разработана с использованием технологии параллельного программирования OpenMP, поэтому вычислительные эксперименты проводились только на системах с общей памятью, в частности на отдельных узлах учебно-экспериментального кластера кафедры «Прикладная математика» МГТУ им. Н.Э. Баумана. В дальнейшем планируется реализовать в разрабатываемом программном комплексе распределенное хранение матриц и векторов и использовать для распараллеливания решения задач в подобластях технологию MPI.

Верификация разработанного метода проводилась на хорошо изученной тестовой задаче об обтекании неподвижного кругового профиля при значении числа Рейнольдса, равном 200. На рассматриваемой последовательности сеток предлагаемый параллельный алгоритм позволяет получить такое же численное решение, что и исходный алгоритм метода LS-STAG без декомпозиции расчетной области. Кроме того, рассчитанные значения коэффициенты лобового сопротивления и числа Струхалия во всех проведенных вычислительных экспериментах хорошо согласуются с известными в литературе экспериментальными и расчетными данными других исследователей.

Как показали проведенные расчеты на грубых сетках, метод декомпозиции области работает также как дополнительный предобуславливатель: даже в последовательном режиме декомпозиция области позволяет ускорить моделирование за счет уменьшения необходимо для достижения заданной точности числа итераций решателя СЛАУ. Например, плохообусловленную СЛАУ, возникающую из уравнения Пуассона для функции давления, удастся решить в среднем за 1-2 итерации в каждой подобласти, в то время как в варианте без декомпозиции расчетной области ее решение занимает от 8 до 12 итераций метода FGMRES с многосеточным предобуславливанием.

Благодаря таким хорошим предобуславливающим свойствам используемого метода декомпозиции области в вычислительных экспериментах на подробных сетках было получено сверхлинейное ускорение. В дальнейшем планируется доработать решатель задачи на интерфейсных границах, чтобы снизить влияние обусловленности интерфейсной задачи на сходимость метода во всей расчетной области.

Список литературы / References

- [1]. Saad Y., *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, 2003. 184 p.
- [2]. Cools S., Vanroose W. The communication-hiding pipelined BiCGstab method for the parallel solution of large unsymmetric linear systems. *Parallel Computing*, 2017, vol. 65, pp. 1-20. DOI: 10.1016/j.parco.2017.04.005
- [3]. Quarteroni A., Valli A. *Domain Decomposition Methods for Partial Differential Equations*. Oxford, 1999. 376 p.
- [4]. OpenFOAM: API Guide: OpenFOAM®: Open source CFD: API, Available at: <https://www.openfoam.com/documentation/guides/latest/api/index.html>, accessed 23.10.2023.
- [5]. Scotch: a software package for graph and mesh/hypergraph partitioning, graph clustering, and sparse matrix ordering, Available at: <https://gitlab.inria.fr/scotch/scotch>, accessed 23.10.2023.
- [6]. METIS, Available at: <https://github.com/KarypisLab/METIS>, accessed 23.10.2023.
- [7]. Mittal R., Iaccarino G. Immersed boundary methods. *Annu. Rev. Fluid Mech*, 2005, vol. 37, pp. 239–261. DOI: 10.1146/annurev.fluid.37.061903.175743
- [8]. Osher S. J., Fedkiw R. *Level set methods and dynamic implicit surfaces*. Springer, 2003. 286 p.
- [9]. Cheny Y., Botella O. The LS-STAG method: A new immersed boundary/level-set method for the computation of incompressible viscous flows in complex moving geometries with good conservation properties. *J. Comput. Phys.*, 2010, vol. 229, pp. 1043-1076. DOI: 10.1016/j.jcp.2009.10.007
- [10]. Nikfarjam F., Cheny Y., Botella O. The LS-STAG immersed boundary / cut-cell method for non-Newtonian flows in 3D extruded geometries. *Comp. Phys. Comm.*, 2018, vol. 226, pp. 67-80. DOI: 10.1016/j.cpc.2018.01.006
- [11]. Marchevsky I.K., Puzikova V.V. Modification of the LS-STAG immersed boundary method for simulating turbulent flows. *Herald of the Bauman Moscow State Technical University. Natural Sciences*, 2017, № 5, pp. 19-34. DOI: 10.18698/1812-3368-2017-5-19-34
- [12]. Marchevsky I.K., Puzikova V.V. The modified LS-STAG method application for planar viscoelastic flow computation in a 4:1 contraction channel. *Herald of the Bauman Moscow State Technical University. Natural Sciences*, 2021, № 3, pp. 46-63. DOI: 10.18698/1812-3368-2021-3-46-63
- [13]. Pacull F., Garbey M. A parallel immersed boundary method for blood-like suspension flow simulations. *Lect. Notes in Comp. Sci. and Eng.*, 2010, vol. 74, pp. 1-8. DOI: 10.1007/978-3-642-14438-7_16
- [14]. Wiens J. K., Stockie J. M. An efficient parallel immersed boundary algorithm using a pseudo-compressible fluid solver. *J. Comp. Phys.*, 2015, vol. 281, pp. 917-941. DOI: 10.1016/j.jcp.2014.10.058
- [15]. Пузикова В. В. Параллельный программный комплекс LS-STAG_DDM для моделирования методом LS-STAG с декомпозицией области; свидетельство о государственной регистрации программы для ЭВМ № 2023665707, зарегистрировано в Реестре программ для ЭВМ 19.07.23 / Puzikova V. V. Parallel software package LS-STAG_DDM for simulation by using the LS-STAG method with domain decomposition; Certificate of state registration of a computer program No. 2023665707, registered in the Register of Computer Programs on 07/19/23 (in Russian).
- [16]. Wesseling P. *An introduction to multigrid methods*. John Wiley & Sons Ltd., 1991. 296 p.
- [17]. Rajani B. N., Kandasamy A., Majumdar S. Numerical simulation of laminar flow past a circular cylinder. *Applied Mathematical Modeling*, 2009, vol. 33, pp. 1228-1247. DOI: 10.1016/j.apm.2008.01.017
- [18]. Qu L., Norberg C., Davidson L., Peng S.-H., Wang F. Quantitative numerical analysis of flow past a circular cylinder at Reynolds number between 50 and 200. *J. Fluids Struct.*, 2013, vol. 39, pp. 347-370. DOI: 10.1016/j.jfluidstructs.2013.02.007
- [19]. Williamson C. H. K. Vortex dynamics in the cylinder wake. *Annu. Rev. Fluid. Mech.*, 1996, vol. 28, pp. 477-539. DOI: 10.1146/annurev.fl.28.010196.002401
- [20]. Wieselberger C. New data on the laws of fluid resistance. *Technical Notes. National Advisory Committee for Aeronautics*, 1923, № 84, pp. 1-16.

Информация об авторах / Information about authors

Илья Константинович МАРЧЕВСКИЙ – доктор физико-математических наук, доцент, профессор кафедры «Прикладная математика» МГТУ им. Н.Э. Баумана. Сфера научных интересов: вычислительная гидродинамика, вихревые методы, теория устойчивости, численные методы, высокопроизводительные вычисления, элементарная математика.

Ilya Konstantinovich MARCHEVSKY – Dr. Sci. (Phys.-Math.), Associate professor, Professor of Applied Mathematics department, Bauman Moscow State Technical University. Research interests: computational fluid dynamics, vortex particle methods, theory of stability, numerical methods, high performance computing, elementary mathematics.

Валерия Валентиновна ПУЗИКОВА – кандидат физико-математических наук, эксперт по разработке программного обеспечения Департамента разработки высокопроизводительных библиотек компании YADRO. Сфера научных интересов: решатели и преобуславливатели СЛАУ, разработка прикладных математических программ, вычислительная гидродинамика, физические движки для AR/VR, высокопроизводительные вычисления, численные методы.

Valeria Valentinovna PUZIKOVA – Cand. Sci. (Phys.-Math.), Software Development Expert in High Performance Libraries Department, YADRO. Research interests: solvers and preconditioners for SLAE, applied mathematics software development, computational hydrodynamics, physics engines for AR/VR, high performance computations, numerical methods.

