

УДК 519.6

**РЕАЛИЗАЦИЯ МЕТОДА РАСЧЕТА
ВЯЗКОЙ НЕСЖИМАЕМОЙ ЖИДКОСТИ
С ИСПОЛЬЗОВАНИЕМ МНОГОСЕТОЧНОГО МЕТОДА
НА ОСНОВЕ АЛГОРИТМА SIMPLE В ПАКЕТЕ ПРОГРАММ ЛОГОС**

А. С. Козелков, Ю. Н. Дерюгин, С. В. Лашкин, Д. П. Силаев,
П. Г. Симонов, Е. С. Тятюшкина
(РФЯЦ-ВНИИЭФ, г. Саров)

Рассматриваются особенности программной реализации многосеточного метода решения систем линейных алгебраических уравнений с агрегативным способом огрубления. Реализация метода выполнена в пакете программ ЛОГОС для модуля, ориентированного на численное решение уравнений гидродинамики вязкой несжимаемой жидкости на произвольных неструктурированных сетках. Рассматриваются варианты решения скалярных систем линейных алгебраических уравнений с использованием V -, W - и F -циклов многосеточного метода. Особое внимание уделяется распараллеливанию метода, показано решение *проблемы грубых уровней*. Приводится сравнение эффективности, полученной при расчетах течений вязкой несжимаемой жидкости с применением алгебраического многосеточного метода и предобусловленного метода сопряженных градиентов, а также многосеточного метода без использования глобального уровня.

Ключевые слова: система уравнений, многосеточный метод, алгоритм, матрица, метод агрегации.

Введение

Течения жидкостей и газов описываются краевыми задачами для системы уравнений Навье—Стокса, представляющей собой нелинейные дифференциальные уравнения в частных производных. Проблема дискретизации этих уравнений, а также их численного решения является одной из ключевых в математическом моделировании.

Использование явных разностных схем для дискретизации уравнений Навье—Стокса во многих случаях приводит к жестким ограничениям на шаг интегрирования по времени, что требуется для сохранения устойчивости вычислительной процедуры, и к большому общему времени счета. Применение неявных схем снимает ограничения на шаг интегрирования по времени, который выбирается исходя из необходимой точности вычислений. Однако трудоемкость решения системы линейных уравнений большого размера, порожденной дискретизацией, в некоторых случаях делает невыгодным использование неявных схем. Другим обстоятельством, препятствующим их широкому использованию в вычислительной практике, является трудность построения параллельных итерационных решателей для систем линейных алгебраических уравнений.

Неявная дискретизация уравнений Навье—Стокса порождает систему линейных уравнений, которая решается тем или иным итерационным методом. Выбор подходящего итерационного метода и его реализация во многом определяют общую эффективность вычислительного алгоритма. На практике используются классические итерационные методы, такие как методы сопряженных и би-сопряженных градиентов, метод глобальной минимизации невязки и др. [1].

В большинстве практических задач дискретизация уравнений Навье—Стокса дает систему линейных алгебраических уравнений с плохо обусловленной матрицей. Такая ситуация зачастую имеет место при дискретизации уравнений на неструктурированных сетках, состоящих из многогранников произвольной формы. При этом классические итерационные методы либо "перестают работать", либо имеют очень медленную скорость сходимости [2].

Одним из универсальных методов решения систем разностных уравнений является многосеточный метод Федоренко, основанный на использовании последовательности вложенных сеток и операторов перехода от одной сетки к другой [3, 4]. В этом методе процесс решения начинается с самой грубой сетки. Полученное на ней решение интерполируется на подробную сетку и используется в качестве начального приближения в каком-либо итерационном процессе, что требует сравнительно небольшого числа итераций для достижения заданной точности. При этом учитывается свойство некоторых итерационных методов (например метода Зейделя) сходиться с высокой скоростью на нескольких итерациях за счет быстрого подавления высокочастотных компонент Фурье [5]. Низкочастотные гармоники сходятся медленнее и составляют основную часть ошибки.

Алгебраические многосеточные методы обычно разделяются на классические селективные методы [6] и методы на основе агрегации [7]. Производительность различных *параллельных* методов агрегации исследуется в работе [8] применительно к решению задач газовой динамики на основе алгоритма SIMPLE [9]. Многосеточный метод использует V-цикл и процедуру Зейделя для сглаживания. Параллельная реализация алгебраического многосеточного метода для решения системы линейных уравнений, порожденной конечно-объемной дискретизацией уравнений Навье—Стокса на неструктурированных сетках, рассматривается в работе [10]. Подходы к параллелизации алгебраических многосеточных методов, использующих различные способы агрегации, сравниваются в работе [11]. В литературе описывается достаточно большое число реализаций алгебраических многосеточных методов, многие из которых, например BoomerAMG [12], имеют открытый исходный код. Обзор алгебраических многосеточных методов приводится в работе [13].

В данной работе представлены особенности реализации алгебраического многосеточного метода с агрегативным способом огрубления. Вариации метода, рассмотренные в [2, 6—8, 10—13], невозможно эффективно применить и реализовать в пакете программ ЛОГОС¹ ввиду его модели памяти, ориентированной на *грани-ячеечное* хранение данных, и механизмов распараллеливания, необходимых для эффективной реализации метода конечных объемов для решения уравнений гидродинамики [9]. Практически все внешние решатели реализуют свою модель памяти, что требует создания дополнительных интерфейсов для доступа к элементам матрицы, а также используют "внутренние" алгоритмы межпроцессорного взаимодействия, не позволяющие эффективно использовать грани-ячеечный формат данных, что негативно сказывается на времени счета, увеличивая его в несколько раз.

В данной работе описаны алгоритмы адаптации многосеточного метода к структуре памяти пакета программ ЛОГОС и его механизмам распараллеливания. Кроме того, предлагается решение *проблемы грубых уровней*, заключающейся в том, что из-за небольшого размера матриц на уровнях с грубыми сетками время, затрачиваемое на обмены, заметно превышает время на вычисления. Для минимизации временных затрат реализован сбор матриц грубого уровня на одном процессе, где происходит формирование *глобального уровня* и дальнейшее продолжение огрубления и решения систем линейных алгебраических уравнений (СЛАУ) в последовательном режиме.

Уравнения для описания течений вязкой несжимаемой жидкости и их дискретизация

Для решения задач вычислительной гидродинамики применяются методы, основанные на решении полной системы уравнений Навье—Стокса и методе расщепления. Одним из наиболее использу-

¹Пакет программ ЛОГОС является разработкой ФГУП "РФЯЦ-ВНИИЭФ" и предназначен для решения сопряженных трехмерных задач конвективного теплопереноса, аэродинамики, гидродинамики и прочности на параллельных ЭВМ [14, 15]. ЛОГОС позволяет рассчитывать течения вязкого сжимаемого и несжимаемого газа с учетом турбулентного перемешивания, процессы распространения тепла в твердом теле, течения в анизотропных пористых средах и другие процессы.

емых на практике методов расщепления является метод SIMPLE, основанный на методе конечных объемов и итерационной процедуре коррекции давления [9, 16–18].

В результате применения метода конечных объемов дискретный аналог системы уравнений Навье–Стокса относительно неизвестной компоненты скорости u_i записывается в виде [16–19]

$$\begin{aligned} \sum_{f=1}^{nb(P)} m_f &= 0; \\ \sum_f (\rho \mathbf{u} u_i)_f \mathbf{S}_f &= \sum_f (\Gamma \nabla u_i)_f \mathbf{S}_f + (Q)_P V, \end{aligned} \quad (1)$$

где $nb(P)$ — количество граней рассматриваемой ячейки P (рис. 1); m_f — массовый поток через грань f ; $\mathbf{u} = \{u_i\}$ — вектор скорости, ρ — плотность; Γ — эффективный коэффициент диффузионного переноса; Q — источник; V — объем ячейки; \mathbf{S}_f — вектор площади грани. Здесь и далее нижние индексы P, N соответствуют значениям в центрах контрольных объемов P и N ; индекс f — значениям на соединяющей их грани. Вектор нормали грани направлен от центра рассматриваемого контрольного объема P к N (см. рис. 1).

Применение метода SIMPLE для решения системы уравнений (1) подразумевает введение нескольких этапов [16–18]. В результате решения системы уравнений на этапе *предиктор*

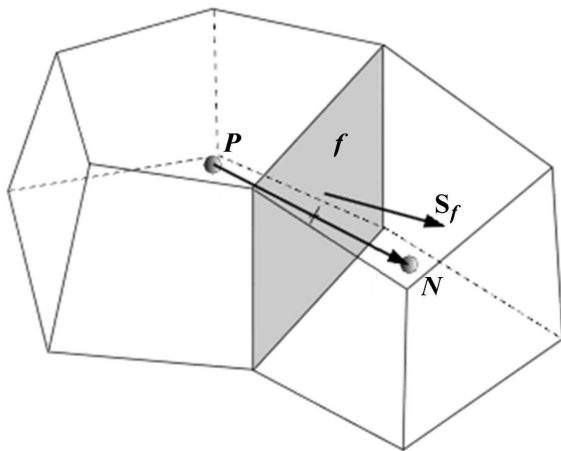


Рис. 1. Контрольные объемы P и N

$$a_P (u_i)_P^* + \sum_N a_N (u_i)_N^* = -V_P \left(\frac{\partial p}{\partial x_i} \right)_P$$

находятся предварительные значения компонент вектора скорости (u_i^*) . В ячейке P

$$(u_i)_P^* = \frac{-\sum_N a_N (u_i)_N^*}{a_P} - \frac{V_P}{a_P} \left(\frac{\partial p}{\partial x_i} \right)_P,$$

или

$$(u_i)_P^* = (\tilde{u}_i)_P^* - \frac{V_P}{a_P} \left(\frac{\partial p}{\partial x_i} \right)_P,$$

где введено обозначение

$$(\tilde{u}_i)_P^* = \frac{-\sum_N a_N (u_i)_N^*}{a_P}.$$

Определенные таким способом предварительные значения компонент вектора скорости не удовлетворяют уравнению неразрывности. Если их подставить в разностный аналог уравнения неразрывности, можно получить следующее выражение:

$$\sum_{f=1}^{nb(P)} \dot{m}_f^* = \delta Q_\rho,$$

где δQ_ρ — результирующий дисбаланс, который устраняется подправкой скорости на этапе *корректор*.

Соотношение, связывающее скорректированные компоненты вектора скорости со скорректированным давлением, запишем в виде

$$(u_i)_P^{n+1} = (\tilde{u}_i)_P^* - \frac{V_P}{a_P} \left(\frac{\partial p}{\partial x_i} \right)_P^{n+1}.$$

В методе SIMPLE для определения давления выводится уравнение Пуассона из предположения выполнения уравнения неразрывности для скорректированных компонент вектора скорости:

$$\operatorname{div} \left(\rho \frac{V_P}{a_P} (\operatorname{grad} p)_P^{n+1} \right) = \operatorname{div} \left(\rho \tilde{u} \right)_P^*.$$

Для обеспечения единственности решения уравнения Пуассона, когда для него заданы только граничные условия второго рода, необходимо зафиксировать значение давления в любой точке либо применить непосредственно граничное условие, которое в общем случае ухудшает обусловленность матрицы. В результате решения находится поправка поля давления в ячейках сетки. По этому полю восстанавливаются компоненты вектора градиента давления в ячейках сетки.

Таким образом, в результате применения метода SIMPLE получается четыре скалярных СЛАУ: три системы для каждой из компонент скоростей и одна для коррекции давления.

Практика показывает, что наиболее затратным по времени является решение матрицы давления: при использовании классических итерационных методов оно занимает около 90 % вычислительного времени расчетного шага, что требует аккуратной и оптимальной реализации даже многосеточного метода.

Матрица СЛАУ для давления симметрична и в большинстве случаев не имеет строгого диагонального преобладания из-за применяемых граничных условий. При использовании выходной границы *статическое давление* число обусловленности матрицы имеет порядок 10^7 – 10^8 [20, 21], что делает ее довольно трудной для решения. Если же использовать выходную границу *свободный выход*, то данное число может достигать порядка 10^{10} и более. Матрицы остальных СЛАУ — для компонент скорости, температуры и параметров турбулентности — имеют числа обусловленности порядка 10^1 – 10^2 , что позволяет решать их классическими методами, такими как метод сопряженных градиентов, метод Зейделя и др.

В приведенных ниже тестах число обусловленности для матрицы давления имеет порядок 10^7 , и практика показывает, что для задач гидродинамики на произвольных неструктурированных сетках это число всегда находится в пределах 10^5 – 10^{10} . Наиболее подходящим методом для решения таких СЛАУ является алгебраический многосеточный метод.

В приведенных выше дискретных соотношениях, в том числе в правой части уравнения Пуассона для давления, основной вычисляемой величиной является поток через грань. В связи с этим для численной реализации этих уравнений наиболее удобно и эффективно использовать грани-ячеечный формат представления и расчета сеточных данных.

Грани-ячеечный формат служит для реализации модели памяти относительно граней (см. рис. 1), где грань f разделяет две ячейки и задает вектор единичной нормали, направленный от ячейки P к ячейке N . При построении численной схемы вычисление потоков через внутреннюю грань осуществляется всегда с использованием двух ячеек, вне зависимости от того, сколько на самом деле ячеек прилегает к исходной грани. Внутренняя грань всегда разделяет две ячейки, и локальный индекс ячейки P строго меньше индекса ячейки N . Благодаря такой структуре заполнение матрицы осуществляется относительно грани и за один цикл формируются сразу две строки матрицы: P и N . В случае же ячеечного представления сеточных данных за один цикл можно сформировать только одну строку матрицы.

Реализация многосеточного метода для скалярных систем

Кратко познакомимся с организацией вычислений с помощью многосеточного метода в соответствии с [2, 3, 5–7, 22]. Рассмотрим систему уравнений

$$A_h x^h = b^h.$$

Оператор интерполяции P с грубой сетки H на подробную сетку h позволяет представить оператор A_H на грубой сетке в виде

$$A_H = R A_h P,$$

где $R = P^T$. Шаг коррекции решения имеет вид

$$x_{new}^h = x_{old}^h + Pe^H.$$

Коррекция e^H является точным решением уравнения

$$A_H e^H = r^H,$$

где $r^H = Rr^h$; $r^h = b^h - A_h x_{old}^h$.

Таким образом, многосеточный метод, использующий схему коррекции решения, представляет собой следующую последовательность шагов:

1. Делается μ_1 приближений решения на сетке h при помощи метода Зейделя (предварительное сглаживание).
2. Невязка $r^h = b^h - A_h x_{old}^h \in V_h$ проецируется на пространство V_H , т. е. $r^H = Rr^h$.
3. Находится приближенное решение $A_H e^H = r^H$ на грубой сетке. Для этого рекурсивно выполняется γ циклов многосеточного метода.
4. Коррекция e^H интерполируется на подробную сетку и производится уточнение решения: $x_{new}^h = x_{old}^h + Pe^H$.
5. Делается μ_2 приближений решения на подробной сетке для подавления ошибки интерполяции (заключительное сглаживание).

В зависимости от числа γ рекурсивных вызовов метода на каждом сеточном уровне выделяют различные типы циклов.

При $\gamma = 1$ имеет место V -цикл, при $\gamma = 2$ — W -цикл (рис. 2, *a*).

Если на каждом уровне рекурсивно сначала вызывать один W -цикл, а затем V -цикл, получим F -цикл (рис. 2, *б*).

В агрегативном методе огрубления с постоянной интерполяцией [23] все переменные разделяются на агрегаты I_k , содержащие все индексы i , соответствующие ячейкам, которые включены в агрегат k .

Построение оператора на грубой сетке производится при помощи соотношения

$$A_H = RA_h P = \left\{ a_{kl}^H \right\}, \quad a_{kl}^H = \sum_{i \in I_k} \sum_{j \in I_l} a_{ij}^h \quad (k, l \in C).$$

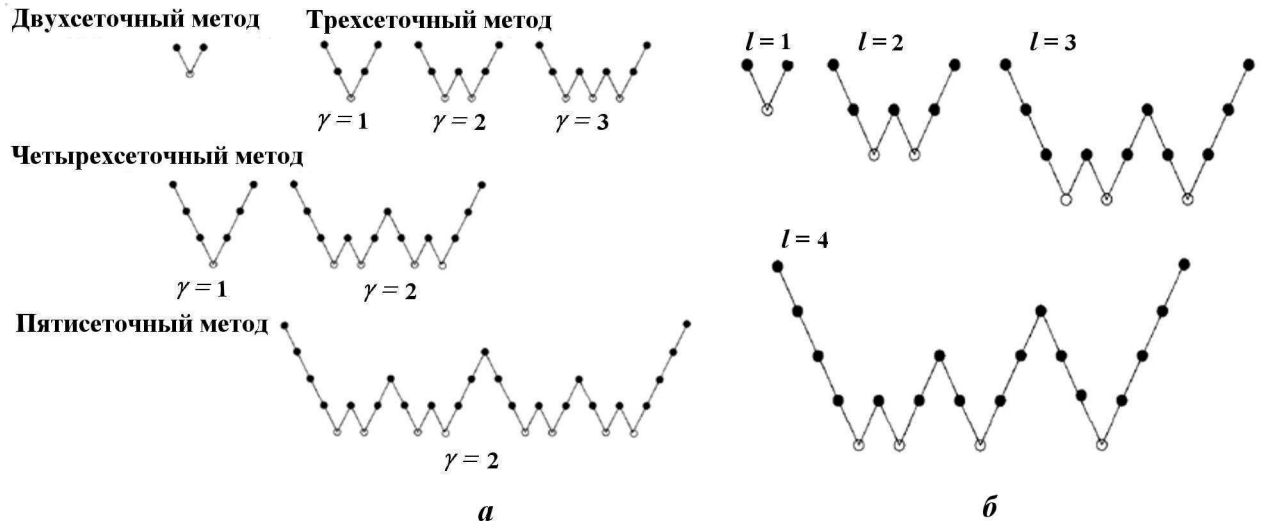


Рис. 2. Типы циклов: *a* — V - и W -циклы; *б* — F -цикл

Рассмотрим сетку, огрубление которой показано на рис. 3. Вычисление матрицы грубого уровня в данном случае будет происходить следующим образом:

$$A_H = RA_hP = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ 0 & a_{32} & a_{33} & a_{34} \\ 0 & 0 & a_{43} & a_{44} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} =$$

$$= \begin{pmatrix} a_{11} + a_{12} + a_{21} + a_{22} & a_{23} \\ a_{32} & a_{33} + a_{34} + a_{43} + a_{44} \end{pmatrix}.$$

Для хранения матрицы использован грани-ячеечный формат. Отдельно хранятся диагональ D , верхнетреугольная U и нижнетреугольная L части матрицы. Соответственно $A = L + D + U$. Три массива, в которых хранятся D , U и L , являются одномерными. Массив, хранящий диагональ, индексируется номером соответствующей ячейки. Два других массива индексируются номерами граней, определяющих значения соответствующих коэффициентов. Направления упорядочения массивов представлены на рис. 4.

В данной работе в качестве *сглаживателя* используется метод Зейделя [20], обладающий признанным сглаживающим оператором [21]

$$x_i^{k+1} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{k+1} - \sum_{j=i+1}^N a_{ij}x_j^k}{a_{ii}}, \quad i \in 1, \dots, N, \quad (2)$$

где N — число ячеек; i — номер строки; j — номер столбца; k — номер шага сглаживания; x_j^k — j -я компонента вектора решения на k -м шаге сглаживания. Представление метода в виде (2) не подходит для эффективной реализации, так как требует выборки элементов из массива нижнетреугольной матрицы не в порядке возрастания индексов (см. рис. 4). Доступ к данным в произвольном порядке существенно медленнее последовательного доступа из-за аппаратных особенностей функционирования оперативной памяти. Добиться строго последовательного доступа к элементам массивов хранения коэффициентов матрицы можно, преобразовав данный алгоритм к виду

$$x_i^{k+1} = \frac{\tilde{b}_i - \sum_{j=i+1}^N a_{ij}x_j^k}{a_{ii}}; \quad \tilde{b}_j = \tilde{b}_j - a_{ji}x_i; \quad i \in 1, \dots, N; \quad j \in i + 1, \dots, N,$$

где $\tilde{\mathbf{b}} = \{\tilde{b}_k\} (k = \overline{1, N})$ — копия вектора \mathbf{b} , подвергающаяся модификации в процессе выполнения алгоритма.

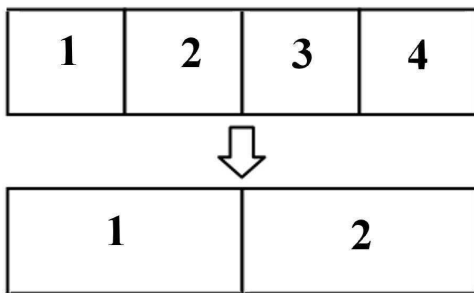


Рис. 3. Пример огрубления сетки

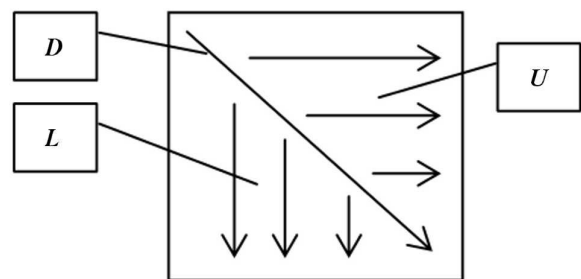


Рис. 4. Расположение матрицы в памяти

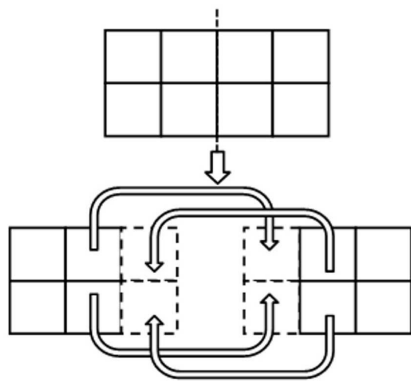


Рис. 5. Передача информации в фиктивные ячейки

В алгоритме распараллеливания пакета программ ЛОГОС вводится понятие фиктивных ячеек, которые представляют собой временные копии реальных счетных ячеек и используются для удобства реализации счетных параллельных алгоритмов. Это вызвано тем, что для расчета требуется информация, относящаяся ко всем соседним ячейкам, в то время как доступ к ней затруднен, поскольку реальные счетные ячейки могут размещаться на других MPI-процессах. На рис. 5 показана передача информации при применении данного подхода.

При распараллеливании огрубление происходит независимо на каждом MPI-процессе. Фиктивные ячейки огрубляются в соответствии с огрублением их действительных прообразов на соответствующих процессах. Таким образом, в ходе огрубления число связей между процессами уменьшается (рис. 6, а).

Однако данный подход к распараллеливанию огрубления порождает две проблемы. Во-первых, огрубление прекращается, если на каждом процессе осталось по одной действительной ячейке. Во-вторых, на грубых уровнях, где размерность матриц невелика, время, затрачиваемое на межпроцессорные обмены, из-за латентности коммуникационной среды начинает многократно превышать время, затрачиваемое на вычисления.

Для решения этих проблем предлагается выполнять сбор всех матриц небольшого размера на одном процессе, формируя глобальный уровень, и продолжать огрубление и решение в последовательном режиме (рис. 6, б).

Таким образом, использование глобального уровня позволяет не только полностью завершить процесс огрубления в параллельном случае, но и, ввиду хранения информации глобального уровня на одном процессе, избежать накладных расходов на избыточные межпроцессорные обмены при обработке уровней с самыми грубыми сетками.

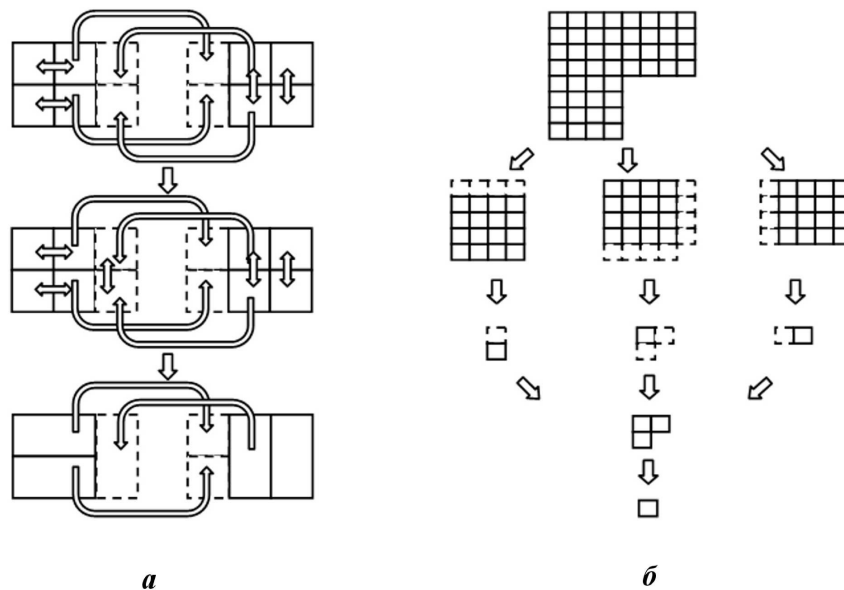


Рис. 6. Огрубление ячеек в параллельном режиме (а) и формирование глобального уровня (б)

Анализ применения многосеточных технологий для расчета течений вязкой несжимаемой жидкости

Для анализа эффективности применения описанной реализации² рассматривается численное решение типичных задач динамики вязкой несжимаемой жидкости.

В этих задачах рассчитываются стационарное изотермическое турбулентное течение вязкой несжимаемой жидкости в прямолинейной трубе с круглым сечением, соответствующее числу Рейнольдса 10 000, и течение за обратным уступом [24].

Данные задачи требуют решения:

- трех СЛАУ для сохранения импульса (матриц скоростей);
- одной СЛАУ относительно коррекции давления (матрицы давления);
- двух СЛАУ для турбулентных параметров (матриц турбулентности).

Матрицы этих СЛАУ сильно различаются по своим свойствам и требуют экспериментального подбора методов их решения. Все матрицы, кроме матрицы давления, являются несимметричными и имеют строгое диагональное преобладание. Матрица давления является симметричной и не имеет строгого диагонального преобладания.

Результаты решения СЛАУ с использованием трех различных методов для задачи с течением в прямолинейной круглой трубе приведены в табл. 1. Относительная точность во всех решателях задавалась равной 0,1; максимальное число внутренних итераций ограничивалось 1 000.

Нужно заметить, что сравнительно низкая относительная точность обеспечивает достаточно высокий темп сходимости решателя уравнения Навье—Стокса, вызывающего линейный решатель [25]. Повышение относительной точности линейного решателя, значительно удорожающего решение СЛАУ, как показала практика, не приводит к существенному изменению (около 10%) числа итераций решателя уравнения Навье—Стокса.

Для метода сопряженных градиентов использовался предобуславливатель Эйзенштата [26].

Решение матрицы СЛАУ многосеточным методом осуществляется с помощью V-цикла как наименее требовательного с вычислительной точки зрения. В параллельном режиме предобуславливатель работает автономно на каждом процессе для локальной матрицы, и после каждой итерации решателя происходят межпроцессорные обмены.

Результаты расчетов, приведенные в табл. 1, показывают, что по времени решения отдельную группу составляют матрицы скоростей и турбулентности, для которых порядок числа обусловленности равен единице. В эту группу не входит матрица давления, имеющая число обусловленности на несколько порядков выше и требующая значительно больших ресурсов для решения.

Из приведенной таблицы следует, что для матриц с небольшим числом обусловленности целесообразнее использовать классические итерационные методы, такие как метод Зейделя и метод сопряженных градиентов. Хотя эти методы имеют более низкую скорость сходимости по сравнению

Таблица 1

Среднее время решения различных матриц за один шаг

Матрица	Число обусловленности матрицы	Время решения СЛАУ		
		Метод Зейделя	Метод сопряженных градиентов	Алгебраический многосеточный метод
Скорость по X	$6,1 \cdot 10^1$	0,001	0,008	0,044
Скорость по Y	$6,1 \cdot 10^1$	0,001	0,008	0,044
Скорость по Z	$6,1 \cdot 10^1$	0,001	0,008	0,044
Давление	$4,8 \cdot 10^7$	11,791	0,562	0,201
Турбулентный параметр K	$5,4 \cdot 10^1$	0,001	0,005	0,045
Турбулентный параметр W	$2,2 \cdot 10^1$	0,001	0,005	0,057

² Общие предварительные исследования данной реализации приведены в [22].

с многосеточным методом, что следует из числа необходимых итераций для решения матриц СЛАУ, они не требуют дополнительных этапов построения последовательности вложенных матриц.

Ситуация кардинально меняется для матриц с большим числом обусловленности, таких как матрица давления. Классические итерационные методы в этом случае требуют большого числа итераций для достижения даже небольшой точности [21]. Использование же многосеточного метода, несмотря на затраты дополнительного времени для построения последовательности вложенных матриц, здесь оправдано за счет более высокой скорости сходимости. Результаты также свидетельствуют о том, что при решении матрицы давления многосеточный метод дает существенный, более, чем в 2 раза, выигрыш по сравнению с наиболее используемым на практике методом сопряженных градиентов.

Чтобы детально проанализировать применимость многосеточного метода для решения матриц давления, получаемых в результате дискретизации основной системы уравнений методом SIMPLE, использовались три теста.

Тест 1. Рассматривалось стационарное изотермическое турбулентное течение вязкой несжимаемой жидкости в прямолинейной гладкой трубе с круглым сечением, соответствующее числу Рейнольдса 10 000. Для анализа применения многосеточного метода на последовательности сгущающихся сеток были выбраны три сеточные модели трубы: с 40 000, 800 000 и 1 500 000 ячеек (рис. 7). Численный расчет проводился на установление до получения стационарного режима течения с использованием SST-модели турбулентности [27] с автоматическим определением ширины пограничного слоя.

Для каждой из представленных моделей задача решалась с использованием двух методов решения СЛАУ: предобусловленного метода сопряженных градиентов (PCG) и алгебраического многосеточного метода с агрегативным способом огрубления (AMG). Применялось граничное условие *статическое давление*. Для такой постановки число обусловленности матрицы давления, как отмечалось выше, имеет значение 10^7 . Задача решалась в параллельном режиме, и для всех вариантов количество процессоров равнялось 12.

Из результатов для матриц давления, представленных в табл. 2, видно, что время решения задачи при использовании метода сопряженных градиентов для самой грубой сеточной модели в 3 раза больше, чем для многосеточного метода. Время решения для самой подробной сеточной модели с использованием метода сопряженных градиентов более чем в 40 раз превышает время решения алгебраическим многосеточным методом.

Отметим, что эти цифры не являются характерными для данного класса задач, а зависят от многих факторов, таких как структура течения, сложность сеточной модели, накладываемые граничные условия и т. д. Опыт практического применения многосеточного метода для решения задач вычислительной гидродинамики показывает, что среднее ускорение счета по сравнению с методом сопряженных градиентов достигает 4–6 раз.

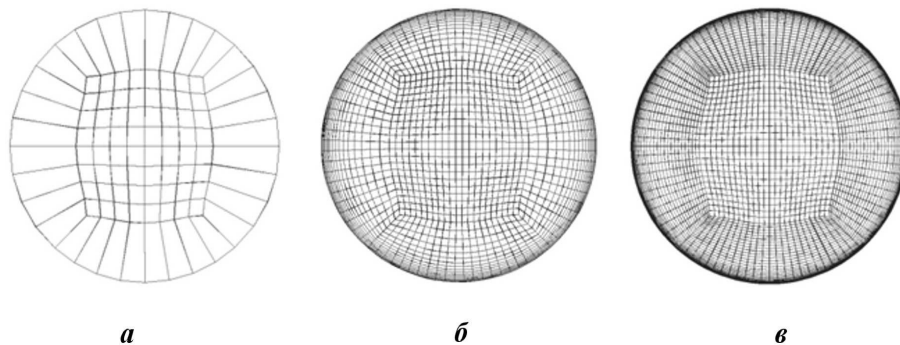


Рис. 7. Варианты сеточных моделей: *a* — 40 000 ячеек; *b* — 800 000 ячеек; *v* — 1 500 000 ячеек

Таблица 2

Тест 1. Полное время решения задачи и число итераций

Решатель СЛАУ	Модель 1 (40 000 ячеек)		Модель 2 (800 000 ячеек)		Модель 3 (1 500 000 ячеек)	
	Число итераций	Время решения, с	Число итераций	Время решения, с	Число итераций	Время решения, с
AMG	461	20,66	590	565	1 168	3 828
PCG	489	62,57	549	16 113	1 168	158 742

Тест 2. Чтобы показать преимущества использования в многосеточном методе глобального уровня, были взяты три различные постановки задачи с течением в "круглой" трубе, отличающиеся друг от друга по количеству используемых MPI-процессов: 6, 12 и 24. Сеточная модель решаемой задачи представлена на рис. 7, б. Задача решалась с помощью решателя AMG в двух вариантах: без использования глобального уровня (AMG1) и с его использованием (AMG2).

Время решения задачи и число итераций до сходимости решения представлены в табл. 3. Видно, что при использовании решателя с глобальным уровнем время решения на 24 процессах уменьшается более чем в 2 раза, хотя на 6 процессах только в 1,2 раза.

Тест 3. Наиболее ярко различие между применением классических и многосеточных методов решения СЛАУ проявляется при расчете задач, содержащих десятки миллионов счетных ячеек, где время счета отдельной задачи может различаться на несколько, а то и несколько десятков часов. В качестве примера приведем решение задачи с течением в канале за обратным уступом (рис. 8, см. также цветную вкладку). Полное описание, постановка и методы решения задачи даны в [24]. В качестве модели турбулентности выбрана SST-модель с автоматическим определением ширины пограничного слоя. Подчеркнем, что цель данного теста — не продемонстрировать точность полученных результатов по сравнению с экспериментом, а показать основные преимущества алгебраического многосеточного метода по сравнению с итерационными методами. Поэтому, чтобы сделать задачу показательной с точки зрения решателей СЛАУ, были построены две сеточные модели: на 14 млн и 30 млн счетных ячеек.

Чтобы наиболее полно и точно оценить результаты теста и сделать соответствующие выводы, для каждой из перечисленных моделей было выбрано 9 тестовых случаев. Каждая модель решалась с использованием разного количества процессов (в зависимости от числа счетных ячеек) и

Таблица 3

Тест 2. Полное время решения задачи и число итераций

Решатель СЛАУ	6 процессов		12 процессов		24 процесса	
	Число итераций	Время решения, с	Число итераций	Время решения, с	Число итераций	Время решения, с
AMG1	1 425	647	1 377	365	1 397	238
AMG2	1 469	552	1 402	233	1 442	104

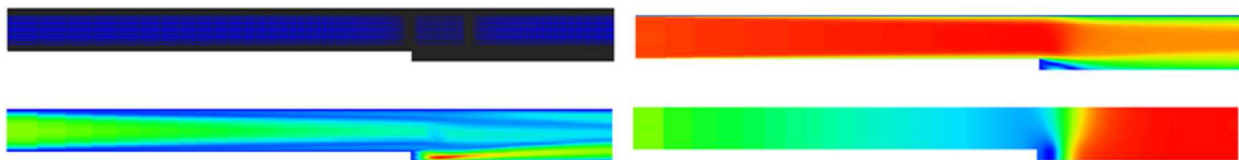


Рис. 8. Тест 3. Сеточная топология. Распределение скорости, турбулентной вязкости и давления

разных решателей СЛАУ: многосеточного метода с глобальным уровнем и без него, а также метода сопряженных градиентов. Все полученные результаты сведены в табл. 4, 5 и позволяют сделать однозначный вывод в пользу использования того или иного решателя. Также в таблицах представлены результаты по количеству внешних итераций до сходимости метода SIMPLE, так как это количество ожидаемо разное и находится в пределах ± 1000 итераций, что также вызывает дополнительные сложности при вычислении временных характеристик решателей СЛАУ.

Что касается эффективности распараллеливания, то, по оценкам авторов, при использовании многосеточных технологий наиболее эффективно на одном вычислительном ядре вести расчет 30–70 тыс. ячеек. Данный диапазон чисел является совокупным показателем и определяется как самой вычислительной системой (используемые процессоры, архитектура, топология коммуникационной среды и т. п.), так и методами распараллеливания, применяемыми в программном комплексе ЛОГОС.

Из представленных в табл. 4, 5 результатов расчетов следует, что использование глобального уровня в многосеточном методе позволяет значительно сократить время решения задачи. Кроме того, многосеточный метод с глобальным уровнем по времени решения является более предпочтительным, чем метод сопряженных градиентов. Тем не менее следует отметить, что метод сопряженных градиентов лучше масштабируется на большом числе процессов. Это связано с тем, что решение на глобальном уровне является этапом, выполняющимся в последовательном режиме, и, следовательно, ухудшает масштабируемость. В перспективе в качестве решения данной проблемы планируется реализовать сбор каскадных уровней для постепенного уменьшения числа процессов, участвующих в счете, сохраняя при этом преимущества глобального уровня.

Таблица 4

Тест 3. Результаты решения задачи на сетке с 14 млн ячеек

Количество процессов	Многосеточный метод с глобальным уровнем		Многосеточный метод без глобального уровня		Метод сопряженных градиентов	
	Время решения, с	Число итераций SIMPLE	Время решения, с	Число итераций SIMPLE	Время решения, с	Число итераций SIMPLE
100	15 927	6 642	223 623	5 909	67 519	5 738
250	7 888	6 237	98 444	5 962	31 177	5 661
500	5 695	6 569	62 181	5 141	19 006	5 918
1 000	5 377	5 853	63 850	6 935	9 814	5 737

Таблица 5

Тест 3. Результаты решения задачи на сетке с 30 млн ячеек

Количество процессов	Многосеточный метод с глобальным уровнем		Многосеточный метод без глобального уровня		Метод сопряженных градиентов	
	Время решения, с	Число итераций SIMPLE	Время решения, с	Число итераций SIMPLE	Время решения, с	Число итераций SIMPLE
100	42 701	10 295	176 973	10 179	177 169	10 414
250	17 458	9 886	126 147	10 240	83 461	10 719
500	11 199	9 882	72 190	10 211	54 799	9 998
1 000	12 352	10 368	75 130	10 355	23 787	10 009
2 000	15 848	10 001	95 201	9 611	19 858	10 172

Выводы

Рассмотрено применение алгебраического многосеточного метода с агрегативным способом огрубления для численного решения уравнений гидродинамики вязкой несжимаемой жидкости на произвольных неструктурированных сетках. Анализ времени счета для типичных задач динамики вязкой несжимаемой жидкости показал, что ускорение счета по сравнению с методом сопряженных градиентов при использовании нескольких сотен процессоров и сетки в несколько десятков миллионов ячеек составляет не менее 4 раз.

Работа выполнена при финансовой поддержке Министерства образования и науки РФ (государственный контракт № 14.514.12.0002) и Российского фонда фундаментальных исследований в рамках научного проекта № 13-07-12079.

Список литературы

1. *Saad Y.* Iterative Methods for Sparse Linear Systems. Philadelphia: Society for Industrial and Applied Mathematics, 2003.
2. *Brandt A.* Guide to multigrid development // Lecture Notes in Mathematics. 1982. Vol. 960. P. 220—312.
3. *Федоренко Р. П.* Релаксационный метод решения разностных эллиптических уравнений // Журнал вычисл. мат. и мат. физ. 1961. Т. 1, № 5. С. 922—927.
4. *Бахвалов Н. С.* О сходимости одного релаксационного метода для эллиптического оператора с естественными ограничениями // Там же. 1966. Т. 6. С. 101—135.
5. *Петров И. Б., Лобанов А. И.* Лекции по вычислительной математике: Учебное пособие. М.: Интернет-университет информационных технологий, БИНОМ, 2006.
6. *Stuben K., Trottenberg U.* Multigrid methods: fundamental algorithms, model problem analysis and applications // Multigrid Methods, Lecture Notes in Mathematics 960 / Ed. by W. Hackbusch and U. Trottenberg. Berlin: Springer, 1982.
7. *Vanek P., Mandel J., Brezina M.* Algebraic multigrid based on smoothed aggregation for second and fourth order problems // Computing. 1996. Vol. 56, No 2. P. 179—196.
8. *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.
9. *Ferziger J. H., Peric M.* Computational Methods for Fluid Dynamics. Berlin, New York, Barcelona, Hong Kong, London, Milan, Paris, Tokyo: Springer, 2002.
10. *Darwish M. S., Saad T., Hamdan Z.* A high scalability parallel algebraic multigrid solver // Proc. of the European Conference on Computational Fluid Dynamics (ECCOMAS CFD 2006). Netherlands, 5—8 September 2006.
11. *Emans M.* Benchmarking aggregation amg for linear systems in CFD simulations of compressible internal flows // Electronic Transactions on Numerical Analysis. 2010. Vol. 37. P. 351—366.
12. *Van Emden H., Meier-Yang U.* BoomerAMG: a parallel algebraic multigrid solver and preconditioner // Appl. Num. Math. 2001. Vol. 41, No 1. P. 155—177.
13. *Stuben K.* A review of algebraic multigrid // J. Comp. and Appl. Math. 2001. Vol. 128, No 1—2. P. 281—309.
14. *Погосян М. А., Савельевских Е. П., Стрелец Д. Ю., Корнев А. В.* Отечественные суперкомпьютерные технологии в авиационной промышленности // Наука и технологии в промышленности. 2012. № 2. С. 26—35.
15. *Козелков А. С., Дерюгин Ю. Н., Зеленский Д. К. и др.* Многофункциональный пакет программ ЛОГОС для расчета задач гидродинамики и тепломассопереноса на многопроцессорных ЭВМ: базовые технологии и алгоритмы // Тр. XII Межд. семинара "Супервычисления

- и математическое моделирование". Саров, 11—15 октября 2010 г. Саров: РФЯЦ-ВНИИЭФ, 2011. С. 215—230.
16. Флетчер К. Вычислительные методы в динамике жидкостей (в двух томах). М: Мир, 1991.
 17. Быстров Ю. А., Исаев С. А., Кудрявцев Н. А., Леонтьев А. И. Численное моделирование вихревой интенсификации теплообмена в пакетах труб. С.-Пб.: Судостроение, 2005.
 18. Шведкий В. С., Ярошенко Ю. Г. Механика жидкости и газа. М.: Академкнига, 2003.
 19. Jasak H. Error Analysis and Estimation for the Finite Volume Method With Applications to Fluid Flows. Thesis submitted for the degree of doctor. Department of Mechanical Engineering, Imperial College of Science, 1996.
 20. Бахвалов Н. С., Жидков Н. П., Кобельков Г. М. Численные методы. М.: Лаборатория базовых знаний, 2002.
 21. Уоткинс Д. С. Основы матричных вычислений: Пер. с англ. М.: БИНОМ, 2009.
 22. Дерюгин Ю. Н., Голубев А. А., Зеленский Д. К. и др. Пакет программ ЛОГОС. Разработка и реализация алгебраического многосеточного метода // Тез. докл. XIV Межд. конф. "Супервычисления и математическое моделирование". Саров: РФЯЦ-ВНИИЭФ, 2012. С. 58
 23. Weiss J. M., Maruszewski J. P., Smith W. A. Implicit solution of preconditioned Navier-Stokes equations using algebraic multigrid // AIAA J. 1997. Vol. 37, No. 1. P. 29—36.
 24. Vogel J. C., Eaton J. K. Combined heat transfer and fluid dynamic measurements downstream of a backward-facing step // J. of Heat Transfer. 1985. Vol. 107. С. 922—929.
 25. Carol S. Woodward. SUNDIALS: Suite of Nonlinear and Differential/Algebraic Equation Solvers. UCRL-PRES-213978.
 26. Meurant G. Computer solution of large linear systems. Amsterdam, Lausanne, New York, Oxford, Shannon, Singapore, Tokyo: Elsevier, 1999.
 27. Menter F. R. Two-equation Eddy-Viscosity turbulence models for engineering applications // AIAA J. 1994. Vol. 32. P. 1598—1605.

Статья поступила в редакцию 02.04.13.
