

ОРДЕНА ЛЕНИНА ИНСТИТУТ ПРИКЛАДНОЙ
МАТЕМАТИКИ
им.М.В.Келдыша
РОССИЙСКОЙ АКАДЕМИИ НАУК

А.Е. Луцкий, А.С. Черногузов

ЧИСЛЕННОЕ ИССЛЕДОВАНИЕ ВЯЗКИХ ТЕЧЕНИЙ
НА МНОГОПРОЦЕССОРНЫХ
ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

Москва, 2005

УДК.519.6

А.Е. Луцкий, А.С. Черногузов

Email: lutsky@kiam.ru, alexch@kiam.ru

ЧИСЛЕННОЕ ИССЛЕДОВАНИЕ ВЯЗКИХ ТЕЧЕНИЙ НА МНОГОПРОЦЕССОРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

Ключевые слова: разностная схема, нестационарные уравнения Навье-Стокса, криволинейная сетка, явная схема, схема Годунова-Колгана, турбулентная вязкость, параллельный алгоритм.

(Работа выполнена при поддержке Программы Фундаментальных Исследований Президиума РАН N17 “Параллельные вычисления и многопроцессорные вычислительные системы“)

АННОТАЦИЯ

Представлен параллельный алгоритм моделирования задач обтекания тел сложной формы. Используются многоблочные сетки. Решение полной системы уравнений Навье-Стокса получается методом установления по времени. Аппроксимация членов без вязкости осуществляется с помощью разностной схемы Годунова-Колгана. Члены с вязкостью аппроксимируются центральными разностями. Представлены результаты исследования эффективности параллельного алгоритма на примере расчетов сверхзвукового обтекания профиля и обратного уступа.

A.E. Lutsky, A.S. Chernoguzov

THE NUMERICAL ANALYSIS OF VISCOUS FLOWS ON THE MULTIPROCESSOR COMPUTERS

ABSTRACT

The parallel algorithm for numerical simulation of viscous flows over complex shaped bodies is presented. Block structured grids are used. The time-stepping procedure is used to find steady-state solution of the full Navier-Stokes equations. The Godunov-Kolgan scheme for the inviscid terms are used, the central differences for approximation of viscous terms are used. The results of efficiency of parallel algorithm research are shown on the supersonic flow over the wing section and on the supersonic flow over the backward-facing step.

Оглавление

Введение	3
§1 Постановка задачи, математическая модель	6
§2 Численная реализация	8
§3 Последовательный и параллельный алгоритмы	10
§3.1 Балансировка загрузки	10
§3.2 Параллельный алгоритм	12
§4 Результаты расчетов и эффективность параллельной реали- зации	14
§4.1 Задача о профиле	15
§4.2 Задача об обратном уступе	16
Литература	18

Введение

Широкий круг проблем, стоящих перед современной наукой и техникой, связан с решением уравнений газовой динамики. Перед исследователями ставятся задачи, для выполнения которых требуется усложнения математической модели. Актуальны задачи проектирования и конструирования летательных аппаратов и элементов их конструкций, оптимизации их форм с целью улучшения аэродинамических характеристик, а также предотвращения негативных нагрузок и уменьшения шума. При расчете обтекания летательных аппаратов и элементов их конструкций возникают особенности, связанные с вязкостью среды, приводящей в ряде случаев к существенной перестройке решения (см., например, [1]). Эффекты вязкости играют особенно большую роль вблизи поверхности тела (пограничный слой) и на подветренной стороне тела, где может образоваться отрывное течение. Для изучения этих проблем необходимо решение полных уравнений Навье-Стокса для сжимаемого теплопроводного газа с учетом дополнительных факторов в зависимости от задачи. Единственным эффективным способом решения таких сложных систем нелинейных уравнений является использо-

вание численных методов на многопроцессорных вычислительных машинах.

Моделирование газодинамических течений на основе численного расчета уравнений Навье-Стокса достаточно дорогостоящая в вычислительном плане процедура. Это связано с большим числом узлов разностных сеток, применяемых в реальных расчетах. В процессе моделирования сверхзвуковых течений возникает необходимость использовать подробную сетку в окрестности течения с большими градиентами газодинамических величин (например, ударные волны). Помимо этого необходимо увеличивать плотность узлов в области пограничного слоя [6].

Таким образом, одной из основных трудностей по применению численных методов для решения задач газовой динамики является большой объем вычислений. С течением времени возможности вычислительной техники увеличиваются, что позволяет повышать точность получаемых результатов и степень детализации картины физических явлений. Одновременно с этим появляется возможность изменить постановку задачи и усовершенствовать применяемый для ее решения математический аппарат. Рост возможностей электронной техники позволяет повышать сложность математических моделей, а усовершенствованные модели требуют для своей реализации технику все большей производительности.

Одним из наиболее распространенных направлений развития вычислительной техники на сегодняшний день представляют многопроцессорные вычислительные системы. Основным параметром их классификации является наличие общей (SMP) или распределенной памяти (MPP). Кроме того, в последнее время во всем мире происходит бурное внедрение вычислительных кластеров, которые являются более дешевым вариантом MPP. Все большую популярность приобретают идеи комбинирования различных архитектур в одной системе и построения неоднородных систем.

Массивно-параллельные системы (MPP) состоят из однородных вычислительных узлов, включающих один или несколько центральных процессоров, локальную память (прямой доступ к памяти других узлов невозможен), коммуникационный процессор или сетевой адаптер, жесткие диски и/или другие устройства ввода/вывода. Узлы связаны через некоторую коммуникационную среду, общее число процессоров в реальных системах достигает нескольких тысяч (ASCI Red, Blue Mountain). Программирование выполняется в рамках модели передачи сообщений (MPI).

Симметричные мультипроцессорные системы (SMP) состоят из нескольких однородных процессоров и массива общей памяти (обычно из нескольких независимых блоков). Все процессоры имеют доступ к любой точке памяти с одинаковой скоростью. Процессоры подключены к памяти либо

с помощью общей шины (базовые 2-4 процессорные SMP-сервера), либо с помощью crossbar-коммутатора. Аппаратно поддерживается когерентность кэшей. Наличие общей памяти сильно упрощает взаимодействие процессоров между собой, однако накладывает сильные ограничения на их число - не более 32 в реальных системах. Для построения масштабируемых систем на базе SMP используются кластерные или NUMA-архитектуры. Программирование для SMP систем выполняется в рамках модели общей памяти. (POSIX threads, OpenMP) [7].

Системы с распределенной памятью, по-видимому, всегда будут лидировать по показателю пиковой производительности, поскольку любые новые однопроцессорные (или многопроцессорные на основе общей памяти) системы могут быть легко объединены сетью и использованы в качестве многопроцессорных вычислительных систем с распределенной памятью. Кроме того, массивно параллельные системы и вычислительные кластеры почти всегда выигрывают у систем с общей памятью по показателю соотношения цены и производительности.

Использование многопроцессорных вычислительных систем обеспечивает высокое быстродействие, необходимое для эффективного решения сложных задач. Однако адаптация последовательных алгоритмов к параллельным системам вообще и к системам с распределенной памятью в особенности является достаточно сложной задачей [8]. Основными на сегодняшний день методами построения параллельных программ и алгоритмов являются алгоритмический параллелизм и геометрический параллелизм.

В качестве следующего препятствия на пути эффективного использования параллельной техники справедливо называют проблему обеспечения равномерной загрузки процессоров в ходе решения задачи. Задача балансировки загрузки является одной из основных задач, решаемых при проведении параллельных вычислений. Необходимо таким образом распределить вычислительную и иную нагрузку между процессорами системы, чтобы по возможности исключить простои отдельных процессоров и наиболее эффективно использовать доступные ресурсы. В основу предложенного параллельного алгоритма для MPP систем положен метод геометрического параллелизма, который предполагает деление общей расчетной области на множество подобластей (блоков), относимых к различным процессорам системы путем соответствующего распределения ячеек сетки. При этом, в случае использования систем с распределенной памятью, необходимо с одной стороны как можно более равномерно распределить между процессорами суммарный объем вычислений, а с другой стороны минимизировать межпроцессорные обмены данными, чтобы сократить общее время работы каждого из процессоров. Как правило, критериями разбиения являются

примерно равное число узлов по подобластям. В данной работе показано, что также необходимо учитывать картину течения в подобластях, потому что в узлах с большими градиентами газодинамических величин расчет конвективных потоков занимает больше времени, чем в узлах с относительно равномерным потоком.

При разработке параллельных алгоритмов и программ моделирования задач аэродинамики большое значение имеет снижение затрат времени на вспомогательные операции, не имеющие отношения непосредственно к процессу расчета. Это загрузка данных, запись промежуточных и окончательных результатов расчетов (контрольных точек). Время выполнения этих действий должно быть минимизировано, и не должно сильно зависеть от числа используемых процессоров.

В данной работе, с учетом приведенных выше требований к параллельным алгоритмам, на основе модели геометрического параллелизма разработан параллельный алгоритм моделирования газодинамических течений, основанный на математической модели двумерных нестационарных полных уравнений Навье-Стокса для вязкого теплопроводного газа для многопроцессорных систем с распределенной памятью. Изучается влияние вязкости на примере 2D сверхзвукового обтекания профиля крыла. Аппроксимация членов без вязкости осуществляется с помощью разностной схемы Годунова-Колгана [2] с кусочно-линейным представлением газодинамических функций, члены с вязкостью аппроксимируются центральными разностями.

§1 Постановка задачи, математическая модель

Алгоритм для расчета обтекания тел сложной формы основан на математической модели нестационарных полных уравнений Навье-Стокса для вязкого теплопроводного газа. Стационарное решение (если таковое существует) получается методом установления по времени. В параметры задачи входят число Маха M_∞ , число Рейнольдса Re . Поле начальных газодинамических величин часто задается как невозмущенный набегающий поток.

Безразмерные величины вводятся следующим образом:

$$\rho = \rho_\infty \bar{\rho}, \quad p = p_\infty \bar{p}, \quad u = U_\infty \bar{u}, \quad U_\infty = \sqrt{\frac{p_\infty}{\rho_\infty}}, \quad t = \frac{L}{U_\infty} \bar{t}$$

$$T = T_\infty \bar{T}, \quad T_\infty = \frac{p_\infty}{\rho_\infty R}$$

Далее будем считать, что все используемые ниже газодинамические величины являются безразмерными. Нестационарные безразмерные 2D урав-

нения Навье-Стокса записываются в криволинейных координатах в виде [4, стр.132-133]:

$$\vec{Q}_t + \vec{f}_\xi + \vec{g}_\eta = 0 \quad f = f^i + f^v, \quad g = g^i + g^v$$

$$Q = \frac{1}{J} \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ e \end{pmatrix} \quad f^i = \frac{1}{J} \begin{pmatrix} \rho U^\xi \\ \rho u U^\xi + p \xi_x \\ \rho v U^\xi + p \xi_y \\ (e + p) U^\xi \end{pmatrix} \quad g^i = \frac{1}{J} \begin{pmatrix} \rho U^\eta \\ \rho u U^\eta + p \eta_x \\ \rho v U^\eta + p \eta_y \\ (e + p) U^\eta \end{pmatrix}$$

$$f^v = \frac{M_\infty \sqrt{\gamma}}{J \cdot Re} \begin{pmatrix} 0 \\ \xi_x \tau_{xx} + \xi_y \tau_{xy} \\ \xi_x \tau_{xy} + \xi_y \tau_{yy} \\ (u \tau_{xx} + v \tau_{xy}) \xi_x + (u \tau_{xy} + v \tau_{yy}) \xi_y - q_\xi \end{pmatrix}$$

$$g^v = \frac{M_\infty \sqrt{\gamma}}{J \cdot Re} \begin{pmatrix} 0 \\ \eta_x \tau_{xx} + \eta_y \tau_{xy} \\ \eta_x \tau_{xy} + \eta_y \tau_{yy} \\ (u \tau_{xx} + v \tau_{xy}) \eta_x + (u \tau_{xy} + v \tau_{yy}) \eta_y - q_\eta \end{pmatrix}$$

Поле скоростей представлено декартовыми составляющими u, v . С индексом i представлены потоки с невязкими членами, с индексом v — потоки с вязкими членами. Здесь введены обозначения:

$$U^\xi = u \xi_x + v \xi_y \quad U^\eta = u \eta_x + v \eta_y$$

Производные ξ_x, η_x и ξ_y, η_y - метрические коэффициенты. Они получаются следующим образом:

$$\xi_x = J \cdot y_\eta, \quad \xi_y = -J \cdot x_\eta, \quad \eta_x = -J \cdot y_\xi, \quad \eta_y = J \cdot x_\xi, \quad J = \frac{1}{y_\eta x_\xi - y_\xi x_\eta},$$

где $J = \partial(\xi, \eta) / \partial(x, y)$ - якобиан преобразования.

Используется модель совершенного газа $p = \varepsilon \rho(\gamma - 1) = \rho R T$. Для воздуха: $\gamma = 1.4$, газовая постоянная $R = 287.0$, число Прандтля $Pr = 0.72$, и турбулентное число Прандтля $Pr_t = 0.9$. Полная энергия, отнесенная к единице объема

$$e = \rho \varepsilon + \rho \frac{u^2 + v^2}{2} = \frac{p}{\gamma - 1} + \rho \frac{u^2 + v^2}{2}$$

Компоненты тензора вязких напряжений задаются соотношениями

$$\tau_{xx} = \frac{2}{3} \mu (2u_x - v_y) = \frac{2}{3} \mu [2(\xi_x u_\xi + \eta_x u_\eta) - (\xi_y v_\xi + \eta_y v_\eta)]$$

$$\tau_{yy} = \frac{2}{3} \mu (2v_y - u_x) = \frac{2}{3} \mu [2(\xi_y v_\xi + \eta_y v_\eta) - (\xi_x u_\xi + \eta_x u_\eta)]$$

$$\tau_{xy} = \tau_{yx} = \mu (u_y + v_x) = \mu [(\xi_y u_\xi + \eta_y u_\eta) + (\xi_x v_\xi + \eta_x v_\eta)]$$

Тепловые потоки задаются соотношениями

$$q_\xi = -\frac{\gamma}{(\gamma - 1)} \left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t} \right) (\xi_x T_x + \xi_y T_y)$$

$$q_\eta = -\frac{\gamma}{(\gamma - 1)} \left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t} \right) (\eta_x T_x + \eta_y T_y)$$

Динамический коэффициент вязкости μ выражается через абсолютную температуру по формуле Саттерленда

$$\frac{\mu}{\mu_0} = \left(\frac{T}{T_0} \right)^{\frac{3}{2}} \frac{T_0 + T_S}{T + T_S},$$

где T_S — постоянная Саттерленда, имеющая для воздуха значение, близкое к $122K$, а T_0 и μ_0 — абсолютная температура и коэффициент вязкости соответствующие начальному состоянию газа [6, стр.741].

В настоящей работе для расчета турбулентных течений используются осредненные по Рейнольдсу уравнения, замыкаемые алгебраической моделью Болдуина-Ломакса для расчета турбулентной вязкости [3, стр.474], [9].

§2 Численная реализация

В плоскости (ξ, η) в каждом блоке записывается явная разностная схема:

$$\frac{Q_{kl}^{n+1} - Q_{kl}^n}{\Delta t} + \frac{\vec{f}_{k+\frac{1}{2}} - \vec{f}_{k-\frac{1}{2}}}{\Delta \xi} + \frac{\vec{g}_{l+\frac{1}{2}} - \vec{g}_{l-\frac{1}{2}}}{\Delta \eta} = 0$$

$$k = 1, \dots, (kt - 1), \quad l = 1, \dots, (lt - 1)$$

Для вычисления значений газодинамических величин Q_{kl}^{n+1} в ячейке на новом слое по времени необходимо выполнить последовательную обработку всех граней этой ячейки, а именно вычислить конвективные и диссипативные потоки через эти грани: $\vec{f}_{k\pm\frac{1}{2}}$, $\vec{g}_{l\pm\frac{1}{2}}$.

На твердой поверхности ставятся граничные условия прилипания $u = v = 0$ и условия изотермической $T = T_w$ или адиабатической $\frac{\partial T}{\partial n} = 0$ стенки. На входных границах задается набегающий поток, на выходных границах стоят „мягкие“ условия, т.е. значения газодинамических величин экстраполируются с нулевым порядком.

Геометрия расчетной области задается координатами узлов расчетной сетки (вершины ячеек), которая является регулярной. Расчетная область состоит из ячеек, ячейки объединены в блоки, которые потом распределяются по процессорам. Блоки связаны друг с другом так, что вершины блоков соприкасаются, сторона одного блока соприкасается со стороной другого блока целиком. Для блоков задается ориентация, определяющая

порядок нумерации ячеек: горизонтальная координата ξ идет от West к East, вертикальная координата η идет от South к North. Используя ориентацию блоков, можно установить связь между ними: для каждой из 4-х сторон блока заданы номер соседнего блока и сторона, которой соседний блок обращен к рассматриваемой стороне. Этих данных достаточно, чтобы определить связи блоков, ориентацию сторон у соприкасающихся блоков и номера процессоров соседних блоков. Согласно используемому шаблону разностной схемы в каждом блоке используется по два слоя соседних ячеек для каждой границы и четыре ячейки по диагоналям. В случае, когда блок граничит с другим блоком, в эти ячейки передаются значения газодинамических величин от соседнего блока, в противном случае значения определяются граничным условием, заданным на этой стороне. На *рис.1* показана схема взаимодействия блоков с помощью граничных ячеек.

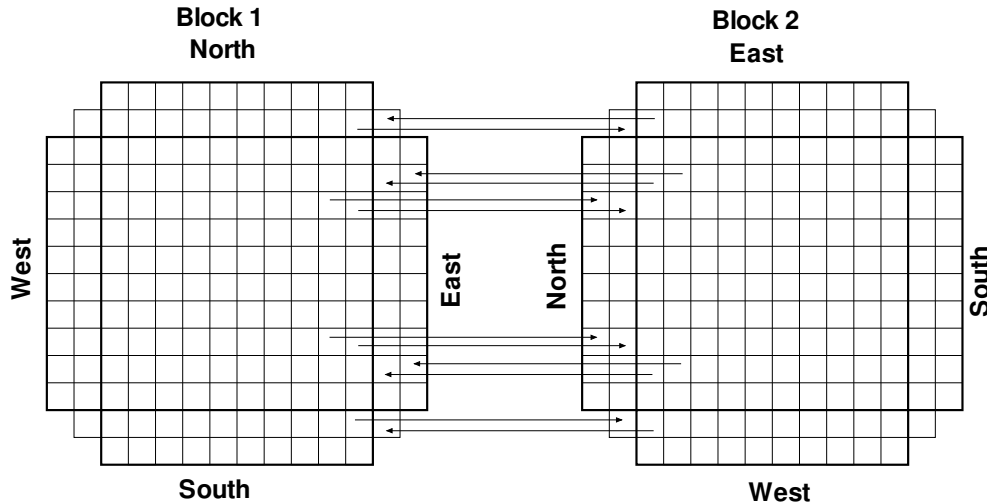


Рис. 1: Схема взаимодействия блоков.

Использование явной схемы упрощает распараллеливание алгоритма и обработку границ между блоками.

Процедура подготовки расчета (параллельного выполнения вычислительных итераций) делится на несколько этапов. На этапе балансировки загрузки выполняется распределение блоков, составляющих расчетную область, по процессорам, в результате чего каждый процессор получает список обрабатываемых блоков. Следующий этап заключается в подготовке процессором данных, необходимых для расчета, в том числе подготовка полученной расчетной подобласти с учетом передаваемых и принимаемых данных и одновременной инициализация массивов приема и передачи данных. В результате такой локализации геометрической модели расчетной подобласти процессора на основе предлагаемого алгоритма её дальнейшая

обработка с алгоритмической точки зрения практически не отличается от обработки сетки целиком, т.е. исключаются неоднородности вычислительного алгоритма, связанные с обработкой данных, принадлежащих другим процессорам.

Шаг по времени вычислялся следующим образом: сначала получаем Δt_x и Δt_y из условия устойчивости схемы для течений с $\mu = 0$, находим минимальные Δt_x и Δt_y , затем

$$\Delta t_{min} = CFL \frac{1}{\frac{1}{\Delta t_x} + \frac{1}{\Delta t_y} + \frac{1}{\Delta t_{viscous}}} = K \cdot \frac{1}{\frac{1}{\Delta t_x} + \frac{1}{\Delta t_y}},$$

$$\text{где } \frac{1}{\Delta t_{viscous}} = \max \left[\frac{4\mu\gamma}{Pr \rho} \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) \right].$$

В большинстве случаев $CFL \approx 0.5$, $K \approx 0.45$.

§3 Последовательный и параллельный алгоритмы

Разработка эффективного последовательного алгоритма имеет большое практическое значение. В процессе его разработки и дальнейшей программной реализации выполняется тестирование эффективности и корректность основных вычислительных модулей. Помимо этого последовательная программа в большинстве случаев является базовой точкой для оценки эффективности разрабатываемого программного обеспечения и правильности работы параллельных программ.

При распараллеливании на основе метода геометрического параллелизма какого-либо последовательного алгоритма расчетная область делится на непересекающиеся подобласти, относимые к обработке на различных процессорах. При этом в случае реализации однородных вычислительных алгоритмов на каждом процессоре решается отдельная подзадача, которая, по сути, мало чем отличается от обработки всей сетки в последовательном режиме. Единственное отличие состоит в необходимости обменов данными и, следовательно, синхронизации работы вычислительных узлов.

Используемые в последовательной программе основные вычислительные модули (определение конвективных и диссипативных потоков через грань контрольного объема) без изменений используются в параллельной реализации алгоритма. Различие заключается в дополнительном обмене данными и способом обмена граничными ячейками между ними.

§3.1 Балансировка загрузки

Одной из основных и наиболее трудоемких задач, решаемых при разработке параллельных алгоритмов, является проблема балансировки загрузки. Справедливо утверждение, что увеличение числа процессоров, между которыми распределена работа, уменьшает время решения многих задач. Однако, в случае использования систем с распределенной памятью, распределение работы между процессорами приводит к необходимости обмена данными между ними, что замедляет вычисления.

Оценка эффективности параллельных алгоритмов обычно выполняется с использованием понятий эффективности и ускорения параллельных алгоритмов. Ускорением параллельного алгоритма (S_P) называют отношение времени выполнения алгоритма на одном процессоре ко времени выполнения алгоритма на системе из P процессоров :

$$S_P = \frac{T_P}{T_1}$$

Эффективностью параллельного алгоритма (E_P) называют отношение его ускорения к числу процессоров, на котором это ускорение получено:

$$E_P = \frac{S_P}{P}$$

$S_P = P$ и $E_P = 1$ — это наилучшие значения для S_P и E_P , но на практике получают меньшие значения. В общем случае эффективность снижается за счет следующих факторов:

- существуют затраты времени на обмен данными между процессорами;
- параллельный алгоритм всегда содержит некоторое количество дополнительных действий, связанных с управлением параллельной программой и синхронизацией ее частей

Можно отметить, что если большая часть обрабатываемых данных помещается в кэш процессора, то бывает что, $S_P > P$, а $E_P > 1$.

Очевидно, что эффективность и ускорение параллельного алгоритма напрямую зависят от качества балансировки загрузки. Одновременно с этим объем передаваемых данных должен быть минимизирован. При распределении нагрузки процессоров также по возможности необходимо учитывать особенности решения в некоторых подобластях расчетной области и пытаться распределять ячейки с учетом этой информации, потому что от этих особенностей решения зависит время вычисления конвективных потоков на гранях ячейки.

Передачу данных целесообразно рассматривать как один из видов работы, суммируя общий объем выполняемой процессором работы с общим

объемом передаваемых/принимаемых им данных. При таком подходе задача сводится к равномерному распределению получившейся „совокупной“ работы.

Различают статическую и динамическую балансировки загрузки. Статическая балансировка выполняется перед началом вычислений. При этом делаются определенные предположения относительно времени выполнения частей программы и объема передаваемых между процессорами данных. Динамическая балансировка загрузки используется, когда время выполнения различных частей программы или эффективная производительность процессорных узлов изменяется в ходе решения задачи. В данной работе используется статическая балансировка загрузки процессоров, пользователь сам должен распределить блоки по процессорам перед началом расчета.

§3.2 Параллельный алгоритм

Традиционно разработка какого-либо параллельного алгоритма для систем с распределенной памятью считается сложнее чем для SMP систем. Это вызвано необходимостью локализации данных для обработки каждым процессором и инициализации массивов межпроцессорного обмена данными. Дополнительные накладные расходы возникают при копировании и считывании информации при работе с соответствующими буферами приема/передачи данных. Возникают задержки, связанные с синхронизацией работы процессоров. Эти задержки зависят от качества балансировки загрузки и других факторов.

Процедура подготовки к выполнению расчета состоит из нескольких этапов. На этапе инициализации определяется управляющий процессор (номер 0), задача которого в дальнейшем будет состоять в централизации работы: рассылает и принимает данные с процессоров, работает с файлами. Управляющий процессор загружает данные о параметрах задачи и рассылает их остальным процессорам. Это информация о свойствах блоков, из которых состоит расчетная область, таблица распределения блоков по процессорам и остальные данные, необходимые для выполнения расчета. В результате каждый процессор получает список обрабатываемых на нем блоков и таблицу, по которой блоки будут обмениваться друг с другом.

На втором этапе управляющий процессор загружает расчетную сетку и область с начальным распределением газодинамических величин, после чего рассылает остальным процессорам их блоки.

На следующем этапе происходит инициализация граничных ячеек блока. В зависимости от граничного условия данные для граничных ячеек могут быть получены из этого же блока, либо из блока, соседнего к рассмат-

риваемой границе. Если известно какими сторонами граничат эти блоки, то известен порядок нумерации ячеек на границах, он может быть сонаправлен или противоположно направлен. В случае обмена данными между процессорами передающий процессор предварительно буферизует данные, чтобы принимающий процессор принял их без буферизации.

При работе параллельной программы опираемся на следующие факты:

- в начале работы каждый процессор знает свой номер, нумерация процессоров идет по порядку начиная с нуля;
- если процессору передают информацию два других процессора, не имеет значения в каком порядке этот процессор будет принимать данные.

Схема межпроцессорного обмена (и одновременно синхронизация) граничными и диагональными ячейками устроена следующим образом: на каждом процессоре начинается обход граней всех блоков; если при обходе граней блока выяснилось, что процессор должен участвовать в обмене данными для текущей грани, то возможны два варианта:

1) если блок расположен на рассматриваемом процессоре, то процессор готовит данные со своей стороны и отправляет соседнему процессору, потом принимает данные от этого процессора;

2) если блок расположен на соседнем процессоре, то происходит прием данных от соседнего процессора, затем подготовка и отправка своих данных соседнему процессору.

Далее начинается расчет. Выполнение основных этапов на шаге происходит в следующей последовательности:

- обработка граничных ячеек и обмен граничными ячейками между блоками;
- расчет турбулентной вязкости во всей области при помощи алгебраической модели Болдуина-Ломакса;
- расчет турбулентной вязкости на границах блоков и обмен полученными значениями между блоками;
- вычисление конвективных потоков с использованием метода Годунова-Колгана;
- вычисление диссипативных потоков для вязких задач;
- вычисление шага по времени;

- обмен шагом по времени между блоками для определения минимального;
- определение газодинамических величин на новом слое по времени;
- расчет невязки и обмен невязкой между блоками для определения максимальной невязки;
- проверка условия выхода + выдачи в файлы.

Схема обмена шагом по времени и невязкой устроена следующим образом:

- каждый процессор определяет по своим блокам минимальный шаг по времени/максимальную невязку;
- управляющий процессор начинает обход процессоров, кроме себя;
- каждый процессор посылает управляющему свой шаг по времени / свою невязку и начинают принимать минимальный шаг по времени / максимальную невязку;
- управляющий процессор по очереди принимает у остальных процессоров информацию и одновременно определяет минимальный шаг по времени / максимальную невязку;
- управляющий процессор по очереди рассылает всем процессорам полученные данные.

В итоге синхронизация работы процессоров, связанная с необходимостью обмена данными, выполняется три раза за итерационный шаг. Первоначально процессоры обмениваются данными перед началом каждой итерации: передаются значения газодинамических переменных на граничных ячейках. Второй обмен данными происходит в процессе вычисления шага по времени, когда среди полученных по обрабатываемым блокам процессоров значений определяется минимальная величина. Заключительный обмен данными происходит после нахождения величин на новом слое по времени: передаются максимальные значения невязки, по которым определяется критерий окончания счета. Если в расчете вычисляется турбулентная вязкость, то между первым и вторым шагами происходит обмен турбулентной вязкостью в граничных ячейках. Как было отмечено выше, прием данных выполняется без промежуточной буферизации, а передача данных только после их предварительного копирования во вспомогательные массивы. Основная задержка времени возникает, когда все процессоры

ждут завершения расчета на одном из процессоров, потому что без него не могут получить минимальный шаг по времени. В большинстве случаев это происходит при неравномерной нагрузке процессоров.

Исходный текст параллельной программы написан на языке C, для обмена сообщений между процессорами использовались функции из библиотеки MPI.

§4 Результаты расчетов и эффективность параллельной реализации

Параллельный алгоритм использовался для проведения расчетов сверхзвукового обтекания профиля крыла. Профиль описывается формулой $y = \pm 2\bar{c} \cdot \bar{x} \cdot (1 - \bar{x}) \cdot b_0$, где $\bar{c} = \frac{c}{b_0} = 0.03$, c -максимальная толщина профиля, $b_0 = 0.526\text{м}$ - длина профиля. Расчеты выполнены для числа Маха $M=3$ и числа Рейнольдса $Re_1 = 0.8 \cdot 10^6$ под углом атаки 20 градусов. Данная задача в однопроцессорном варианте ранее рассматривалась в работе [10].

Также этот алгоритм был опробован на задаче обтекания обратного уступа с углом наклона 45° . Данные для расчета и геометрия расчетной области взяты из работы [11]. Расчеты выполнены для числа Маха $M=2.85$ и числа Рейнольдса $Re_1 = 32 \cdot 10^6$.

Расчет задач выполнялся на разном числе процессоров с целью оценки эффективности работы параллельного алгоритма. Результаты показывают высокую эффективность параллельного алгоритма. Некоторое снижение эффективности при увеличении числа процессоров объясняется неравномерностью загрузки процессоров. Было замечено, что времена расчетов у одинаковых по размеру блоков заметно отличается.

§4.1 Задача о профиле

Расчетная область разбивается на 30 блоков: два маленьких блока перед профилем, 16 блоков окружают профиль сверху и снизу, 12 блоков за профилем, как показано на *рис.2*. Маленький блок имеет размеры 20×150 , остальные блоки 100×150 . Блоки распределялись по процессорам по возможности равномерно (с учетом сложности картины течения).

В *таблице 1* и на *рис.3* приведены зависимости ускорения и эффективности параллельной программы от числа процессоров. Как видно из результатов, реализация параллельного алгоритма имеет высокую эффективность.

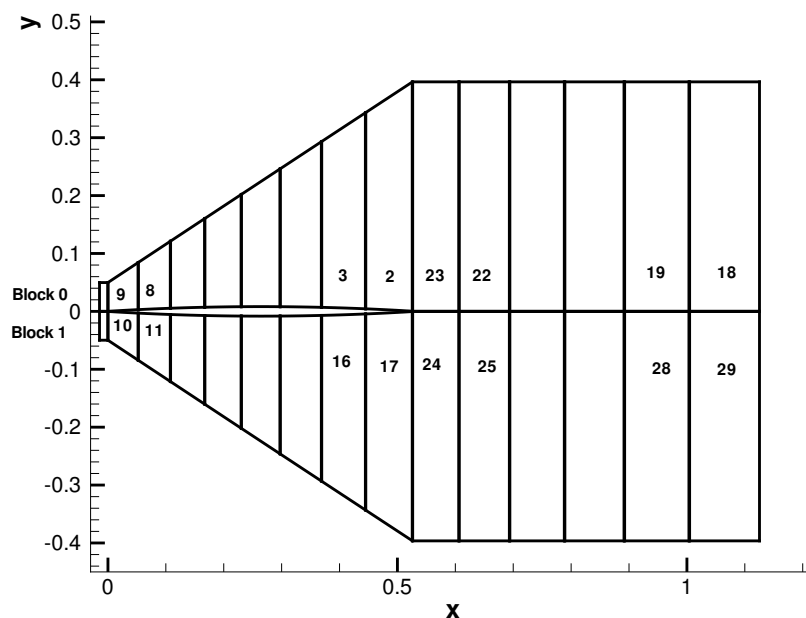


Рис. 2: Область с профилем

Число процессоров	Ускорение	Эффективность, %
1	1.00	100.00
2	1.94	96.91
3	2.84	94.53
4	3.47	86.84
5	4.38	87.56
6	4.99	83.14
8	6.51	81.37
10	8.37	83.73
15	12.39	82.61
29	23.76	81.92

Таблица 1: Ускорение и эффективность

Некоторое снижение ускорения с ростом числа процессоров (*рис.4*) объясняется неравномерностью загрузки процессоров.

На *рис.5* показана зависимость среднего времени расчета на каждом из блоков, времена расчетов отличаются максимально на 23%.

Для оценки накладных расходов, связанных с обменом данными между процессорами, возьмем самый загруженный вариант, когда использовалось 29 процессоров (два маленьких блока считались на управляющем процессоре). За 10000 шагов по всем процессорам среднее время (в секундах) арифметических операций занимает 324.07 с, время простаивания 46.74 с, обмен данными 17.02 с, работа с файлами 1.06 с; На обмен данными ушло примерно 4% времени всего расчета, на арифметические операции

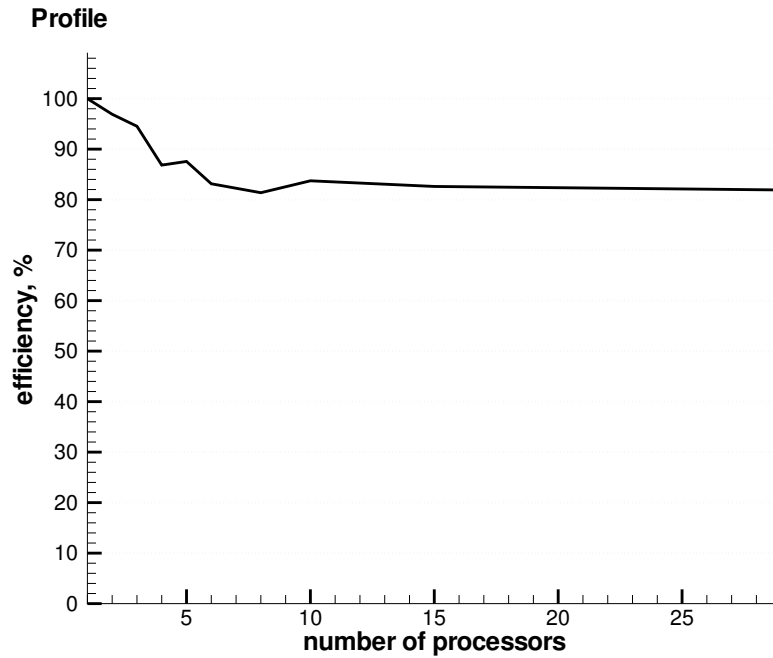


Рис. 3: Зависимость эффективности от числа процессоров для профиля

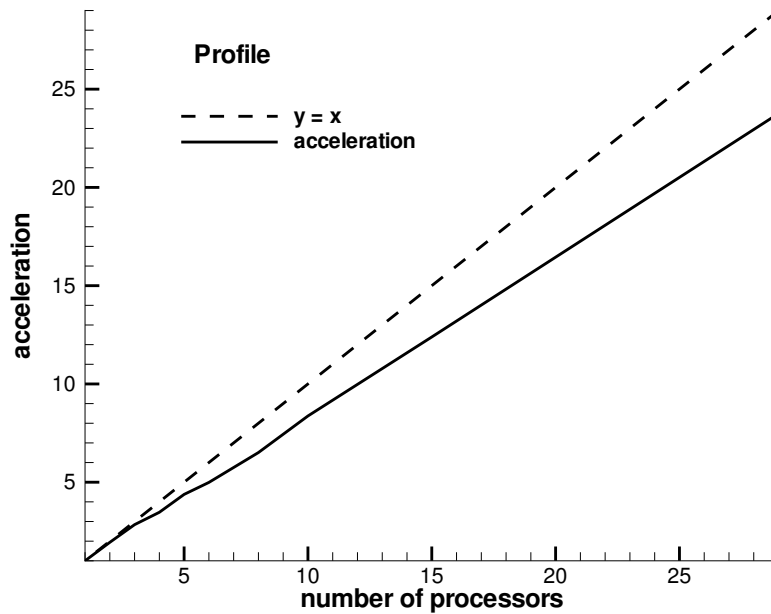


Рис. 4: Снижение ускорения для профиля

около 83%. Меньше всех простаивали и дольше всех считали 2-й и 3-й процессоры со 2-м и 3-м блоками (рис.2, 5): время простаивания 4.8 с и 2.22 с соответственно. В этих блоках присутствует сложная картина течения с образованием вихря, поэтому все процессоры на каждом шаге ждали окончания обработки этих двух блоков.

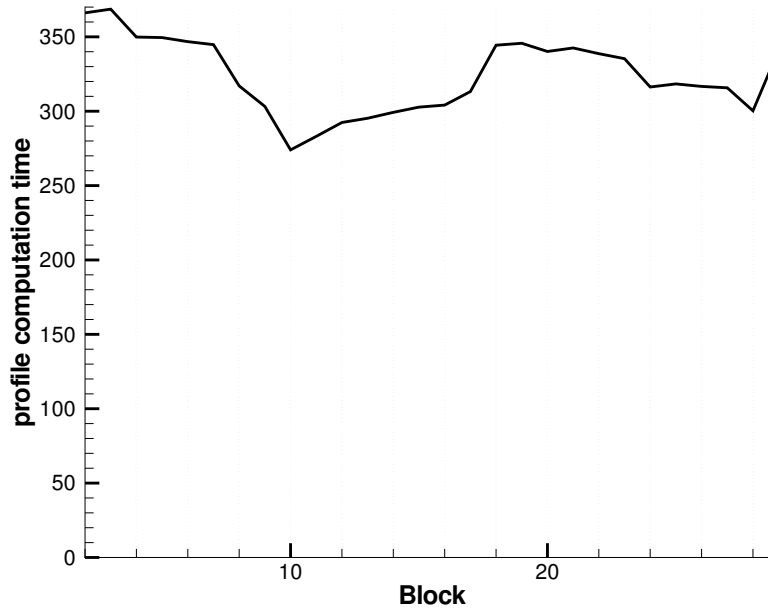


Рис. 5: Время расчета на каждом блоке для профиля

§4.2 Задача об обратном уступе

Расчетная область состоит из 24-х блоков, как показано на *рис.6*, каждый блок содержит 150x300 ячеек.

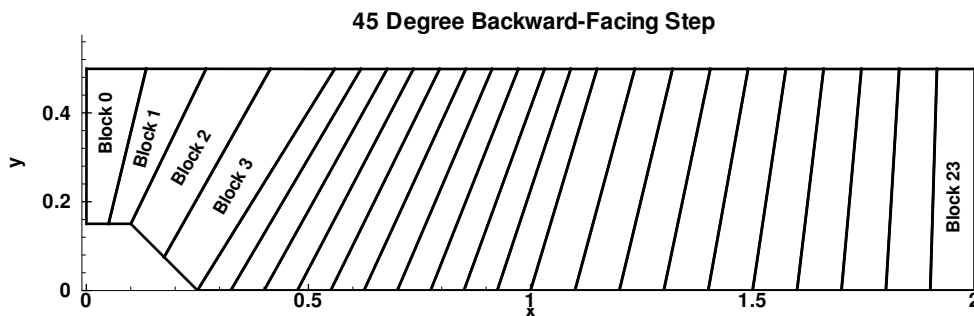


Рис. 6: Область с обратным уступом

Ускорение и эффективность работы программы при расчете обратного уступа представлены в *таблице 2* и на *рис.7*. В случае использования двух процессоров эффективность распараллеливания получилась больше 100%, это можно объяснить особенностью работы кэша процессора.

Снижение ускорения с ростом числа процессоров (*рис.8*) (мин. 93.81%) заметно не так сильно, как в случае с профилем (мин. 81.92%).

На *рис.9* показана зависимость среднего времени расчета на каждом из блоков, времена расчетов отличаются максимально на 18%.

Число процессоров	Ускорение	Эффективность, %
1	1.00	100.00
2	2.19	109.42
3	2.96	98.51
4	3.84	96.00
6	5.68	94.72
12	11.52	96.00
24	22.51	93.81

Таблица 2: Ускорение и эффективность для обратного уступа

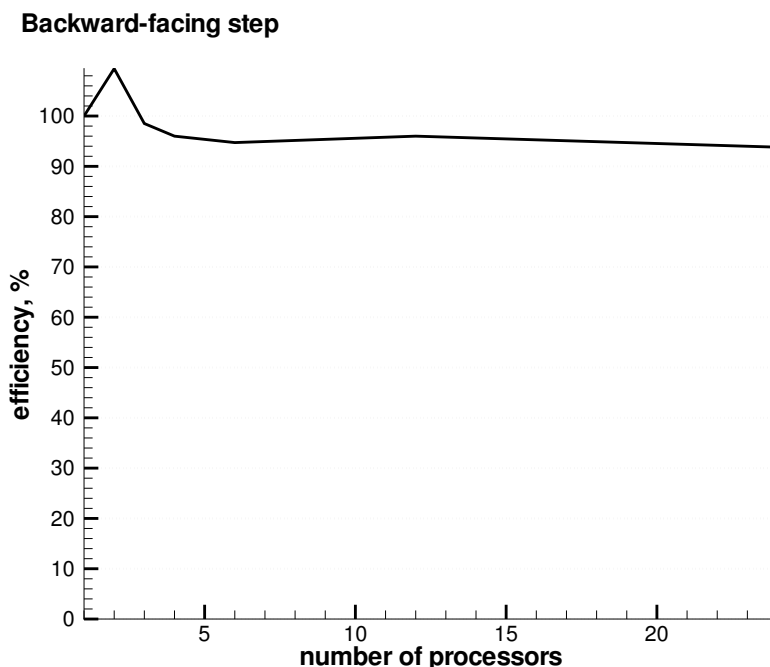


Рис. 7: Зависимость эффективности от числа процессоров для обратного уступа

Результаты показывают, что параллельный алгоритм более эффективно использовался в задаче с обратным уступом, чем в задаче с профилем. Во-первых, различалось количество обменов между блоками: в задаче с обратным уступом у блока есть 2 соседних блока, а в задаче с профилем — 3 соседних блока. В задаче с профилем процессоры были загружены неравномерно, в расчетную область входят два маленьких блока (по размеру маленький блок равен 0.2 большого блока), процессор, на котором они размещаются, заканчивает итерационный шаг раньше всех остальных блоков. К тому же разница во времени расчета у одинаковых по размеру блоков в задаче с профилем (23%) больше, чем в задаче с обратным уступом (18%).

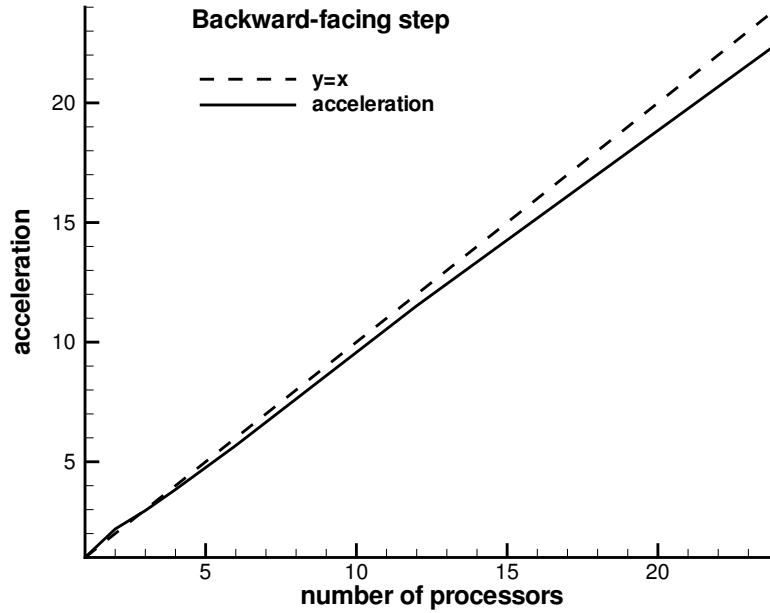


Рис. 8: Снижение ускорения для обратного уступа

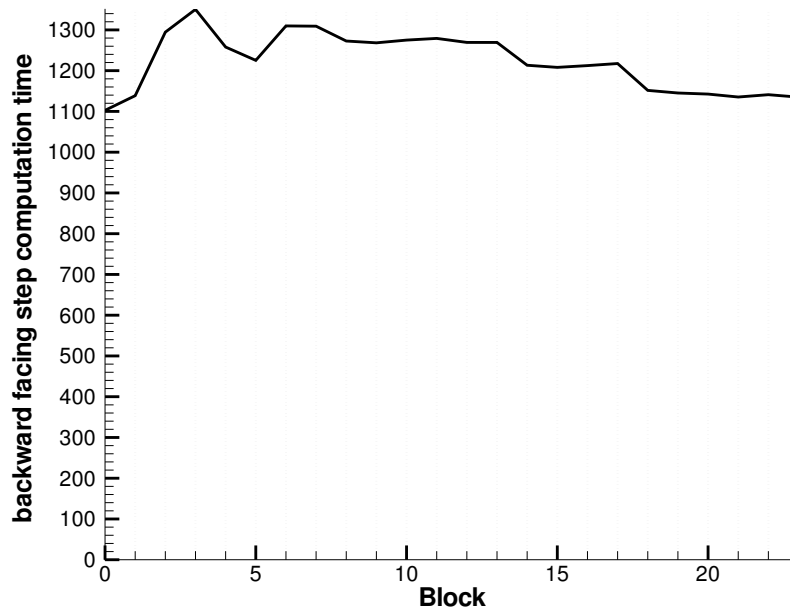


Рис. 9: Время расчета на каждом блоке для обратного уступа

Заключение

Представлен параллельный алгоритм моделирования задач обтекания тел сложной формы. Высокая эффективность параллельной реализации продемонстрирована на примере расчетов обтекания профиля и обтекания об-

ратного уступа. Эффективность распараллеливания во многом зависит от качества балансировки задачи. Балансировка заключается в способе распределения блоков, составляющих расчетную область, по процессорам с учетом количества ячеек в блоке и сложности картины течения в блоке. Также на эффективность распараллеливания влияет количество связей между блоками.

Авторы признательны А.В. Забродину за внимание к работе.

Литература

- [1] Ландау Л.Д., Лифшиц Е.М. Теоретическая физика: Учеб. пособ.: Для вузов. В 10т. Т. VI Гидродинамика. — 5-е изд.,испр. — М.: ФИЗМАТЛИТ,2003. — 736 с.
- [2] С.К. Годунов,А.В. Забродин, М.Я. Иванов, А.Н. Крайко, Г.П. Прокопов "Численное решение многомерных задач газовой динамики", М.: Наука, 1976
- [3] Флетчер К. "Вычислительные методы в динамике жидкостей": Т.2: Пер. с англ. — М.: Мир, 1991
- [4] Гильманов А.Н. "Методы адаптивных сеток в задачах газовой динамики", М.: Наука. Физматлит, 2000
- [5] М.Д.Бродецкий, А.В.Забродин, А.Е.Луцкий, А.М.Харитонов, А.М.Шевченко "Численное и экспериментальное моделирование сверхзвукового течения на подветренной стороне треугольного крыла", 2002 г. Журнал "Аэромеханика и газовая динамика"№1 январь-март 2002
- [6] Л.Г. Лойцянский "Механика жидкости и газа": Учеб. для вузов. — 7-е изд., испр. — М.: Дрофа, 2003. — 840 с.,311 ил., 22 табл. — (Классики отечественной науки).
- [7] <http://parallel.ru/computers/classes.html>
- [8] Воеводин В.В.,Воеводин Вл.В., Параллельные вычисления. — СПб.: БХВ — Петербург, 2002. — 608 с.: ил.
- [9] Sergey V.Yershov and Andrey V.Rusanov „Modification of algebraic turbulence model used in code FlowER (R)“
- [10] А.Е. Луцкий, А.С. Черногузов, Препринт Института прикладной математики им. М.В. Келдыша РАН „Влияние вязкости на параметры сверхзвукового обтекания профиля крыла“, 2004 г., N26.
- [11] http://www.csa.ru/~ignatiev/valid_tur_2.html