

УДК 519.6

Д.В. Деги, А.В. Старченко

**ЧИСЛЕННОЕ РЕШЕНИЕ УРАВНЕНИЙ НАВЬЕ – СТОКСА
НА КОМПЬЮТЕРАХ С ПАРАЛЛЕЛЬНОЙ АРХИТЕКТУРОЙ**

В данной работе представлены алгоритмы численного решения уравнений Навье – Стокса с использованием высокопроизводительной вычислительной техники, такой, как многопроцессорные системы с распределенной памятью и графические ускорители. В качестве тестовой задачи рассматривается классическая задача вычислительной гидродинамики – движение жидкости в прямоугольной каверне.

Ключевые слова: уравнения Навье – Стокса, численное решение, библиотека MPI, технология CUDA.

Создание новых эффективных параллельных вычислительных систем является стратегическим направлением развития компьютерной техники. Это обстоятельство вызвано как ограничением максимально возможного быстродействия обычных последовательных ЭВМ, так и существованием вычислительных задач, для решения которых ресурсов существующих средств вычислительной техники не всегда оказывается достаточным. Такие современные проблемы науки и техники как моделирование климата, геновая инженерия, проектирование интегральных схем, анализ загрязнения окружающей среды, создание лекарственных препаратов, требуют для анализа ЭВМ с производительностью более 1000 миллиардов операций с плавающей точкой в секунду (1 TFLOPS).

К настоящему времени создано большое количество вычислительных алгоритмов, ориентированных на последовательную модель программирования. Однако далеко не всегда удается разработать эффективную параллельную реализацию для многих из них. Поэтому актуальной в настоящее время становится проблема создания новых параллельных алгоритмов решения задач науки и техники на суперкомпьютерах, в частности, для имеющих большие теоретическое и прикладное значение задач гидродинамики.

Основными уравнениями гидродинамики являются уравнения Навье – Стокса. С помощью уравнений Навье – Стокса описываются такие процессы, как движение волн на поверхности жидкости, движение крови по кровеносным сосудам человека и животных, течения в мантии Земли. Задача о движении жидкости в каверне также описывается уравнениями Навье – Стокса, и в ней отражаются следующие характерные черты гидродинамических процессов: конвективная нелинейность, различные соотношения между инерционными и вязкими силами, одновременное существование областей малых и больших градиентов и т.п., поэтому задача о каверне широко распространена в качестве «тестовой» при выборе эффективного численного алгоритма для моделирования гидродинамических процессов [1]. Зачастую для получения приближенного решения с высокой точностью используют сетку с большой плотностью узлов, при этом существенно возрастает время расчета и тем самым становится актуальным применение многопроцессорных ЭВМ для уменьшения времени расчетов.

Физическая постановка задачи

Рассматривается стационарное, рециркуляционное, ламинарное течение несжимаемой, вязкой жидкости в прямоугольной полости (каверне) с верхней стенкой, движущейся в своей плоскости со скоростью U (рис. 1).

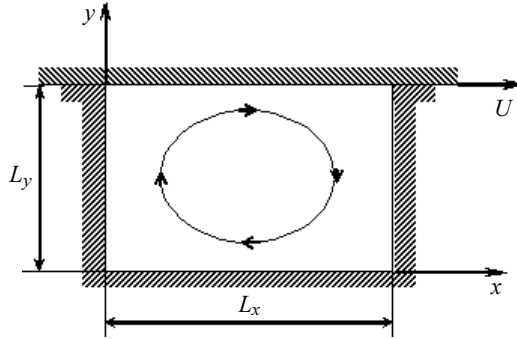


Рис. 1. Движение жидкости в каверне

Жидкость, целиком заполняющая каверну, вовлекается в движение силами вязкости. Действие массовых сил (сил тяжести, инерционных, центробежных или кориолисовых сил) незначительное. Имеет место случай изотермического движения вязкой несжимаемой жидкости с постоянными значениями плотности и коэффициентов вязкости.

Кроме того, требуется выполнение условия прилипания частиц жидкости к твердой стенке. Это означает отсутствие скольжения жидкости по поверхности. Таким образом, выполняется граничное условие равенства нулю скорости жидкости на поверхности неподвижных стенок и совпадение скорости жидкости со скоростью движущейся стенки, с которыми частицы жидкости соприкасаются (равенство касательной компоненты скорости жидкости и скорости стенки).

Математическая постановка задачи

Рассмотренный физический процесс можно описать с помощью следующих уравнений:

- уравнения движения в проекции на ось Ox :

$$\frac{\partial v_x v_x}{\partial x} + \frac{\partial v_y v_x}{\partial y} = -\frac{1}{\rho} \frac{\partial p}{\partial x} + \mu \left(\frac{\partial^2 v_x}{\partial x^2} + \frac{\partial^2 v_x}{\partial y^2} \right);$$

- уравнения движения в проекции на ось Oy :

$$\frac{\partial v_x v_y}{\partial x} + \frac{\partial v_y v_y}{\partial y} = -\frac{1}{\rho} \frac{\partial p}{\partial y} + \mu \left(\frac{\partial^2 v_y}{\partial x^2} + \frac{\partial^2 v_y}{\partial y^2} \right);$$

- уравнения неразрывности:

$$\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} = 0.$$

Здесь $\vec{v} = (v_x, v_y)$ – скорость движения жидкости, p – давление, ρ – плотность жидкости, μ – коэффициент кинематической вязкости.

Граничные условия в соответствии с физической постановкой будут следующими:

- из условия непротекания:

$$x = 0 : v_x = 0; x = L_x : v_x = 0;$$

$$y = 0 : v_y = 0; y = L_y : v_y = 0;$$

- из условия прилипания:

$$x = 0 : v_y = 0; x = L_x : v_y = 0;$$

$$y = 0 : v_x = 0; y = L_y : v_x = U,$$

где L_x и L_y – соответственно длина и ширина каверны, U – скорость движения крышки.

Получение разностной схемы

Приближенное решение ищется на равномерной шахматной сетке Ω_h (рис. 2) с помощью метода конечных объемов [2].

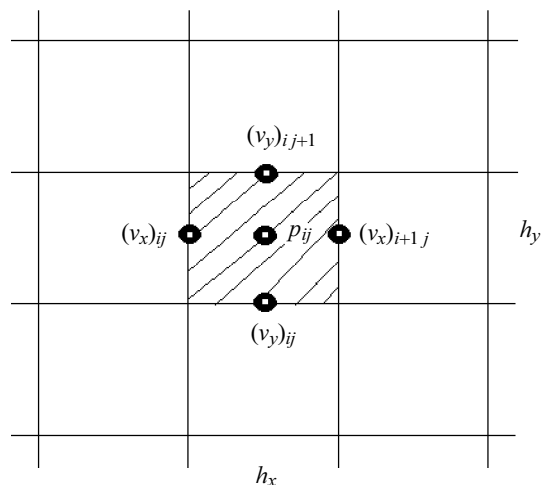


Рис. 2. Фрагмент используемой шахматной сетки (штриховка нанесена на конечный объем)

При получении разностной схемы используется следующее:

1. Значения сеточных неизвестных (v_x, v_y) рассматриваются на соответствующих гранях конечных объемов, а давление (p_{ij}) – в их центрах.

2. Приближенное интегрирование членов дифференциальных уравнений проводится с помощью квадратурной формулы средних прямоугольников.

3. Для устранения возникающей нелинейности в разностной схеме значение одного из множителей следующих произведений: $v_x v_x, v_x v_y, v_y v_y$ – полагается равным значению, полученному на предыдущей итерации.

4. Конвективные члены аппроксимируются с помощью схемы «против потока», диффузионные и производные от давления – центрально-разностной схемой.

5. Значения сеточных функций с дробными индексами $(i \pm 1/2), (j \pm 1/2)$, известные с предыдущей итерации, в точках, не совпадающих с узлами сетки, рассчитываются из значений в ближайших соседних узлах с помощью линейной интерполяции.

В итоге получается, что разностная схема, аппроксимирующая уравнение неразрывности, имеет вид

$$[(v_x)_{i+1j} - (v_x)_{ij}]h_y + [(v_y)_{ij+1} - (v_y)_{ij}]h_x = 0, \quad i = \overline{0, Nx}; j = \overline{0, Ny};$$

разностная схема для проекции уравнения движения по оси Ox

$$aP_{i+1j}(v_x)_{i+1j} = aE_{i+1j}(v_x)_{i+2j} + aW_{i+1j}(v_x)_{ij} + aN_{i+1j}(v_x)_{i+1j+1} + \\ + aS_{i+1j}(v_x)_{i+1j-1} + d_{i+1j}(p_{ij} - p_{i+1j}), \quad i = \overline{0, Nx-1}, j = \overline{0, Ny};$$

разностная схема для проекции уравнения движения по оси Oy

$$bP_{ij+1}(v_y)_{ij+1} = bE_{ij+1}(v_y)_{i+1j+1} + bW_{ij+1}(v_y)_{i-1j+1} + bN_{ij+1}(v_y)_{ij+2} + \\ + bS_{ij+1}(v_y)_{ij} + l_{ij+1}(p_{ij} - p_{ij+1}), \quad i = \overline{0, Nx}, j = \overline{0, Ny-1}.$$

Здесь h_x и h_y – шаги сетки по горизонтальной и вертикальной оси соответственно, $l_{ij+1} = h_x/\rho$ и $d_{i+1j} = h_y/\rho$, N_x и N_y – размер сетки по горизонтали и вертикали соответственно. Коэффициенты $aP_{ij}, aE_{ij}, aW_{ij}, aN_{ij}, aS_{ij}, bP_{ij}, bE_{ij}, bW_{ij}, bN_{ij}, bS_{ij}$ – известные сеточные функции, зависящие от поля скорости, полученного на предыдущей итерации. У этих коэффициентов имеются следующие свойства [2]:

- неотрицательность всех коэффициентов;

- $aP_{ij} \geq aE_{ij} + aW_{ij} + aN_{ij} + aS_{ij}$ и $bP_{ij} \geq bE_{ij} + bW_{ij} + bN_{ij} + bS_{ij}$, причем строгое неравенство выполняется на границе.

Предполагая, что функции $v_x = v_x(x, y)$, $v_y = v_y(x, y)$ и $p = p(x, y)$, представляющие точное решение, имеют достаточное число непрерывно дифференцируемых частных производных, и, используя формулу Тейлора, можно доказать, что погрешность аппроксимации разностной схемой для уравнения неразрывности имеет второй порядок относительно каждого координатного направления, а для уравнений движения – первый.

Алгоритм численного метода решения

В силу нелинейности уравнений Навье – Стокса становится невозможным прямое решение системы и рассматриваемый процесс приобретает итерационный характер. Для совместного решения общей системы из трех систем разностных уравнений был применен алгоритм SIMPLE [2], который можно записать в следующем виде:

1. Задаем поле давления $\{p_{ij}^*\}$;

2. Решаем системы уравнений

$$\begin{cases} aP_{i+1j}(v_x^*)_{i+1j} = \sum_{nb} a_{nb}(v_x^*)_{nb} + d_{i+1j}(p_{ij}^* - p_{i+1j}^*), \\ bP_{ij+1}(v_y^*)_{ij+1} = \sum_{nb} b_{nb}(v_y^*)_{nb} + l_{ij+1}(p_{ij}^* - p_{ij+1}^*) \end{cases} \quad (1)$$

и находим

$$\left\{ (v_x^*)_{ij} \right\} \text{ и } \left\{ (v_y^*)_{ij} \right\},$$

где $\sum_{nb} a_{nb} (v_x^*)_{nb}$ – означает суммирование по соседним узлам сеточного шаблона;

3. Решаем систему уравнений

$$\begin{aligned} & [(v_x^*)_{i+1j} - (v_x^*)_{ij}]h_y + [(v_y^*)_{ij} - (v_y^*)_{i+1j}]h_x + \left[\frac{d_{i+1j}}{aP_{i+1j}} (p'_{ij} - p'_{i+1j}) - \right. \\ & \left. \frac{d_{ij}}{aP_{ij}} (p'_{i-1j} - p'_{ij}) \right]h_y + \left[\frac{l_{ij+1}}{bP_{ij+1}} (p'_{ij} - p'_{ij+1}) - \frac{l_{ij}}{bP_{ij}} (p'_{ij-1} - p'_{ij}) \right]h_x = 0 \end{aligned} \quad (2)$$

и определяем поправку давления $\{p'_{ij}\}$;

4. По соотношениям

$$\begin{cases} (v_x^*)_{i+1j} = (v_x^*)_{i+1j} + d_{i+1j} (p'_{ij} - p'_{i+1j}) / aP_{i+1j}, \\ (v_y^*)_{ij+1} = (v_y^*)_{ij+1} + l_{ij+1} (p'_{ij} - p'_{ij+1}) / bP_{ij+1} \end{cases}$$

вычисляем сеточные функции $\{(v_x)_{ij}\}$ и $\{(v_y)_{ij}\}$, удовлетворяющие разностному уравнению неразрывности;

5. Корректируем давление по формуле $p = p^* + \omega p'$ ($0 < \omega < 1$);

6. Проверяем условия завершения итерационного процесса, если они не выполняются, то принимаем $p^* = p$ и переходим к пункту 2.

В качестве критерия завершения итерационного процесса выбрано выполнение следующих условий:

$$\begin{aligned} 1) \quad \|p'_h\| &= \max_{j=1, \dots, N_y} \sum_{i=1}^{N_x} |(p'_h)_{ij}| \leq \varepsilon; \\ 2) \quad \max_{j=1, N_y} \sum_{i=1}^{N_x} &|[(v_x)_{i+1j} - (v_x)_{ij}]h_y + [(v_y)_{ij} - (v_y)_{i+1j}]h_x| \leq \varepsilon. \end{aligned}$$

Основные вычислительные трудности данного алгоритма – это решение уравнений для компонентов скорости (1), а также для поправки давления (2) на каждой глобальной итерации.

Один из перспективных методов решения систем линейных алгебраических уравнений вида

$$a_{ij}\Phi_{ij} - b_{ij}\Phi_{i-1j} - c_{ij}\Phi_{i+1j} - d_{ij}\Phi_{ij-1} - e_{ij}\Phi_{ij+1} = a_{ij}\Phi_{ij} - \sum_{nb} a_{nb}\Phi_{nb} = f_{ij}, i = \overline{1, n}, j = \overline{1, m};$$

на многопроцессорных вычислительных системах является метод релаксации [3]. Метод релаксации имеет вид

$$\begin{aligned} \Phi_{ij}^{k+1} &= \omega \cdot \left(\frac{b_{ij}\Phi_{i-1j}^{k+1} + c_{ij}\Phi_{i+1j}^k + d_{ij}\Phi_{ij-1}^{k+1} + e_{ij}\Phi_{ij+1}^k + f_{ij}}{a_{ij}} \right) + (1 - \omega)\Phi_{ij}^k, \\ i &= \overline{1, n}, j = \overline{1, m}, k = 0, \dots, K. \end{aligned} \quad (3)$$

Здесь k – номер итерации. Для каждой из полученных разностных схем для определения $\{(v_x)_{ij}\}, \{(v_y)_{ij}\}$ и $\{p_{ij}\}$ выполняется равенство

$$a_{ij} \geq b_{ij} + c_{ij} + d_{ij} + e_{ij} = \sum_{nb} a_{nb},$$

причем $a_{ij} > \sum_{nb} a_{nb} = b_{ij} + c_{ij} + d_{ij} + e_{ij}$ имеет место на границе рассматриваемой области, что является достаточным для сходимости итерационного процесса [3, с. 89].

При $0 < \omega < 1$ получается метод нижней релаксации. Использование метода нижней релаксации замедляет изменение зависимой переменной. С помощью данного метода решались уравнения для скоростей.

При $1 < \omega < 2$ имеет место метод верхней релаксации. Он увеличивает скорость сходимости итерационного процесса. Уравнение для поправки давления решалось данным методом.

Из рекуррентной формулы (3) видно, что производится строго последовательный пересчет значений неизвестной сеточной функции. Иными словами, чтобы рассчитать Φ_{ij}^{k+1} , нужно знать значения $\Phi_{i-1,j}^{k+1}$ и $\Phi_{i,j-1}^{k+1}$, которые, в свою очередь, вычисляются через значения сеточной функции в левом и нижнем узлах сеточного шаблона разностной схемы. А это означает, что вычисления ведутся последовательно в заранее predetermined порядке. Распараллеливание вычислений по рекуррентным формулам представляет определенную проблему для компьютера с параллельной архитектурой, поэтому рассмотрим другой, отличный от используемого в (3), способ обхода узлов сетки, который основывается на их «красно-черном» упорядочивании [3]. В соответствии с этим приемом (рис.3) все узлы сетки делятся на два подмножества $\Omega_h = \Omega_h^R \cup \Omega_h^B$ (сеточные узлы «раскрашиваются» в черные и красные цвета в шахматном порядке), а поскольку при получении разностных схем использовался шаблон «крест», то, как видно из рис. 3, для расчета значения сеточной функции Φ_{ij}^{k+1} в узле черного цвета нужны значения Φ_{nb}^k в узлах красного цвета и наоборот. В таком случае, используя порядок

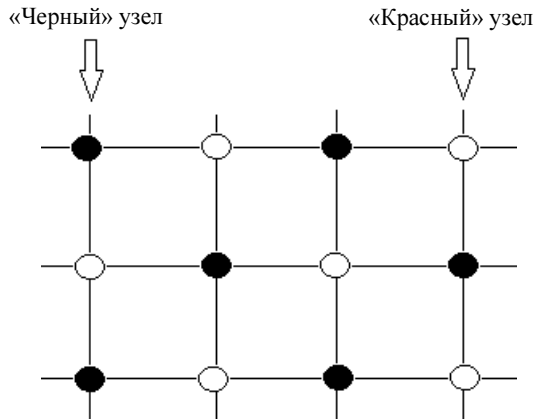


Рис. 3. «Красно-черное» упорядочивание узлов сетки

вычислений значений сеточной функции сначала во всех узлах черного цвета, а затем в оставшихся красного, метод релаксации (3) преобразуется в двухэтапный вычислительный процесс с использованием явных (не рекуррентных) формул:

$$\begin{aligned} (\Phi_{ij}^{k+1})_B &= \omega \cdot \left(\frac{(b_{ij}(\Phi_{i-1,j}^k)_R + c_{ij}(\Phi_{i+1,j}^k)_R + d_{ij}(\Phi_{i,j-1}^k)_R + e_{ij}(\Phi_{i,j+1}^k)_R + (f_{ij})_B)}{a_{ij}} \right) + \\ &\quad + (1-\omega)(\Phi_{ij}^k)_B; (i,j) \in \Omega_h^B; \\ (\Phi_{ij}^{k+1})_R &= \omega \cdot \left(\frac{(b_{ij}(\Phi_{i-1,j}^{k+1})_B + c_{ij}(\Phi_{i+1,j}^{k+1})_B + d_{ij}(\Phi_{i,j-1}^{k+1})_B + e_{ij}(\Phi_{i,j+1}^{k+1})_B + (f_{ij})_R)}{a_{ij}} \right) + \\ &\quad + (1-\omega)(\Phi_{ij}^k)_R; (i,j) \in \Omega_h^R. \end{aligned} \tag{4}$$

Здесь символ «R» означает обход узлов красного цвета, а «B» – черного. При выбранном способе обхода узлов одна итерация в методе релаксации сводится к двум этапам использования явных расчетных формул (типа формул метода Якоби), применяемого сначала к расчету сеточных значений в «черных» узлах, а затем в «красных», при этом объем вычислений на каждой итерации последовательного алгоритма остается прежним.

Другой важной особенностью метода «красно-черного» упорядочивания является то, что количество итераций последовательного и параллельного алгоритмов при проведении вычислений с заданной точностью для задачи одного и того же размера совпадают. Это позволяет утверждать, что при увеличении количества процессоров для решения рассматриваемой задачи, вычислительная нагрузка на каждый из них будет прямо пропорционально уменьшаться.

Время работы последовательной программы

В таблице показано время работы программы, построенной по последовательному алгоритму с «красно-черным» упорядочиванием, в зависимости от размеров используемой сетки. Расчеты проводились на одном вычислительном узле кластера ТГУ СКИФ Cyberia [4]. Использовался компилятор Intel 12.

**Время работы программы при использовании различных сеток
(точность – ε = 0,001)**

Сетка	Итераций	Время, с
128×128	1004	21
256×256	3176	294
512×512	12160	5037
1024×1024	53713	107655

Из таблицы видно, что при уменьшении каждого шага равномерной сетки в 2 раза число итераций увеличивается нелинейно, возрастает объем вычислений и время работы программы при каждом изменении размеров сетки увеличивается более чем на порядок. Тем самым становится актуальным сокращение времени работы программы за счет использования ЭВМ с параллельной архитектурой.

Для проверки правильности работы построенного алгоритма и программы были проведены сравнения расчетных данных с результатами, представленными в [5]. Слева на рис. 4 для квадратной каверны показан профиль продольной скорости

v_x в среднем поперечном сечении $x = L_x$, получаемый в результате вычислений, справа построен график поперечной скорости v_y в среднем продольном сечении $y = L_y$ ($Re = UL_x / \mu = 1000$).

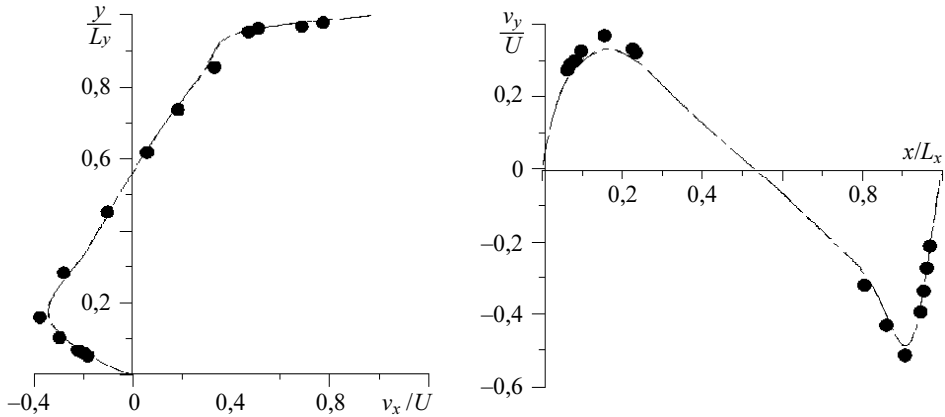


Рис. 4. График продольной скорости (при $x=L_x/2$) и поперечной (при $y=L_y/2$) соответственно, полученный на сетке 256×256 , значки – расчеты [5]

Видно, что даже при использовании для конвективных членов противопотоковой аппроксимации первого порядка на подробной сетке имеет место хорошее согласование с расчетами, полученными на основе использования разностной схемы более высокого порядка на относительно более грубой сетке [5].

Ускорение программы на многопроцессорных ЭВМ

Построение параллельной версии алгоритма SIMPLE осуществлялось на основе принципа геометрической декомпозиции [6] сеточной области, когда вся область исследования делится на равные по площади (или по количеству сеточных узлов) части, вычисления в которых следует проводить одновременно и независимо. Как указывалось выше, основная вычислительная сложность алгоритма SIMPLE – это решение на каждой глобальной итерации систем сеточных уравнений для компонент скорости и поправки давления. Для решения таких систем в данной работе предлагается использовать метод релаксации, поскольку другие более быстро сходящиеся при последовательных вычислениях методы (GMRES, BiCGStab, CG и др.) при параллельной реализации на основе метода геометрической декомпозиции показывают увеличение общего объема вычислительной работы (числа итераций для обеспечения сходимости с заданной точностью) при возрастании количества применяемых в вычислениях процессоров по сравнению с последовательной версией [7].

В данной работе использовался более эффективная по сравнению с одномерной двумерная декомпозиция сеточной области [8], когда вся расчетная область делится на двумерные блоки одинаковых размеров (рис. 5), в каждом из которых значения сеточных функций вычисляются одновременно и независимо по явным формулам (4). Однако для обеспечения таких идеально параллельных вычислений необходимо обеспечить каждую подобласть дополнительными значениями сеточ-

ных функций, которые принадлежат узлам из соседних по декомпозиции подобластей, но однако же необходимы для вычислений в соответствии с выбранным шаблоном «крест». Такая декомпозиция называется декомпозицией с перекрытием и ее реализация требует обменов «приграничных» значений сеточной функции на каждой итерации метода релаксации. Для получения эффективного параллельного алгоритма необходимо, чтобы затраты времени на передачу данных между соседними подобластями были существенно меньше времени вычислений на каждой итерации.

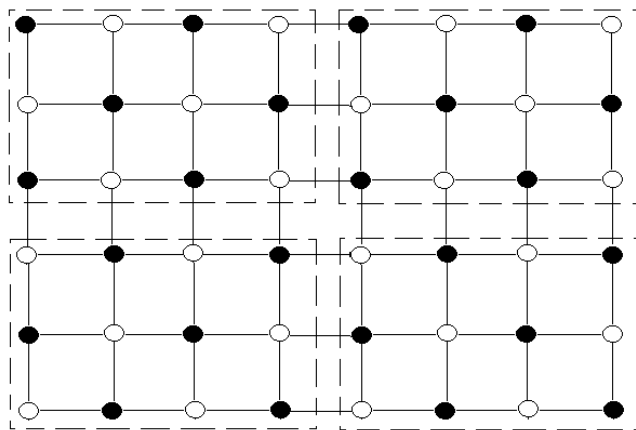


Рис. 5. Геометрическая декомпозиция данных

Разработанный параллельный алгоритм был реализован на параллельных ЭВМ с различной архитектурой: вычислительный кластер ТГУ СКИФ Cyberia [4] и сервер с видеокартой NVIDIA GTX260, которая включает 192 процессора с небольшой тактовой частотой. При создании параллельной программы для кластера использовался стандарт Message Passing Interface [8], а для сервера с графическими процессорами – технология Compute Unified Device Architecture [9]. Параллельная версия программы для каждого случая получается на основе последовательной, при этом применение библиотеки MPI изменяет код последовательной программы лишь частично, в то время как для технологии CUDA требуется полное перестроение последовательной программы.

На рис. 6 показано ускорение (отношение времени работы последовательной программы к времени работы параллельной) параллельных программ, написанных с использованием CUDA или MPI-интерфейса, на различных сетках. MPI-программа была запущена на 32 и 64 вычислительных ядрах кластера СКИФ Cyberia. Из рисунка видно, что ускорение параллельных программ растет с увеличением размеров сетки, что связано с увеличением объема вычислительной работы по сравнению с коммуникационными затратами. Для решения рассматриваемой задачи на сетках с различной плотностью узлов производительность сервера с видеокартой оказалась лучше 32 вычислительных ядер кластера, но хуже чем 64, однако с экономической точки зрения расчеты на видеокарте GTX 260 предпочтительнее использования 64 вычислительных ядер кластера СКИФ Cyberia в силу меньшего энергопотребления и стоимости.

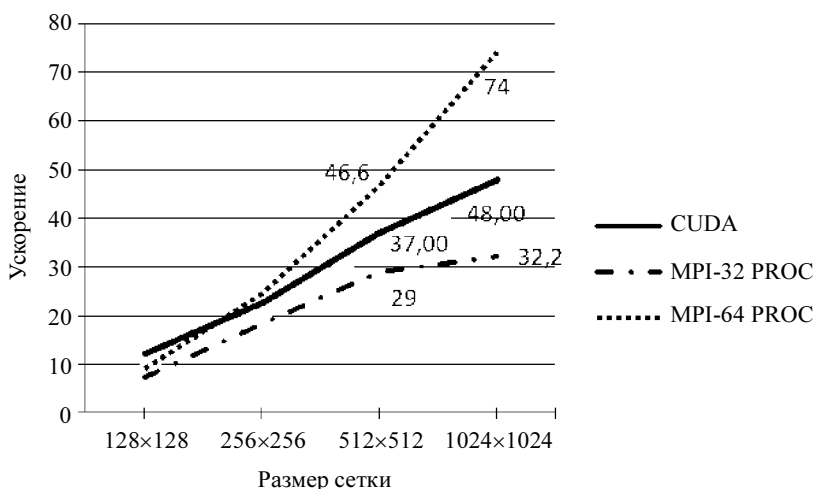


Рис. 6. Ускорение параллельных программ на различных сетках

На сетке 1024×1024 время расчета последовательной программы составляет около 30 часов, согласно рис. 6, время работы программы с использованием сервера с видеокартой NVIDIA GTX260 сократилось до ~ 37 мин, и до ~ 24 мин с использованием 64 вычислительных ядер кластера СКИФ Cyberia.

Заключение

Для решения задач вычислительной гидродинамики разработаны версии алгоритма SIMPLE, ориентированные на использование высокопроизводительной вычислительной техники с параллельной архитектурой, а именно многопроцессорного linux-кластера с распределенной памятью или сервера с графическими ускорителями. В основе построенных параллельных алгоритмов лежит применение принципа неодноточечной геометрической декомпозиции, «красно-черного» упорядочивания при обходе узлов сетки и метода релаксации для решения сеточных уравнений. В результате полученные параллельные алгоритмы численного решения уравнений Навье – Стокса обладают замечательным свойством прямо пропорционального уменьшения числа арифметических операций, выполняемых одним процессором/ядром, при увеличении общего количества используемых вычислительных процессоров/ядер. При решении задачи о течении вязкой несжимаемой жидкости в каверне с движущейся верхней крышкой полученное свойство параллельных алгоритмов позволило обеспечить ускорение в вычислениях на сетках с более чем 10^6 узлов в несколько десятков раз. Кроме того, анализ результатов вычислений показал высокую эффективность построенного параллельного алгоритма на графических процессорах, что существенно расширяет возможности при численном исследовании задач механики жидкости и газа.

ЛИТЕРАТУРА

1. Лойцянский Л.Г. Механика жидкости и газа: учебник для вузов. 6-е изд., перераб. и доп. М.: Наука, 1987. 840 с.
2. Патанкар С. Численные методы решения задач теплообмена и динамики жидкости. М.: Энергоатомиздат, 1984. 124 с.

3. *Ортега Дж.* Введение в параллельные и векторные методы решения линейных систем / под. ред. Х.Д. Икрамова. М.: Мир, 1991. 367 с.
4. <http://skif.tsu.ru>
5. U., K.N. Ghia and C.T. Shin. High-Resolutions for incompressible flow using the Navier-Stokes equations and a multigrid method // J. Comp. Phys. 1982. V. 48. P. 387–411.
6. *Богословский Н.Н., Есаулов А.О., Старченко А.В.* Параллельная реализация алгоритма вычислительной гидродинамики SIMPLE // Сибирская школа семинар по параллельным вычислениям. Томск: Изд-во Том. ун-та, 2002. С. 118–124.
7. *Старченко А.В., Данилкин Е.А.* Параллельная реализация численного метода решения системы уравнений Навье – Стокса при моделировании крупных вихрей турбулентных течений // Вестник Новосибирского государственного университета. Серия: Информационные технологии. Новосибирск: НГУ, 2009. Т. 7. № 2. С. 49–61.
8. *Беликов Д.А. Говязов И.В. и др.* Высокопроизводительные вычисления на кластерах: учеб. пособие / под ред. А.В. Старченко. Томск: Изд-во Том. ун-та, 2008. 198 с.
9. *Боресков А.В., Харламов А.А.* Основы работы с технологией CUDA. М.: ДМК Пресс, 2010. 232 с.

Статья поступила 13.02.2012 г.

Degi D.V., Starchenko A.V. NUMERICAL SOLUTION OF NAVIER-STOKES EQUATIONS ON COMPUTERS WITH PARALLEL ARCHITECTURE. In this paper, we numerically solve the Navier-Stokes equations using high-performance computing technology, such as systems with distributed memory and graphic accelerators. As a test problem, we consider the classical problem of computational fluid dynamics, namely, the motion of a fluid in a rectangular cavity.

Keywords: Navier-Stokes equations, OpenMP system, MPI library, CUDA technology.

DEGI Dmitrii Vladimirovich (Tomsk State University)

E-mail: dimadegi@sibmail.com

STARCHENKO Alexandr Vasil'evich (Tomsk State University)

E-mail: starch@math.tsu.ru