Численный метод решения уравнений Навье-Стокса на многопроцессорных вычислительных системах[1]

А. О. Есаулов, А. В. Старченко
Томский государственный университет, Томск

Во всех современных методах решения уравнений гидродинамики более 50 % вычислительных затрат алгоритмов занимает интегрирование уравнения Пуассона для определения давления (в переменных «скорость-давление») или функции тока (в переменных «функция тока-завихренность»). В связи с этим в данной работе построен новый эффективный метод решения уравнений Навье-Стокса, в котором на конечно-разностном уровне проведено исключение сеточной функции давления.

Рассматривается система уравнений Навье-Стокса, описывающая изотермическое движение вязкой несжимаемой жидкости [1, 2] в двумерной области:

- уравнение неразрывности:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0; \tag{1}$$

- уравнения движения:

$$\begin{aligned} u\frac{\partial u}{\partial x} + \nu\frac{\partial u}{\partial y} &= \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) - \frac{1}{\rho}\frac{\partial p}{\partial x}; \\ u\frac{\partial v}{\partial x} + \nu\frac{\partial v}{\partial y} &= \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) - \frac{1}{\rho}\frac{\partial p}{\partial y} \end{aligned} \tag{2}$$

- уравнения состояния:

$$\nu = \text{const}, \ \rho = \text{const} \tag{3}$$

где u, v — горизонтальная и вертикальная проекции вектора скорости; ν — коэффициент кинематической вязкости; ρ — плотность; p — давление.

[1] Работа выполнена при финансовой поддержке РФФИ, грант № 07-05-01126.

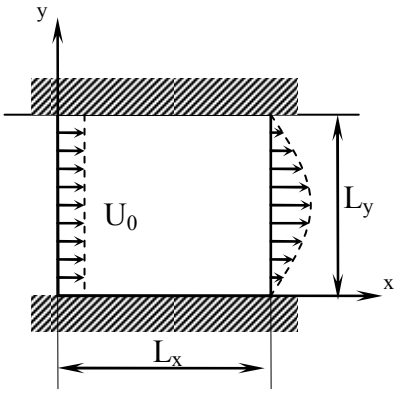


Рис. 1. Канал

Предлагаемый алгоритм решения двумерных уравнений гидродинамики строится на примере ламинарного внутреннего течения вязкой несжимаемой жидкости в прямоугольной области. Требуется выполнение условий прилипания частиц жидкости на твердой стенке и непротекания на ограничивающих течение твердых поверхностях. Имеет место простейший случай изотермического движения вязкой несжимаемой жидкости с постоянными значениями плотности и коэффициента вязкости. Граничные условия в данном случае задаются следующим образом:

$$\begin{aligned} &x = 0 \ : \ \mathrm{u} = U_o; \mathrm{v} = 0; \ \ x = \mathrm{L_x} \ : \ \frac{\partial \mathrm{u}}{\partial \mathrm{x}} = 0; \mathrm{v} = 0; \\ &y = 0 \ : \ \mathrm{u} = \mathrm{v} = 0; \ \ y = \mathrm{L_y} \ : \mathrm{u} = 0; \mathrm{v} = 0; \end{aligned} \qquad (4)$$

Получение конечно-разностного аналога исходной системы дифференциальных уравнений (1)—(4) производится методом конечного объема на шахматной сетке. В центре каждого конечного объема рассчитывается давление, а на его гранях — компоненты скорости [3]. Путем интегрирования по конечному объему получаются конечно-разностные аналоги уравнений (1)—(2).

При аппроксимации конвективных членов используется противопотоковая разность первого порядка, диффузионных членов — центрально-разностная схема. При интегрировании по смещенным конечным объемам системы (2) получаются сеточ-

ные уравнения, содержащие (u, v)-переменные, а также разности значений функции давления в соседних конечных объемах [3]:

$$ae_{i+1j}\left(u_{i+1j}-u_{i+2j}\right)+aw_{i+1j}\left(u_{i+1j}-u_{ij}\right)+an_{i+1j}\left(u_{i+1j}-u_{i+1j+1}\right)+$$
$$+as_{i+1j}\left(u_{i+1j}-u_{i+1j-1}\right)=hy_j\cdot\left(p_{ij}-p_{i+1j}\right); \quad (5a)$$
$$i=1,Nx-2;\quad j=1,Ny-1;$$

$$be_{ij+1}\left(v_{ij+1}-v_{i+1j+1}\right)+bw_{ij+1}\left(v_{ij+1}-v_{i-1j+1}\right)+bn_{ij+1}\left(v_{ij+1}-v_{ij+2}\right)+$$
$$+bs_{ij+1}\left(v_{ij+1}-v_{ij}\right)=hx_i\cdot\left(p_{ij}-p_{ij+1}\right); \quad (5b)$$
$$i=1,Nx-1;\quad j=1,Ny-2;$$

$$ae_{i+1j+1}\left(u_{i+1j+1}-u_{i+2j+1}\right)+aw_{i+1j+1}\left(u_{i+1j+1}-u_{ij+1}\right)+$$
$$+an_{i+1j+1}\left(u_{i+1j+1}-u_{i+1j+2}\right)+$$
$$+as_{i+1j+1}\left(u_{i+1j+1}-u_{i+1j}\right)=hy_{j+1}\cdot\left(p_{ij+1}-p_{i+1j+1}\right); \quad (5c)$$
$$i=1,Nx-2;\qquad j=0,\ Ny-2;$$

$$be_{i+1j+1}\left(v_{i+1j+1}-v_{i+2j+1}\right)+bw_{i+1j+1}\left(v_{i+1j+1}-v_{ij+1}\right)+$$
$$+bn_{i+1j+1}\left(v_{i+1j+1}-v_{i+1j+2}\right)+$$
$$+bs_{i+1j+1}\left(v_{i+1j+1}-v_{i+1j}\right)=hx_{i+1}\cdot\left(p_{i+1j}-p_{i+1j+1}\right); \quad (5d)$$
$$i=0,Nx-2;\qquad \mathrm{j}=1,Ny-2.$$

Давление исключается при комбинации (5a) – (5b) – (5c) + (5d) дискретных аналогов уравнений движения для четырех соседних контуров. С помощью уравнения неразрывности из получившегося соотношения также можно исключить вертикальную компоненту скорости v. В итоге получается система сеточных уравнений:

$$A_{i+1j}u_{i+1j}+B_{i+1j}u_{i+2j}+C_{i+1j}u_{ij}+D_{i+1j}u_{i+1j+1}+E_{i+1j}u_{i+1j-1}+$$
$$F_{i+1j}u_{i+2j+1}+G_{i+1j}u_{ij+1}+H_{i+1j}u_{i+1j+2}+\sum_{m=-1}^{3}\sum_{k=1}^{j}I^m_{i+1j}u_{i+m,k}=0; \quad (6)$$
$$i=1,Nx-2,\ j=1,Ny-2.$$

В качестве алгоритма решения исходной задачи предлагается следующая последовательность действий:

1. Задание начальных распределений и граничных условий.

2. Расчет коэффициентов A_{i+1j}, B_{i+1j}, C_{i+1j}, D_{i+1j}, E_{i+1j}, F_{i+1j}, G_{i+1j}, H_{i+1j}.

3. Расчет горизонтальной компоненты скорости u из СЛАУ (6). (7)

4. Расчет вертикальной компоненты v.

5. Проверка невязки и ошибки; если они велики, то возврат к п. 2.

Пункт 4 алгоритма представляет собой расчет компоненты скорости v по столбцам методом прогонки из следующих сеточных уравнений:

$$v_{ij+1} - 2v_{ij} + v_{ij-1} = \frac{\left(hy_j\left(u_{i+1j} - u_{ij}\right) - hy_{j-1}\left(u_{i+1j-1} - u_{ij-1}\right)\right)}{hx_i}; \quad (8)$$

$$i = 1, Nx;\ \ j = 2, Ny - 1;$$

Полученная система (6) удовлетворяет одновременно уравнениям движения и неразрывности и связывает значения сеточной функции скорости во всей расчетной области. Однако данная система уравнений является плохообусловленной и при больших размерностях разностной задачи в общем случае требует разработки специальных методов решения.

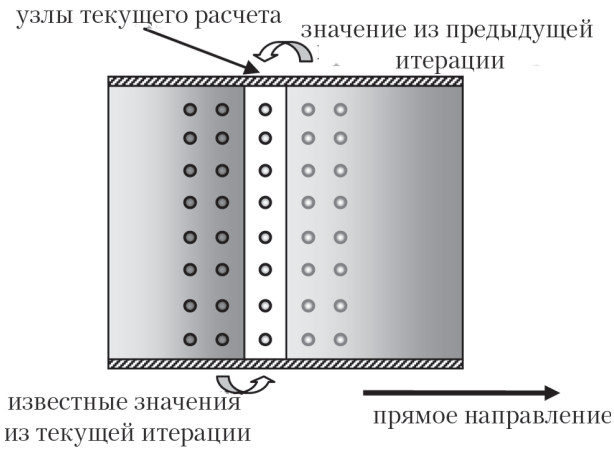


Рис. 2. Схема маршевого метода

В данной работе для отрывного течения с преобладающим направлением потока построен итерационный «маршевый» алгоритм расчета, позволяющий в несколько раз быстрее получить результат по сравнению с известным алгоритмом SIMPLE Патанкара для решения уравнений Навье-Стокса [3]. Суть этого алгоритма заключается в последовательном расчете по столбцам (рис. 2) значений сеточной функции скорости u во всей расчетной области (значения переменных из соседних столбцов берутся из текущей или предыдущей итерации). Таким образом, пункт 3 алгоритма (7) для нахождения поля скорости u по формуле (6) заменяется на схему, дающую приближение к точному решению этих сеточных уравнений. Подобная замена замедляет сходимость алгоритма в целом, но упрощает его программную реализацию и требования к вычислительным ресурсам.

Проверка правильности работы предложенного метода и адекватности построенной модели проводилось путем сравнения результатов расчетов с приближенным аналитическим решением в случае развивающегося течения в начальном участке канала, а также на примере сопоставления численного и физического экспериментов [5] течения жидкости в канале с внезапным расширением (рис. 3). Во всех случаях наблюдается хорошее согласование результатов, что говорит о корректности проводимых расчетов.

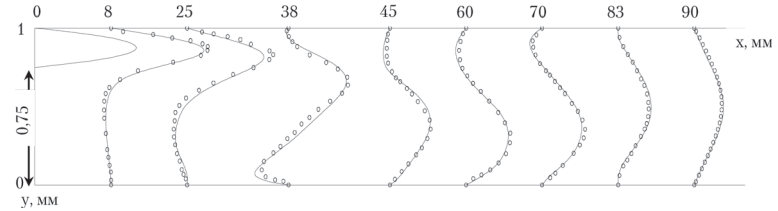


Рис. 3. Профили продольной скорости для канала с расширением. Re = 142, U_0 = 43,9 см/с, ○ — экспериментальные данные, ∿ — результаты расчетов

В табл. 1 приведено время расчетов последовательных алгоритмов на различных сетках, проведенных на одном вычислительном узле кластера ТГУ СКИФ Cyberia.

Предложенный «маршевый» алгоритм эффективно распараллеливается на многопроцессорных системах с распределенной памятью с использованием конвейерной схемы организации

параллельных вычислений. Учитывая специфику применяемого алгоритма, сложность вычислительного шаблона, а также тот факт, что для нахождения значений в одном столбце горизонтальной скорости применяется прямой метод решения СЛАУ, эффективной может быть только одномерная декомпозиция расчетной области поперек направления доминирующего течения. На рис. 4 схематично изображена одномерная декомпозиция, а также значения сеточной функции *u*, которыми должны обмениваться соседние параллельные процессы.

Таблица 1

Сравнение времени счета алгоритмов на различных сетках (в секундах)

	16×16	**32×32**	**48×48**	**64×64**	**80×80**
Simple	0,25	2,14	7,42	29,91	109,59
Маршевый метод	0,00648	0,22	1,6	5,99	16,63

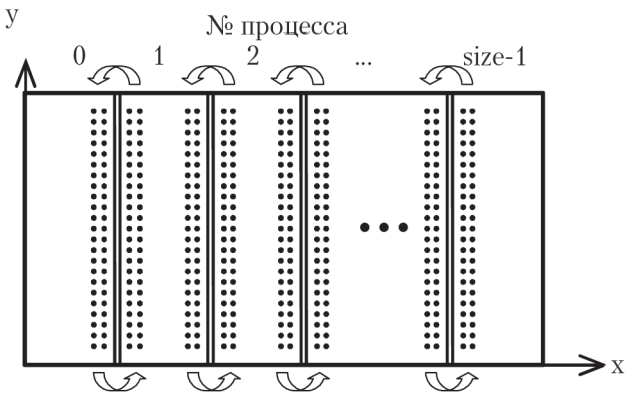


Рис. 4. Схема одномерной декомпозиции расчетной области

В ходе одного шага работы параллельного алгоритма каждый процесс проводит «маршевый» просчет горизонтальной компоненты скорости, обменивается граничными значениями с соседними процессами, затем вычисляет вертикальную компоненту скорости и пересчитывает коэффициенты системы (6). Если в качестве начального приближения взять нулевое поле скоростей, т. е. отсутствие движения во всем канале, то расчеты в каждой по-

добласти будут давать также нулевое поле скоростей в случае нулевых граничных условий на левой и правой границе. Так как на выходе канала заданы мягкие граничные условия, то на распространение возмущения скорости влияет только граничное условия на входе канала. Таким образом, за первый шаг итерационное приближение получает только один процесс, за второй шаг — два, за третий шаг — три и т. д. Шаги работы параллельной программы до момента, когда возмущение достигнет последней расчетной подобласти соответствует загрузке конвейера. Следующим этапом является параллельный расчет приближенного решения каждым процессом в своей подобласти при равномерном распределении вычислений между всеми вычислительными процессами.

Подобная «конвейерная» организация параллельных вычислений практически повторяет все операции исходного последовательного алгоритма и требует близкого числа глобальных итераций. Замедление ускорения расчетов связано лишь с затратами на обмены данными между процессами. Таким образом, итерационный в силу нелинейности исходной задачи характер нахождения решения позволил создать эффективный параллельный алгоритм.

Тестирование параллельной программы проводилось на кластере ТГУ СКИФ Cyberia на примере задачи о течении жидкости в канале с внезапным расширением и показало значительное ускорение расчетов. На рис. 5. приведен график ускорения работы параллельного алгоритма на различных сетках. Из графика можно сделать вывод о том, что распараллеливание эффективно даже при небольших размерах сетки. Ускорение при $N = 100$ получилось даже несколько больше, чем при $N = 200$, что объясняется, возможно, более эффективным использованием КЭШ-памяти вследствие небольшого объема вычислительной информации на каждом вычислительном процессе.

В целом заметна закономерность, заключающаяся в том, что с увеличением размерности сетки ускорение будет все ближе приближаться к идеальному (штриховая линия на рис. 5). Действительно, с увеличением загрузки каждого процесса доля межпроцессорных обменов в общем объеме временных затрат уменьшается, в то же время параллельные программы практически не требуют дополнительных итераций метода в сравнении с последовательной версией.

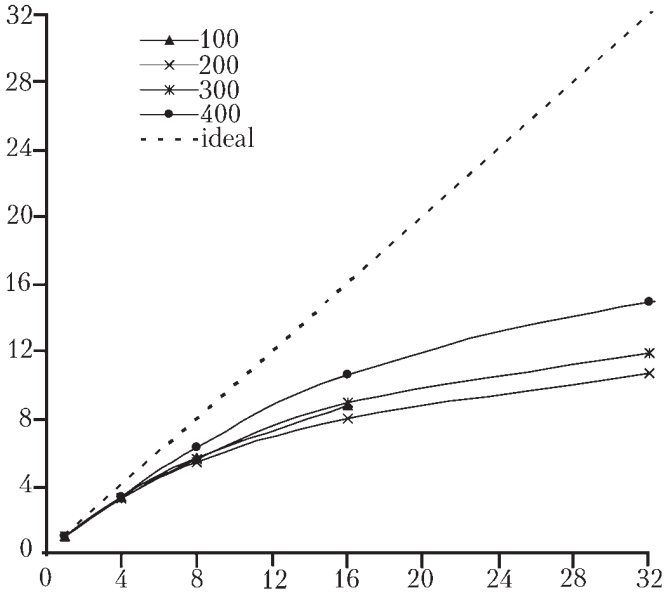


Рис. 5. Ускорение работы параллельной программы в зависимости от числа процессов

Таким образом, применение конвейерной организации параллельных вычислений на многопроцессорных вычислительных системах с распределенной памятью эффективно для разностной задачи любой размерности и дает значительное ускорение в решении задач гидродинамики с доминирующим направлением течения.

Литература

1. **Андерсон Д., Таннехилл Дж., Плетчер Р.** Вычислительная гидромеханика и теплообмен: В 2 т. / Пер. с англ. — М. : Мир, 1990. — Т. 2. — 392 с.

2. **Лойцянский Л. Г.** Механика жидкости и газа: Учеб. для вузов. — изд. 6-е, перераб. и доп. — М. : Наука. Гл. ред. физ.-мат. лит., 1987. — 840 с.

3. **Патанкар С.** Численные методы решения задач теплообмена и динамики жидкости / Пер. с англ. — М. : Энергоатомиздат, 1984. — 152 с.

4. **Ghia U., Ghia K. N., Shin C. T.** High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method // J. Comp. Phys. — 1982. — Vol. 48. — P. 387—411.

5. **Tropea C. D., Gackstatter R.** The Flow Over Two-Dimensional Surface-Mounted Obstacles at Low Reynolds Numbers // Journal of Fluids Engineering. — 1985. — № 4. — P. 489—494.