

**А.А. НЕСТЕРОВ**

## **МЕТОД КОРРЕКЦИИ ДЛЯ ПАРАЛЛЕЛИЗАЦИИ ЧИСЛЕННЫХ МОДЕЛЕЙ ГИДРОДИНАМИКИ ВОДОЁМОВ СО СВОБОДНОЙ ПОВЕРХНОСТЬЮ**

---

**Abstract:** This paper proposes a technique using MPI and domain decomposition to transform serial algorithms of numerical models for hydrodynamics simulations in the water reservoirs with free surface into parallel algorithms. The advantage of the proposed technique is a comparatively simple realization due to the necessity of only additional correction procedures instead of significant transformations of existing serial program code. The impacts of various factors on the method of efficiency are studied in terms of the speedup of computations with the increase of number of utilized processors.

**Key words:** hydrodynamics models of water reservoirs with free surface, parallel computations, MPI, domain decomposition, POM, THREETOX.

**Анотація:** В роботі запропоновано метод перетворення послідовних алгоритмів моделей гідродинаміки резервуарів з вільною поверхнею в паралельні, використовуючи MPI та розбивку розрахункової області на підобласті, перевагою якого є відносна простота реалізації, обумовлена потребою тільки у додаткових процедурах корекції замість значних перетворень існуючих програм послідовного розрахунку. Досліджується вплив різноманітних факторів на ефективність методу в термінах зменшення тривалості розрахунків зі збільшенням числа застосованих процесорів.

**Ключові слова:** моделі гідродинаміки водоймищ із вільною поверхнею, паралельні обчислювання, MPI, розбиття розрахункової області, POM, THREETOX.

**Аннотация:** В работе предложен метод преобразования последовательных алгоритмов моделей гидродинамики резервуаров со свободной поверхностью в параллельные, используя MPI и разбивку расчётной области на подобласти, преимуществом метода является относительная простота реализации, обусловленная потребностью только в дополнительных процедурах коррекции вместо значительных преобразований существующих программ последовательного расчёта. Исследуется влияние различных факторов на эффективность метода в терминах уменьшения длительности параллельных расчётов с увеличением числа используемых процессоров.

**Ключевые слова:** модели гидродинамики водоёмов со свободной поверхностью, параллельные вычисления, MPI, разбивка расчётной области, POM, THREETOX.

### **1. Введение**

Предметом интереса задач численного моделирования гидродинамики водоёмов и качества воды, как и многих других задач геофизической гидродинамики [1], являются не только «основные» переменные гидродинамики, такие как плотность, давление и скорость, а и «дополнительные»: температура, солёность, концентрации взвешенных частиц, загрязняющих веществ и других химико-биологических параметров, описываемых адвективно-диффузионными уравнениями с источниками. Во многих случаях уравнения являются взаимосвязанными, а решение системы не может быть разбито на последовательное решение независимых уравнений или их подсистем. К примеру, градиенты плотности, обусловленные температурой и солёностью, приводят к возникновению бароклинных сил, порождающих циркуляцию, которая в свою очередь влияет на перенос тепла и солёности. Однако большое число зависимых друг от друга переменных и высокое численное разрешение, необходимое для дискретизации уравнений в зонах, где существуют большие градиенты переменных, приводят к понижению скорости расчётов, что в особенности неприемлемо для прогностических моделей.

Параллелизация вычислений является эффективным средством уменьшения их длительности. Более того, последние достижения в развитии вычислительной техники показывают, что для повышения производительности используется именно параллелизация на различных уровнях вместо увеличения тактовой частоты процессоров: от выполнения инструкций различными

исполнительными блоками одного и того же процессора до использования сетей [2]. Поскольку многие модели водоёмов со свободной поверхностью, такие как POM [3] и THREETOX [4], были созданы с применением последовательных алгоритмов расчёта, то для уменьшения длительности расчётов и увеличения численного разрешения сеток необходимо их распараллеливание.

В настоящем применяется несколько «технологий» параллелизации и их комбинаций, из которых можно выделить векторизацию, OpenMP, MPI (Message Passing Interface) и PVM (Parallel Virtual Machine). Векторизация [2, 5] относится к выполнению одной и той же операции над несколькими операндами одновременно, однако её применение обуславливается специфическими возможностями процессора, как в [5]. OpenMP [2] используются для высокоэффективной параллелизации расчётов на системах с общей памятью, недостатком которых является высокая стоимость узлов с более чем двумя процессорами. Исключение составляют многоядерные процессоры, но их производительность существенно ограничена пропускной способностью шины данных. В то же время MPI [6] может быть применён как для расчётов на системах с отдельной памятью, так и для расчётов на многопроцессорных и многоядерных системах, а небольшой вычислительный кластер может быть собран на основе широкодоступных процессоров Intel или AMD, соединённых сетевым коммутатором. Поэтому MPI был выбран основой для параллелизации алгоритмов в данной работе.

Одним из наиболее универсальных методов распараллеливания численных алгоритмов решения уравнений гидродинамики является разбивка исходной расчётной области на подобласти (Domain Decomposition). Размерность численной сетки по горизонтали в задачах геофизической гидродинамики обычно значительно превышает число вертикальных слоёв, а необходимость применения неявных схем для численной аппроксимации, например, диффузионных слагаемых в вертикальном направлении, при небольшом числе слоёв приводит к логическому допущению о необходимости хранения всех переменных, описывающих одну и ту же колонну, на вычислительном узле с общей памятью [7]. По этой причине эффективной является только разбивка в горизонтальных направлениях. Она применяется во многих численных моделях геофизической гидродинамики, таких как GESIMA [7], POLCOMS [8, 9], POP [10], TRISULA [11], а также в параллельных версиях POM [12].

Одним из простейших способов реализации метода разбивки расчётной области на подобласти является перекрытие подобластей с обменом необходимой информацией на каждом шаге интегрирования перед непосредственными вычислениями, как осуществлено в [10, 12]. Использование неявных алгоритмов расчёта приводит к необходимости решения систем линейных уравнений, например, для уровня [13, 14], с большой разреженной матрицей. В этом случае разбивка на подобласти приводит к соответствующей разбивке матрицы. При этом применяются разновидности методов Шварца [11] для дискретной параметризации взаимодействия подобластей, а параллельный алгоритм часто строится таким образом, чтобы он сходил к решению последовательного расчёта в результате итераций. Однако недостатками упомянутых методов является то, что они предполагают значительную реструктуризацию программы расчёта, внедрение специальных граничных условий.

В данной работе предлагается метод коррекции, подразумевающий использование только дополнительных подпрограмм, без значительных изменений в существующих алгоритмах последовательного расчёта. Метод применён для параллелизации конечно-разностной модели, использующей разнесенные сетки [15] как с полунявным алгоритмом расчёта уровня поверхности [13, 14], так и с явным алгоритмом [3]. Полученное в результате параллельного расчёта решение является идентичным (в терминах значений переменных в соответствующих ячейках сетки) последовательному расчёту. Преимуществом предложенного метода является то, что параллелизация модели с последовательным расчётом может быть выполнена относительно быстро по сравнению с разработкой новой модели, с вполне приемлемой эффективностью на кластерах, состоящих из ~10-20 процессоров, а также на узлах с общей памятью и многоядерных процессорах.

## 2. Уравнения модели

Метод коррекции для параллелизации алгоритмов расчёта рассматривается далее для решения уравнений, описывающих гидродинамику резервуаров со свободной поверхностью в гидростатическом приближении Буссинеска, дополненных уравнениями переноса для температуры и солёности [3]. Уравнения записаны в декартовой системе координат  $(x, y)$  по горизонтали и  $\sigma$ -системе координат по вертикали ( $\sigma = 0$  на поверхности и  $\sigma = -1$  на дне) [3]. Уравнение неразрывности имеет вид

$$\frac{\partial UD}{\partial x} + \frac{\partial VD}{\partial y} + \frac{\partial \omega}{\partial \sigma} + \frac{\partial \eta}{\partial t} = 0, \quad (1)$$

где  $\eta = \eta(x, y, t)$  – отклонение поверхности от невозмущённого уровня,  $H = H(x, y)$  – глубина при невозмущённой поверхности,  $D = H + \eta$  – полная глубина,  $\omega$  – преобразованная "вертикальная" компонента вектора скорости [3], введённая с целью консервативной формулировки уравнений. Относительно малые отклонения  $\rho' = \rho - \rho_0$  удельной плотности  $\rho = \rho(T, S, P)$  (где  $T$  – температура,  $S$  – солёность и  $P$  – давление) от "стандартной" плотности  $\rho_0 = const$  учитываются в уравнениях количества движения путём расщепления гидростатического давления

$P = gD \int_{\sigma}^0 \rho(\sigma') d\sigma'$  на баротропную и бароклинную составляющие [3]:

$$\frac{\partial DU}{\partial t} + \frac{\partial DU^2}{\partial x} + \frac{\partial DUV}{\partial y} + \frac{\partial U\omega}{\partial \sigma} = -gD \frac{\partial \eta}{\partial x} + F_{BrcX} + fDV + \frac{\partial}{\partial \sigma} \left( \frac{v}{D} \frac{\partial U}{\partial \sigma} \right) + F_{VisX}, \quad (2)$$

$$\frac{\partial DV}{\partial t} + \frac{\partial DUV}{\partial x} + \frac{\partial DV^2}{\partial y} + \frac{\partial V\omega}{\partial \sigma} = -gD \frac{\partial \eta}{\partial y} + F_{BrcY} - fDU + \frac{\partial}{\partial \sigma} \left( \frac{v}{D} \frac{\partial V}{\partial \sigma} \right) + F_{VisY}. \quad (3)$$

Бароклинные компоненты давления рассчитываются как

$$F_{BrcX} = -\frac{gD^2}{\rho_0} \int_{\sigma}^0 \left[ \frac{\partial \rho'}{\partial x} - \frac{\rho'}{D} \frac{\partial D}{\partial x} \frac{\partial \rho'}{\partial \sigma'} \right] d \sigma', \quad F_{BrcY} = -\frac{gD^2}{\rho_0} \int_{\sigma}^0 \left[ \frac{\partial \rho'}{\partial y} - \frac{\rho'}{D} \frac{\partial D}{\partial y} \frac{\partial \rho'}{\partial \sigma'} \right] d \sigma', \quad (4)$$

где  $fV, fU$  – Кориолисовы слагаемые,  $F_{VisX}, F_{VisY}$  – проекции сил вязкости в горизонтальном направлении, которые записываются в приближённой консервативной форме, обоснованной в [3]:

$$F_{VisX} = \frac{\partial}{\partial x} \left( A_h H \frac{\partial U}{\partial x} \right) + \frac{\partial}{\partial x} \left( \frac{A_h H}{2} \left( \frac{\partial U}{\partial y} + \frac{\partial V}{\partial x} \right) \right),$$

$$F_{VisY} = \frac{\partial}{\partial y} \left( A_h H \frac{\partial V}{\partial y} \right) + \frac{\partial}{\partial y} \left( \frac{A_h H}{2} \left( \frac{\partial U}{\partial y} + \frac{\partial V}{\partial x} \right) \right). \quad (5)$$

Коэффициент вязкости  $A_h$  определяется по формуле Смагоринского:

$$A_h = \alpha \Delta x \Delta y \sqrt{\left( \frac{\partial U}{\partial x} \right)^2 + \frac{1}{2} \left( \frac{\partial U}{\partial y} + \frac{\partial V}{\partial x} \right)^2 + \left( \frac{\partial V}{\partial y} \right)^2}, \quad (6)$$

где  $\alpha$  является постоянной ~0,2. Система уравнений дополняется уравнениями переноса для температуры и солёности  $C$ :

$$\frac{\partial DC}{\partial t} + \frac{\partial DUC}{\partial x} + \frac{\partial DVC}{\partial y} + \frac{\partial \omega C}{\partial \sigma} =$$

$$= DS_C + \frac{\partial}{\partial \sigma} \left( \frac{v'}{D} \frac{\partial C}{\partial \sigma} \right) + \frac{\partial}{\partial x} \left( A'_h H \frac{\partial C}{\partial x} \right) + \frac{\partial}{\partial y} \left( A'_h H \frac{\partial C}{\partial y} \right). \quad (7)$$

Коэффициент вертикальной вязкости  $v$  и коэффициент диффузии  $v'$  в вертикальном направлении рассчитываются согласно  $k/\varepsilon$  модели турбулентности [16], в которой перенос турбулентной кинетической энергии  $k$  и диссипации  $\varepsilon$  описывается уравнениями, подобными (7).

На твёрдых границах задаются условия прилипания для воды и условия непроницаемости для температуры и солёности. На открытых границах задаются либо уровни, либо расходы, а также значения температуры и солёности. Потоки тепла и импульса через поверхность рассчитываются исходя из метеорологических условий для каждой приповерхностной ячейки сетки [4].

За исключением баротропных составляющих давления и вязкостных слагаемых в вертикальном направлении  $\sigma$ , слагаемые в уравнениях (2), (3) аппроксимировались в данной модели явными схемами. Явная аппроксимация баротропных составляющих давления обуславливает существенное ограничение на шаг интегрирования по времени, а полунявная, как, например, в [13, 14], приводит к большой системе линейных уравнений. Поэтому в работе рассмотрено построение параллельных алгоритмов как для явного метода с разбиением на так называемые моды [3], так и для полунявного метода [13]. Так как разбивка расчётной области на подобласти осуществляется в горизонтальном направлении, то предметом дальнейшего

рассмотрения будут слагаемые, содержащие пространственные производные по координатам  $x$  и  $y$ . Аппроксимация и алгоритмы расчёта в направлении  $\sigma$  являются такими же, как при последовательном расчёте. Далее также полагается, что дискретизация граничных условий имеет порядок аппроксимации не выше, чем самих уравнений.

### 3. Параллелизация алгоритмов расчёта явных слагаемых

Суть метода коррекции заключается в том, что на первом этапе каждого шага интегрирования предполагается наличие фиктивной твёрдой границы между подобластями, что позволяет использовать граничные условия, свойственные для модели, с последующей коррекцией на втором этапе. Рассмотрим простейший пример одномерной адвективной схемы 1-го порядка вверх по потоку [15] с числом Куранта, равным единице. В этом случае последовательный алгоритм упрощается:

$$C_i^{n+1} = C_{i-1}^n, \quad i = 1, \dots, m, \quad C_1^{n+1} = C_*(t),$$

где  $C_*(t)$  – граничное условие, заданное на левой границе,  $m$  – число узлов сетки.

Разделим теперь всю расчётную область на две подобласти. Пусть  $C_{1,i}$  ( $i = 1, \dots, m_1$ ) и  $C_{2,i}$  ( $i = 1, \dots, m_2$ ) – значения скаляра в узлах левой и правой подобластей, соответственно,  $m_1 + m_2 = m$ . На первом этапе алгоритма для правой подобласти будем предполагать фиктивное граничное условие, например,  $C_{2,1} = 0$ , что даёт возможность разбить алгоритм на два независимых:

$$\begin{cases} C_{1,i}^{n+1} = C_{1,i-1}^n, & i = 1, \dots, m_1 \\ C_{1,1}^{n+1} = C_*(t) \end{cases}, \quad \begin{cases} C_{2,i}^{n+1} = C_{2,i-1}^n, & i = 1, \dots, m_2 \\ C_{2,1}^{n+1} = 0 \end{cases}.$$

После первого этапа значение  $C_{2,1}$  является неправильным. Оно заменяется значением из левой подобласти на втором этапе (этапе коррекции) параллельного алгоритма, который может быть реализован с применением процедур пересылки данных MPI, *таких как MPI\_SEND, MPI\_RECV* [6]:

$$C_{2,1} = C_{1,m_1}.$$

После коррекции последовательный и параллельный алгоритмы дают идентичные решения на каждом шаге по времени. Преимуществом такого подхода является то, что на первом этапе расчёта для каждой подобласти применяется тот же последовательный алгоритм, что и для исходной области.

Порядок численной аппроксимации по пространству зависит от числа узлов, участвующих в шаблоне схемы. Соответственно, число узлов, в которых значения должны быть откорректированы после первого этапа, зависит от порядка аппроксимации. Рассмотрим TVD-схемы адвекции [17, 18], имеющие формально 2-й порядок аппроксимации по пространству, для одномерного случая уравнения переноса (7), описываемые алгоритмом

$$C_i^{n+1} D_i^{n+1} = C_i^n D_i^n + U_L D_L \frac{\Delta t}{\Delta x} C_L - U_R D_R \frac{\Delta t}{\Delta x} C_R, \quad (8)$$

где  $U_L, U_R$  – скорость на левой и правой гранях численной ячейки,  $D_L, C_L, D_R, C_R$  – глубина и значения скаляра на левой и правой гранях соответственно,  $n$  – шаг по времени (индексы  $n$  и  $i$  для  $D$  и  $C$  опущены с целью сокращения выкладок). Различные схемы отличаются различным определением  $C_L, C_R$ . Например, в схеме центральных разностей

$$C_R = \frac{1}{2}(C_i^n + C_{i+1}^n) \quad [15]. \text{ TVD-схемы могут быть представлены, как в [18]:}$$

$$C_R = \begin{cases} C_R^+, & U_R \geq 0, \quad \text{где } C_R^+ = C_i^n + \frac{1}{2}\psi(r_i^+)(1-c)(C_{i+1}^n - C_i^n), \\ C_R^-, & U_R < 0, \quad \text{где } C_R^- = C_{i+1}^n - \frac{1}{2}\psi(r_i^-)(1+c)(C_{i+1}^n - C_i^n). \end{cases} \quad (9)$$

Здесь  $c$  – локальное число Куранта со знаком,  $r_i$  – отношение градиента скаляра вверх по потоку к локальному градиенту:

$$c = U_R \frac{\Delta t}{\Delta x}, \quad r_i^+ = \frac{C_i^n - C_{i-1}^n}{C_{i+1}^n - C_i^n}, \quad r_i^- = \frac{C_{i+2}^n - C_{i+1}^n}{C_{i+1}^n - C_i^n}. \quad (10)$$

Из (8)–(10) следует, что на значение  $C_i^{n+1}$  могут оказывать влияние  $C_{i-2}^n, C_{i-1}^n, C_i^n, C_{i+1}^n, C_{i+2}^n$ , в зависимости от направления потока и функции  $\psi$ , которыми TVD-схемы отличаются друг от друга [18].

На рис. 1 показаны разбишка исходной расчётной области, расширение (перекрытие) подобластей и процедура коррекции для схем 2-го порядка аппроксимации. Расширение подобластей в обе стороны необходимо из-за того, что поток может иметь различные направления. По этой же причине процедура коррекции является симметричной несмотря на то, что рассматриваемые TVD-схемы используют информацию только с одного узла вниз по потоку, и ассиметричная коррекция была бы более эффективной. В противном случае, если скорость потока близка к нулю, то численный шум, вызванный ошибками округления на двух различных процессорах, мог бы приводить к неустойчивости схемы обмена. Симметричный же обмен информацией постоянен на каждом шаге вне зависимости от направления скорости. Так как обсуждаемая модель использует разнесенные сетки [15], то для того, чтобы использовать одни и те же подпрограммы обмена для компонент скорости и для скаляра, удобным является метод перекрытия подобластей на один дополнительный слой. Как показал численный эксперимент, потеря производительности при этом незначительна.

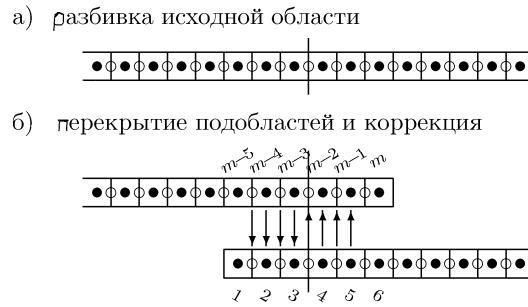


Рис. 1. Разбивка расчётной области и процедура коррекции

Рассмотрим теперь схему диффузии скаляра для (7), которая в 1-ом случае принимает вид

$$\frac{\partial}{\partial x} \left( A_h H \frac{\partial C}{\partial x} \right) \approx \frac{1}{\Delta x} \left( A_{h \ i+1/2}^n H_{i+1/2} \frac{C_{i+1}^n - C_i^n}{\Delta x} - A_{h \ i-1/2}^n H_{i-1/2} \frac{C_i^n - C_{i-1}^n}{\Delta x} \right). \quad (11)$$

Так как (11) использует всего один узел вверх по потоку и один вниз по потоку, то схема коррекции, представленная на рис. 1, является достаточной, чтобы заменить ошибочные значения, обусловленные фиктивными граничными условиями между подобластями на первом этапе. Расчёт  $A_h$  производится по формуле (6), которая включает вычисление производных в горизонтальном направлении. Поэтому применение процедуры коррекции к уравнению (7) более эффективно после того, как все адвективные и диффузионные слагаемые вычислены и сложены, чем по-отдельности.

Аппроксимация переноса импульса в (2), (3) может быть выполнена с использованием схемы центральных разностей [15] или TVD-схем [19], а вязкостных слагаемых (5) – аналогично диффузии скаляра. Уравнения (2), (3) также содержат слагаемые  $F_{BrCx}$ ,  $F_{BrCy}$ , вычисляемые согласно (4). Если порядок их аппроксимации не выше 2-го, то процедура коррекции может быть применена к (2) и (3) после того, как все упомянутые слагаемые вычислены и сложены.

Схема коррекции для двухмерных и трёхмерных уравнений аналогична. На рис. 2 показаны пример разделения расчётной области на 4 подобласти и их перекрытие. В трёхмерном случае рис. 2 является "видом сверху" на перекрытие трёхмерных подобластей; каждая показанная ячейка соответствует вертикальному столбу воды. Те ячейки 1-й подобласти, значения в которых должны быть заменены после первого этапа, расположены между границей подобласти после расширения (выделена чёрной жирной линией) и границей подобласти до её расширения (выделена белым). Эта часть 1-й подобласти перекрывается с "внутренними" частями остальных подобластей, где значения переменных не "испорчены" после первого этапа. Они используются для коррекции значений 1-й подобласти на втором этапе расчёта. Так как коррекция должна осуществляться для всех подобластей, то применение процедур пересылок  $MPI\_ISEND$  и  $MPI\_IRECV$  [6] более удобно, чем применение  $MPI\_SEND$  и  $MPI\_RECV$ , которое подразумевает строгую последовательность отправки и приёма данных, в то время как  $MPI\_ISEND$  и  $MPI\_IRECV$  могут посылать и принимать данные в произвольном порядке, не дожидаясь завершения пересылок. Ожидание завершения обмена данных и обработка возможных ошибок осуществляются позже, например, с помощью процедуры  $MPI\_WAITALL$ .

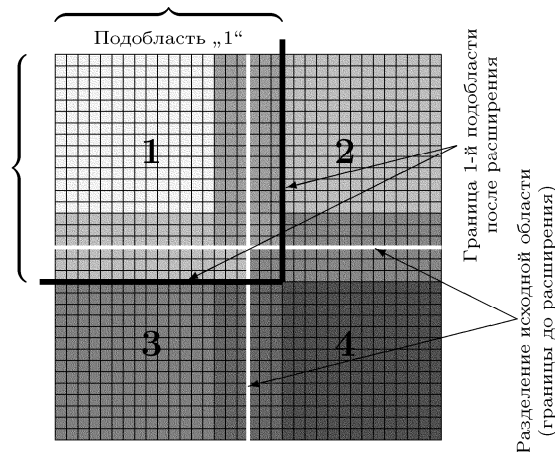


Рис. 2. Разбивка исходной двумерной (трёхмерной) области на 4 подобласти

Оптимизация разбивки в двумерном и трёхмерном случаях не является тривиальной задачей. Наличие «сухих» ячеек приводит к задаче о максимизации расчётных подобластей при минимизации длины их границ, которая может решаться, как в [8, 9]. В данной работе «приближение» оптимальной разбивки определялось таким образом, чтобы число ячеек подобласти было пропорциональным производительности соответствующего процессора, на котором производился расчёт этой подобласти. Однако учёт особенностей кластера может повлиять на оптимальную разбивку. Кроме того, процедуры вычисления источников, потоков импульса и тепла через поверхность [4] могут иметь различную длительность даже для подобластей, имеющих одинаковое число ячеек.

#### 4. Расчёт баротропных слагаемых и уровня

Далее рассмотрены следующие методы расчета баротропных слагаемых и уровня:

- 1) явный метод: расщепление расчета на внутреннюю (трехмерную) и внешнюю (двухмерную) моды с последующим их сопряжением [3]
- 2) полунявный метод: сведение системы уравнений к одной системе линейных уравнений для уровня поверхности  $\eta_{i,j}^{n+1}$ , как описано для 2-го случая в [13], и для 3-го – в [14].

В первом из этих методов предполагается, что  $\frac{\partial \eta}{\partial x} \approx \frac{\eta_{i,j}^n - \eta_{i-1,j}^n}{\Delta x}$ ,  $\frac{\partial \eta}{\partial y} \approx \frac{\eta_{i,j}^n - \eta_{i,j-1}^n}{\Delta y}$  –

явная аппроксимация 2-го порядка в разнесенных сетках для вычисления баротропных слагаемых. Увеличение производительности трёхмерной модели достигается тем, что ограничение на шаг интегрирования по времени, связанное с волновым числом Куранта, применяется только к двумерной моде. При этом алгоритм интегрирования двумерной моды является аналогичным двумерной модели гидродинамики с расчётом баротропных сил по явной схеме. Для параллелизации алгоритма расчёта каждый шаг интегрирования как трёхмерной, так и двумерной моды, состоит из двух этапов: на первом производится расчёт с использованием фиктивных граничных условий; на втором применяется процедура коррекции. Коррекция трёхмерной моды осуществляется после процедуры «склейки» мод.



Метод полунявного расчёта уровня и баротропных слагаемых является либо безусловно устойчивым, либо с более слабым ограничением на шаг интегрирования. Например, если представить  $\frac{\partial \eta}{\partial x} \approx \theta \frac{\eta_{i,j}^{n+1} - \eta_{i-1,j}^{n+1}}{\Delta x} + (1 - \theta) \frac{\eta_{i,j}^n - \eta_{i-1,j}^n}{\Delta x}$  с параметром  $\theta \in (0,1]$ . Однако он приводит к необходимости решения большой системы линейных уравнений с положительно определённой матрицей, для чего часто применяются разновидности методов сопряжённых градиентов и наискорейшего спуска [20]. Рассмотрим двумерную модель [13] с  $\theta = 1$ . Уравнение неразрывности (1) дискретизируется как

$$\frac{U_{i+1,j}^{n+1} D_{i+1/2,j}^{n+1} - U_{i,j}^{n+1} D_{i-1/2,j}^{n+1}}{\Delta x} + \frac{V_{i,j+1}^{n+1} D_{i,j+1/2}^{n+1} - U_{i,j}^{n+1} D_{i,j-1/2}^{n+1}}{\Delta y} + \frac{\eta_{i,j}^{n+1} - \eta_{i,j}^n}{\Delta t} = 0, \quad (12)$$

где  $D_{i-1/2,j}^{n+1} = \frac{D_{i-1,j}^{n+1} + D_{i,j}^{n+1}}{2}$  и т.д.

Дискретизация уравнения (2), осреднённого по глубине, имеет вид

$$\frac{U_{i,j}^{n+1} D_{i-1/2,j}^{n+1} - U_{i,j}^n D_{i-1/2,j}^n}{\Delta t} = -g D_{i-1/2,j}^n \frac{\eta_{i,j}^{n+1} - \eta_{i-1,j}^{n+1}}{\Delta x} - \lambda_{i-1/2,j}^n D_{i-1/2,j}^{n+1} U_{i,j}^{n+1} + \underbrace{\dots}_{\text{явные слагаемые}}.$$

Силы трения у дна (с коэффициентом  $\lambda_{i-1/2,j}^n$ , [13]) также дискретизируются полунявно. Остальные упущенные слагаемые рассчитываются по явным схемам. Аналогичное дискретное уравнение составляется для (3). Выражая из этих уравнений слагаемые вида  $U_{i,j}^{n+1} D_{i-1/2,j}^{n+1}$  как линейные функции неизвестных  $\eta_{i,j}^{n+1}$ ,  $\eta_{i-1,j}^{n+1}$ ,  $\eta_{i+1,j}^{n+1}$ ,  $\eta_{i,j-1}^{n+1}$ ,  $\eta_{i,j+1}^{n+1}$  и подставляя в (12), получаем уравнение для  $\eta$ :

$$a_{0,i,j}^n \eta_{i,j}^{n+1} - a_{1,i,j}^n \eta_{i-1,j}^{n+1} - a_{2,i,j}^n \eta_{i+1,j}^{n+1} - a_{3,i,j}^n \eta_{i,j-1}^{n+1} - a_{4,i,j}^n \eta_{i,j+1}^{n+1} = \eta_{i,j}^n + \underbrace{\dots}_{\text{явные слагаемые}}, \quad (13)$$

$$a_{0,i,j}^n = 1 + a_{1,i,j}^n + a_{2,i,j}^n + a_{3,i,j}^n + a_{4,i,j}^n, \quad a_{1,2,i,j}^n = g \frac{\Delta t^2}{\Delta x^2} \frac{D_{i\pm 1/2,j}^n}{1 + \lambda_{i\pm 1/2,j}^n}, \quad a_{3,4,i,j}^n = g \frac{\Delta t^2}{\Delta x^2} \frac{D_{i,j\pm 1/2}^n}{1 + \lambda_{i,j\pm 1/2}^n}. \quad (14)$$

Набор этих уравнений для каждой ячейки  $(i, j)$  вместе с граничными условиями образует систему линейных уравнений с 5-диагональной матрицей  $A$ . В трёхмерном случае вывод системы линейных уравнений для  $\eta$  подробно описан в [14]. При разбивке расчётной области на подобласти части матрицы  $A$  на первом этапе вычисляются независимо параллельными процессами согласно (14) и хранятся в памяти соответствующих вычислительных узлов как 5 подвекторов. Те элементы матрицы  $A$ , которые зависят от фиктивных граничных условий, являются ошибочными после первого этапа, если уровень и скорости не были откорректированы предварительно. В этом случае необходимо применение процедуры коррекции, подобной процедуре коррекции для уравнений (2), (3).

Рассмотрим построение параллельного алгоритма для решения системы линейных уравнений

$$A\vec{x} = \vec{b} \quad (15)$$

на основе метода наискорейшего спуска [20], предполагая, что части матрицы  $A$  и векторов  $\vec{x}$ ,  $\vec{b}$  вычисляются различными процессорами. Решение (15) находится в результате итераций

$$\vec{\omega}^n = A\vec{x}^n - \vec{b}, \quad \Delta_n = \frac{(\vec{\omega}^n, \vec{\omega}^n)}{(A\vec{\omega}^n, \vec{\omega}^n)}, \quad \vec{x}^{n+1} = \vec{x}^n - \Delta_n \vec{\omega}^n. \quad (16)$$

Начальное приближение решения  $\vec{x}^0$  может быть взято с предыдущего шага интегрирования. Алгоритм (16) требует вычисления двух скалярных произведений векторов и одного произведения матрицы на вектор на каждой итерации. Так как подвекторы хранятся в памяти различных вычислительных узлов, то операция вычисления скалярного произведения векторов  $\vec{x}$  и  $\vec{y}$  может быть представлена как сумма скалярных произведений подвекторов, вычисляемых независимо:

$$p = (\vec{x}, \vec{y}) = (\vec{x}_1, \vec{y}_1) + (\vec{x}_2, \vec{y}_2) + \dots + (\vec{x}_m, \vec{y}_m),$$

где  $m$  – число подобластей,  $\vec{x}_1$  и  $\vec{y}_1$  – соответствующие подвекторы. Вычисление полной суммы  $p$  в данной работе осуществлялось при помощи MPI функций *MPI\_REDUCE* и *MPI\_BROADCAST* [6]. Последняя необходима для передачи  $p$  всем процессам. При использовании сетевого коммуникатора 1 Гбит, соединяющего два узла с процессорами Intel P4 3ГГц, для посылки и приёма одного числа требуется ~130 мкс, что эквивалентно выполнению ~500 операций умножения таким же процессором, т.е. использование MPI неэффективно для векторов, состоящих менее чем из 1000 элементов.

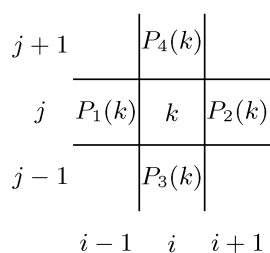


Рис. 3. Индексация ячеек сетки

Каждый ряд матрицы  $A$  содержит один положительный элемент на главной диагонали и 4 отрицательных элемента, расположенных на различных диагоналях в зависимости от расчётной области, в то время как остальные элементы равны нулю. При перемножении  $k$ -го ряда матрицы на вектор эти 5 элементов перемножаются с элементами вектора, соответствующих ячейкам  $(i, j)$ ,

$(i - 1, j)$ ,  $(i + 1, j)$ ,  $(i, j - 1)$ ,  $(i, j + 1)$ . Для операции умножения матрицы на вектор удобно ввести индексацию соответствующих элементов  $P_0(k)$ ,  $P_1(k)$ ,  $P_2(k)$ ,  $P_3(k)$ ,  $P_4(k)$ , как показано на рис. 3.

Тогда операция умножения  $\vec{x}^* = A\vec{x}$  поэлементно принимает вид

$$x_k^* = a_{0\ k} x_k + a_{1\ k} x_{P_1(k)} + a_{2\ k} x_{P_2(k)} + a_{3\ k} x_{P_3(k)} + a_{4\ k} x_{P_4(k)}, \quad (17)$$

где  $x_k^*$  и  $x_k$  – элементы векторов  $\vec{x}^*$  и  $\vec{x}$  соответственно. При этом  $P_0(k) = k$ ,  $P_1(P_2(k)) = k$ ,  $P_2(P_1(k)) = k$ ,  $P_3(P_4(k)) = k$ ,  $P_4(P_3(k)) = k$ , что может использоваться для проверки алгоритма.

Чтобы в результате параллельного и последовательного расчётов получить одинаковые решения, необходимо обеспечить идентичность векторов, полученных в результате параллельного и последовательного расчётов на каждом шаге итераций (16). Если элемент  $x_k$  соответствует ячейке сетки, прилегающей к поверхности разделения расчётной области, то на первом этапе каждой итерации  $x_k^*$  вычисляется согласно (17), используя фиктивные значения. Если обозначить  $\Omega$  – подмножество элементов  $(i, j)$  в схеме коррекции для адвекции скаляра 1-го порядка и  $k = M(i, j)$  – однозначную трансформацию пары  $(i, j)$  в векторный индекс  $k$ , то подмножество элементов  $k = M(\Omega(i, j))$  вектора  $\vec{x}^*$  должно быть замещено на этапе коррекции. Поскольку  $\Omega$  является подмножеством элементов, которые корректируются при использовании схем переноса 2-го и более высокого порядков, те же самые процедуры коррекции могут быть использованы, как и для скаляра. Как показал численный эксперимент, потеря производительности за счёт "бесполезных" умножений незначительна, если подобласти достаточно велики.

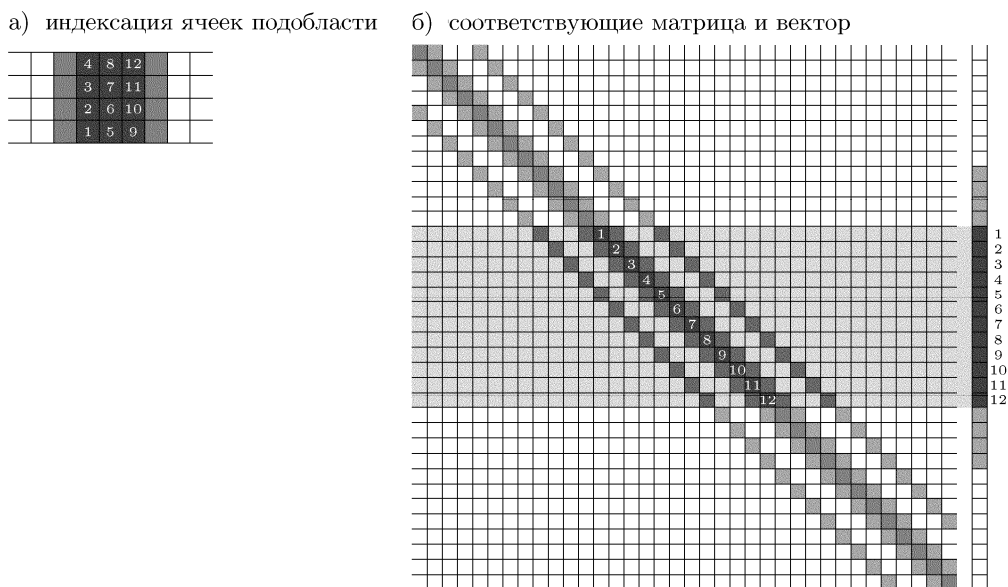


Рис. 4. Умножение матрицы на вектор для прямоугольной подобласти

Рис. 4 поясняет операцию умножения матрицы на вектор для прямоугольной расчётной области. Выделенная подобласть состоит из двенадцати элементов, четыре из которых (1, 2, 3, 4) находятся у левой границы и четыре (9, 10, 11, 12) – у правой. После первого этапа умножения части матрицы (затемнена) на часть вектора, которые соответствуют рассматриваемой подобласти, элементы  $(x_1^*, x_2^*, x_3^*, x_4^*)$  и  $(x_9^*, x_{10}^*, x_{11}^*, x_{12}^*)$  являются ошибочными. На втором этапе эти элементы заменяются элементами из внутренних частей соответствующих соседних подобластей.

Синхронизация параллельных процессов и обмен данными, которые осуществляются на каждой итерации (16), также приводят к потере производительности. Число итераций, зависящее от скорости сходимости, желаемой точности и начального приближения, может быть существенно уменьшено применением предобуславливателя – такой несингулярной матрицы  $C$ , которая преобразует исходную систему уравнений (15) в эквивалентную систему с улучшенной сходимостью:

$$A^* \vec{x}^* = \vec{b}^*, \quad (18)$$

где  $A^* = C^{-1}AC^{-1}$ ,  $\vec{x}^* = C\vec{x}$ ,  $\vec{b}^* = C^{-1}\vec{b}$ . Такое преобразование требует вычисления  $\vec{b}^*$  и  $A^*$ , что накладывает определённые ограничения на свойства  $C$ : решение уравнения  $C\vec{b}^* = \vec{b}$  не должно быть трудоёмким; матрица  $A^*$  не должна содержать большого числа ненулевых элементов; вычисления должны предусматривать параллелизацию. Простейший диагональный предобуславливатель [21], элементы которого рассчитываются как

$$c_{ij} = \begin{cases} \sqrt{a_{ij}}, & i = j, \\ 0, & i \neq j, \end{cases} \quad (19)$$

удовлетворяет этим требованиям. При этом  $b_i^* = \frac{1}{\sqrt{a_{ii}}}b_i$ ,  $x_i = \frac{1}{\sqrt{a_{ii}}}x_i^*$  могут вычисляться

независимыми параллельными процессами. Принимая во внимание то, что каждый ряд  $k$  матрицы  $A$  имеет 5 ненулевых элементов, расположенных в колонках  $P_0(k) = k$ ,  $P_1(k)$ ,  $P_2(k)$ ,  $P_3(k)$ ,  $P_4(k)$ , легко видеть, что  $A^*$  имеет ненулевые элементы на тех же самых местах, что и матрица  $A$ :

$$a_{P_i(k)k}^* = \frac{a_{P_i(k)k}}{c_{kk}c_{P_i(k)P_i(k)}}, \quad i = 0,1,2,3,4 \quad (\Rightarrow a_{kk}^* \equiv 1). \quad (20)$$

Если ряд  $k$  матрицы  $A^*$  соответствует элементу сетки, прилегающему к поверхности разделения подобластей, то те элементы этого ряда, которые зависят от фиктивных  $c_{P_i(k)P_i(k)}$ , должны быть заменены путём применения тех же самых процедур коррекции, как при вычислении матрицы  $A$ . Диагональный предобуславливатель имеет следующие преимущества: матрица  $A^*$  имеет ту же структуру, что и  $A$ ; могут применяться те же процедуры умножения матрицы на вектор, как и для  $A$ ;  $C$  не требует много памяти; процедура коррекции для уравнения переноса скаляра может быть использована для коррекции  $C$ ,  $\vec{b}^*$ ,  $A^*$ ; число необходимых умножений уменьшается на 20%, так как  $a_{kk}^* \equiv 1$ ; возможно применение SIMD (Single Instruction Multiple Data) инструкций процессора [2].

## 5. Проверка метода

Метод коррекции проверялся путём сравнения результатов модели без разбивки и с разбивкой на подобласти. В качестве примера далее показано образование вихревой дорожки Кармана позади препятствия в потоке: во-первых, относительно сложная структура потока реализуется при простой геометрии твёрдых границ, в результате чего поток может пересекать границы подобластей в различных направлениях; во-вторых, поток является неустойчивым, что приводит к существенным отличиям в гидродинамических полях даже при малых возмущениях на стыках подобластей.

Течение в канале постоянной глубины 10м с препятствием, расположенным в центре канала перпендикулярно его оси, рассчитывалось при помощи двумерной модели. Канал был покрыт сеткой 520x33 ячеек с разрешением 10м. С целью уменьшения диссипации энергии коэффициент горизонтальной вязкости рассчитывался согласно (6) с  $\alpha = 0,02$ , а шероховатость дна полагалась  $z_0 = 3 \times 10^{-5}$  м. На двух открытых границах задавались уровни  $\pm 12,5$  см. Начальные условия полагались нулевыми как для скорости, так и для уровня. С целью избежания образования "ударной" волны уровень плавно изменялся на границах в течение 0,5 дня после начала расчёта, после чего оставался постоянным. Параллельный расчёт проводился с разбивкой на 4 подобласти, которые имели общую область перекрытия позади препятствия – зона, в которой модель наиболее чувствительна к всевозможным возмущениям.

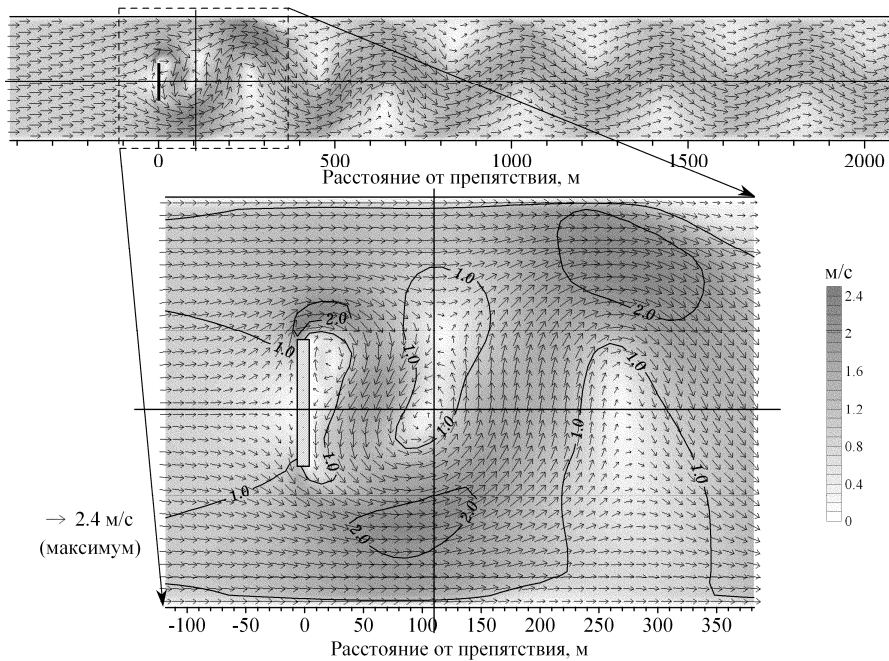


Рис. 5. Образование вихревой дорожки Кармана позади препятствия. На нижнем рисунке показан вектор скорости для каждой ячейки численной сетки

На рис. 5 показано поле скорости на момент времени  $t = 1,5$  дня после начала расчётов после «склейки» 4-х подобластей. На верхнем рисунке показана общая картина течения, на нижнем – поле скорости вблизи препятствия с вектором скорости для каждой ячейки сетки. Как показало сравнение, при вычислениях с двойной точностью результат является идентичным полученному без разделения на подобласти в терминах значений переменных в соответствующих ячейках численных сеток для исходной области и подобластей.

## 6. Производительность

Физическое время  $T_{real}$ , затраченное на выполнение расчёта, зависит от длительности моделируемого интервала времени  $T_{sim}$ , от шага интегрирования по времени  $\Delta t$  (который может изменяться на протяжении расчётов и зависит от конкретных численных схем) и от числа операций, производимых процессором за единицу физического времени. Поэтому в рамках данной работы под термином производительности  $P$  будем подразумевать отношение числа шагов интегрирования  $n_T$  (если  $\Delta t = const$ , то  $n_T = T_{sim}/\Delta t$ ), необходимого для интегрирования системы уравнений от  $t = 0$  до  $t = T_{sim}$ , к физическому времени  $T_{real}$ , затраченному на расчёт:

$$P = n_T / T_{real} .$$

Для изучения производительности параллельных вычислений с применением метода коррекции для двумерной модели было проведено 4 тестовых расчёта с численными сетками различных размеров: 520x50, 1040x50, 520x100, 1040x100 узлов, покрывающих площади 52 км x 5 км, 104 км x 5 км, 52 км x 10 км, 104 км x 10 км соответственно. Батиметрическая глубина во всех случаях была 10м. На левой открытой границе задавалось синусоидальное отклонение уровня поверхности от невозмущённого с амплитудой 2м и периодом 0,5 дня. Для расчёта уровня поверхности использовалась полуняевная схема. Тест проводился на кластере, состоящем из двухпроцессорных узлов AMD Opteron 248 2,2 ГГц. Было проанализировано два способа разбивки исходной расчётной области на конгруэнтные подобласти, обозначенные как "А" и "Б" на рис. 6.



Рис. 6. Способы разбивки на конгруэнтные подобласти в проведенных экспериментах

В случае разбивки "Б" две соседних подобласти с более длинной общей границей рассчитывались на одном и том же узле кластера с общей памятью. Тесты проводились под управлением операционной системы "Windows XP". Дополнительно были проанализированы два способа выполнения процессов системой: с привязкой процесса к фиксированному процессору ("Ф") и с возможностью произвольного выбора процессора операционной системой ("П").

Таблица 1. Увеличение производительности, раз

Конфиг.	520x50		1040x50		520x100		1040x100	
	Ф	П	Ф	П	Ф	П	Ф	П
1(1)А	1	0,93	1	0,94	1	0,93	1	0,95
2(1)А	2,16	1,46	1,92	1,42	2,02	1,47	2,02	1,54
2(1)Б	1,85	1,19	1,73	1,3	1,92	1,37	1,95	1,54
2(2)А	1,95	1,95	1,87	1,82	1,91	1,87	1,96	2,03
2(2)Б	1,62	1,59	1,62	1,58	1,78	1,73	1,89	1,91
3(2)А	2,86	2,06	2,86	2,38	2,86	2,32	3,11	2,63
3(3)А	2,86	2,75	2,85	2,83	2,84	2,78	3,13	3,12
4(2)А	3,48	2,4	3,86	2,71	3,87	2,79	4,13	3,29
4(2)Б	3,29	1,98	3,38	2,4	3,7	2,58	3,93	3,13
4(4)А	2,83	3,19	3,18	3,77	3,43	3,78	3,76	3,95
5(3)А	4,21	2,86	4,8	3,51	4,84	3,24	5,11	3,88
6(3)А	4,59	3,03	5,47	3,66	5,31	3,56	6,14	4,44
6(3)Б	3,99	2,48	4,84	3,11	5,31	3,54	5,68	4,25
7(4)А	4,97	3,29	<b>6,23</b>	3,83	<b>6,21</b>	3,83	<b>6,93</b>	5,01
8(4)А	<b>5,05</b>	3,44	6,17	4,03	5,98	3,87	6,61	5,42
8(4)Б	4,46	2,89	5,47	3,49	5,22	3,62	5,68	4,92

В табл.1 показана производительность, отнесенная к производительности последовательного расчёта. Первая колонка описывает конфигурацию разбиения и эксперимента: число задействованных процессоров; число задействованных узлов кластера (в скобках); "А" и "Б" – способы разбивки. Во всех случаях наилучшая производительность (выделена жирным шрифтом) достигалась, если каждый вычислительный процесс был привязан к определённому процессору, при условии, что были задействованы оба процессора одного и того же узла. Если был задействован только один из двух процессоров узла, то производительность могла быть выше без привязки (как в случае "4(4)А"). По-видимому, это связано со случайным выбором операционной системой процессора для выполнения задачи: излишнее копирование информации из кэш-памяти одного процессора в кэш-память другого приводит к значительной потере производительности. Необходимо также отметить важность использования предобуславливателя в моделях с неявным расчётом уровня. Без предобуславливателя среднее число итераций на каждом шаге интегрирования, необходимых для решения (15) с абсолютной погрешностью  $10^{-6}$ м, было приблизительно 13,2, 12,1, 13,1, 12,0 для каждой из упомянутых задач соответственно, в то время как с предобуславливателем требовалось в среднем 7,3, 6,2, 7,7 и 6,5 итераций для решения (18) с эквивалентной точностью.

По сравнению с двумерной моделью, производительность трёхмерной модели зависит от многих других факторов, таких как число вертикальных слоёв; применение неявных схем по вертикали; вычисление трёхмерных источников, включающих трудоёмкие операции, как, например, в  $k/\varepsilon$  модели турбулентности [16]. Так как разбивка на подобласти осуществляется в горизонтальных направлениях, то перечисленные факторы приводят только к увеличению части времени, затраченного на выполнение независимых операций и, как следствие, к увеличению эффективности параллелизации.

Производительность трёхмерной модели изучалась на примере моделирования течения в канале прямоугольной формы, покрытого сеткой 520x50x30 ячеек. Размеры канала и граничные условия были такими же, как в первом из описанных экспериментов для двумерной модели. Гидродинамика описывалась уравнениями (1)–(6), дополненными  $k/\varepsilon$  моделью для вычисления коэффициентов вертикальной вязкости. Были протестированы два метода расчёта уровня поверхности: полунеявный и с расщеплением на моды.

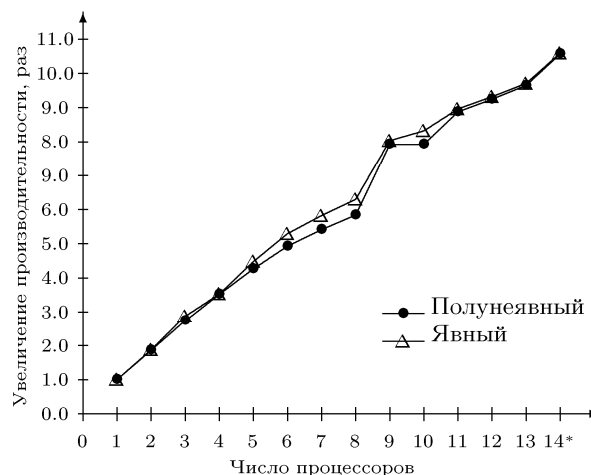


Рис. 7. Производительность трёхмерной модели

На рис. 7 показано увеличение производительности в зависимости от числа процессоров. Обе модели (с полунявным и явным расчётом уровня) показали схожую производительность: помимо того, что в трёхмерной модели большая часть времени тратится на расчёт вертикального распределения, синхронизация процессов и обмен данными на каждом шаге итераций при решении системы уравнений являются подобным синхронизации процессов и обмену данными на каждом шаге интегрирования двухмерной моды. Способ разбивки "Б" на 14 подобластей оказался предпочтительнее вследствие уменьшения площади подобластей при неизменной длине границ между подобластями разбивки "А". Наихудшая эффективность, определённая как  $E = P_N / (P_1 N) \cdot 100\%$ , где  $P_1$  – производительность последовательной модели,  $P_N$  – производительность с использованием  $N$  процессоров, в тестовой задаче была ~69% в случае разбивки "А" и ~75% в случае разбивки "Б" на 14 подобластей. Тем не менее производительность при этом была наибольшей. Некоторое увеличение эффективности, когда было задействовано 9 процессоров по сравнению с 8, по-видимому, обусловлено более эффективным использованием кэш-памяти.

Метод коррекции с применением MPI также может быть использован для расчётов на двухъядерных процессорах. Вышеописанный эксперимент трёхмерного моделирования течения в канале показал эффективность ~85% (явный метод) и ~88% (полунявный метод) при использовании процессора Intel Dual Core 2,4 ГГц и разбивке всей расчётной области на две подобласти.

## 7. Выводы

Предложен метод параллелизации алгоритмов расчёта, используемых в численных гидростатических моделях водоёмов, основанный на разбивке расчётной области на подобласти. Преимуществом его является то, что на первом этапе каждого шага интегрирования или итерации алгоритм расчёта для каждой подобласти не отличается от последовательного. Коррекция результатов производится на втором этапе с помощью дополнительных процедур с применением MPI. Полученное решение является точно таким же, как при последовательном расчёте, а эффективность параллелизации вполне приемлема при расчётах на кластерах, состоящих из ~10-20 процессоров, а также на узлах с общей памятью и многоядерных процессорах.

Тестовые задачи показали, что эффективность параллелизации в большей степени зависела от индивидуальной производительности процессоров и размеров подобластей, чем от пропускной способности сетевых коммутаторов. В трёхмерной модели рост производительности с увеличением числа процессоров оказался схожим при использовании явного и полунявного методов расчёта уровня поверхности и наблюдался во всех проведенных экспериментах с использованием до 14-ти процессоров. Важными факторами увеличения производительности и эффективности являлось использование диагонального предобуславливателя для моделей с полунявным методом расчёта уровня, в особенности двухмерных, а также привязка каждого процесса к определённому процессору при расчётах на многопроцессорных системах.

Дальнейшая работа включает оптимизацию разбивки расчётной области; параллелизацию негидростатических моделей, приводящих к необходимости решения системы линейных уравнений



для давления (или его составляющей) с более чем пятидиагональной матрицей; исследование эффективности предложенного метода параллелизации при большем числе процессоров.

Автор выражает искреннюю благодарность доктору физико-математических наук, профессору В.С. Мадеричу за замечания и советы. Автор признателен TMSI, National University of Singapore за оказанную поддержку в работе.

## СПИСОК ЛИТЕРАТУРЫ

1. Педлоски Дж. Геофизическая гидродинамика. – Москва: Мир, 1984. – Т. 1. – 398 с.
2. Intel. Intel 64 and IA-32 Architectures Optimization Manual №248966-015. – Intel Corporation: Denver, 2007. – 488 p.
3. Blumberg A.F., Mellor G.L. A description of a three-dimensional coastal ocean circulation // Three-Dimensional Coastal Ocean Models / N. Heaps (ed). – Washington, D.C.: Am. Geoph. Union, 1987. – P. 1 – 16.
4. Моделирование распространения тепла во внутренних водах и прибрежных областях морей / В. Кошебуцкий, В. Мадерич, А. Нестеров и др. // Прикладная гидромеханика. – 2004. – № 6 (78). – С. 34 – 44.
5. Ashworth M., Davies A.M. Performance of a three dimensional hydrodynamic model on a range of parallel computers // Proc. of Euromicro Workshop on Parallel and Distributed Processing. – Gran Canaria, Spain, 1993. – P. 383 – 390.
6. Gropp W. et al. Portable Parallel Programming with the Message Passing Interface, 2<sup>nd</sup> Ed. / W. Gropp, E. Lusk, A. Skjellum. – Cambridge, Mass.: MIT Press, 1999. – 395 p.
7. Parallelization of the GESIMA mesoscale atmospheric model / M. Ashworth, A. Foelkel, V. Gulzov et al. // Parallel Computing. – 1997. – N 23. – P. 2201 – 2213.
8. Towards modeling flow and water structure in the Rockall Slope / A.J. Souza, S. Walkelin, J.T. Holt et al. // Proc. Of the PECS'02. – Hamburg, 2002. – P. 409 – 412.
9. Ashworth M. Optimization of the POLCOMS Hydrodynamic Code for Terascale High Performance Computers / M. Ashworth, J.T. Holt, R. Proctor // Proc. of 18<sup>th</sup> Int. Parallel & Distributed Proc. Symp. – New Mexico, 2004. – P. 383 – 390.
10. Kerbyson D.J., Jones P.W. A performance model for Parallel Ocean Program // Int. J. of High Performance Computing Applications. – 2005. – Vol.19. – P. 261 – 276.
11. A domain decomposition method for three-dimensional shallow water equations / E.D. Goede, J. Groeneweg, K.H. Tan et al. // Simulation Practice and Theory. – 1995. – N 3. – P. 307 – 325.
12. The parallelization of the Princeton Ocean Model / L.A. Boukas, N.Th. Mimikou, N.M. Missirlis et al. // Springer, Lect. Notes In Comp. Sci.; Proc. Of Euro-Par'99. – 1999. – P. 1395 – 1402.
13. Casulli V. Semi-Implicit Finite Difference Methods for the Two-Dimensional Shallow Water Equations // J. of Computational Physics. – 1990. – N 86. – P. 56 – 74.
14. Casulli V., Cheng R.T. Semi-Implicit Finite Difference Methods for Three-Dimensional Shallow Water Flow // Int. J. for Numer. Meth. In Fluids. – 1992. – N 15. – P. 629 – 648.
15. Флетчер К. Вычислительные методы в динамике жидкостей. – Москва: Мир, 1991. – Т. 1: Основные положения и общие методы. – 502 с.
16. Burchard H., Peterson O. Models of turbulence in the marine environment – a comparative study of two-equation turbulence models // Journal of Marine Systems. – 1999. – N 21. – P. 29 – 53.
17. Tannehill J.C. et al. Computational Fluid Mechanics and Heat Transfer, 2<sup>nd</sup> Ed. / J.C. Tannehill, D.A. Anderson, R.H. Pletcher. – Washington: Taylor and Francis, 1997. – 792 p.
18. Fringer O-B. et al. Reducing numerical diffusion in interfacial gravity wave simulations / O-B. Fringer, S.W. Armfield, R.L. Street // Int. J. Numer. Meth. Fluids. – 2005. – N 49. – P. 301 – 329.
19. Stelling G.S., Duijnmeijer S.P.A. A staggered conservative scheme for every Froude Number in rapidly varied shallow water flows // Int. J. for Num. Meth. In Fluids. – 2003. – N 43. – P. 1329 – 1354.
20. Бахвалов Н.С. Численные методы. Анализ, алгебра и обыкновенные дифференциальные уравнения. – Москва: Наука, 1975. – 631 с.
21. Templates for the solution of linear systems: building blocks for iterative methods / R. Barrett, M. Berry, T.F. Chan et al. – 2nd Ed. – Phil.: SIAM, 1994. – 112 p.

*Стаття надійшла до редакції 04.02.2008*