

УДК 519.63; 519.683; 519.688

doi 10.26089/NumMet.v20r108

МЕТОД БАЛАНСИРОВКИ НАГРУЗКИ ВЫЧИСЛЕНИЙ С ИСПОЛЬЗОВАНИЕМ КРИВЫХ ГИЛЬБЕРТА ПРИМЕНИТЕЛЬНО К ПАРАЛЛЕЛЬНОМУ АЛГОРИТМУ РЕШЕНИЯ УРАВНЕНИЙ МЕЛКОЙ ВОДЫ

А. В. Чаплыгин¹, Н. А. Дианский², А. В. Гусев³

Представлен метод балансировки нагрузки вычислений с использованием кривых Гильберта применительно к параллельному алгоритму решения уравнений мелкой воды. Рассматриваемая система уравнений мелкой воды возникает в сигма-модели общей циркуляции океана INMOM (Institute of Numerical Mathematics Ocean Model) при разрешении гравитационных волн и является одним из основных блоков модели. Из-за наличия в океанах островов и берегов балансировка нагрузки вычислений на процессоры является особенно актуальной задачей. В качестве одного из таких методов был выбран метод балансировки нагрузки вычислений с использованием кривых Гильберта. Продемонстрирована большая эффективность этого метода по сравнению с равномерным разбиением без балансировки нагрузки и показано, что этот метод служит хорошей альтернативой библиотеке разбиений METIS. Оптимальность реализованного разбиения для мелкой воды точно соответствует оптимальности и для трехмерной сигма-модели INMOM в силу одинакового количества вертикальных уровней во всей расчетной области.

Ключевые слова: уравнения мелкой воды, метод декомпозиции области, параллельные вычисления, метод балансировки нагрузки вычислений, кривые Гильберта.

1. Введение. Рассматривается система нелинейных уравнений мелкой воды, являющаяся блоком сигма-модели общей циркуляции океана INMOM (Institute of Numerical Mathematics Ocean Model), развиваемой в ИВМ РАН [1, 2]. Модель относится к классу сигма-моделей океана: в ней в качестве вертикальной переменной используется безразмерная величина $\sigma \in [0, 1]$, которая определяется из соотношения:

$$\sigma = \frac{z + \zeta}{H + \zeta},$$

где z — направленная вниз обычная вертикальная координата по глубине с началом на невозмущенной поверхности океана; ζ — отклонение уровня океана от невозмущенной поверхности; H — глубина океана в состоянии покоя. Модель написана на языке Fortran 90/95. Предыдущая версия модели INMOM используется в качестве океанического блока климатической модели INMCM (Institute of Numerical Mathematics Climate Model), созданной в ИВМ РАН и участвующей в программе IPCC (Intergovernmental Panel on Climate Change) по прогнозированию изменений климата [22].

Система уравнений мелкой воды является неотъемлемой и одной из самых важных подзадач в моделях общей циркуляции океана [24, 25]. Решение этой системы уравнений занимает существенную часть времени, необходимого для решения полной задачи циркуляции океана [2]. Кроме того, на основе нелинейных уравнений мелкой воды реализуются наиболее продвинутые системы предупреждения о цунами [4], реализованные для Мирового океана с высоким пространственным разрешением. Поэтому возникает вопрос об эффективной параллельной реализации алгоритма решения системы уравнений мелкой воды.

В работах [2, 11] реализован параллельный алгоритм решения уравнений нелинейных уравнений мелкой воды и протестирован для задач моделирования цунами в Тихом океане и шторма 2013 г. на Азовском море [21, 23]. Использовался алгоритм, основанный на методе декомпозиции области, который разбивает расчетную область на подобласти и ставит каждому процессору в соответствие ту подобласть, на которой процессор проводит свои вычисления. В работах [2, 11] использовалось равномерное разбиение

¹ Московский государственный университет им. М. В. Ломоносова, факультет вычислительной математики и кибернетики, Ленинские горы, 119992, Москва; студент, e-mail: achaplygin99@gmail.com

² Московский государственный университет имени М. В. Ломоносова, физический факультет, Ленинские горы, 119991, Москва; главный науч. сотр., e-mail: nikolay.diansky@gmail.com

³ Институт вычислительной математики им. Г. И. Марчука РАН (ИВМ РАН), ул. Губкина, 8, 119333, Москва; ст. науч. сотр., e-mail: anatoly.v.gusev@gmail.com

на подобласти. Однако из-за наличия берегов и островов такое разбиение не обеспечивает равномерную нагрузку на процессоры, что приводит к снижению эффективности параллельной программы. В этой связи возникает задача о разбиении, которое обеспечивало бы равномерную или близкую к равномерной нагрузку вычислений на процессоры.

Под балансировкой нагрузки вычислений имеется в виду такое распределение подзадач между процессорами, которое обеспечивает наиболее равномерно распределенную нагрузку на процессоры и минимальные затраты на передачу данных между ними. Методы балансировки нагрузки вычислений позволяют снизить время простоя отдельных процессоров и тем самым повысить эффективность параллельной программы, что существенно важно при решении задач на высокопроизводительных вычислительных системах. Балансировка нагрузки вычислений особенно актуальна в задачах моделирования циркуляции океана и моделирования цунами, потому что из-за наличия берегов и островов в этих задачах равномерное разбиение будет давать особенно несбалансированные подобласти, как будет показано далее в настоящей статье. Важно отметить, что оптимальность разбиения для мелкой воды будет точно соответствовать оптимальности для трехмерной сигма-модели INMOM, так как количество расчетных уровней по глубине в сигма-модели одинаково для всех точек сетки по горизонтали. Отметим также, что для z -моделей в общем случае это не так.

Существуют два основных способа балансировки нагрузки: методы, основанные на графах, и геометрические методы. Методы, основанные на графах, реализованы, например, в библиотеках METIS и parMETIS [19] и пользуются довольно большой популярностью. Такие методы представляют расчетную область в виде графа, в котором вершины соответствуют узлам сетки, а ребра — связям между узлами. Однако у них есть недостаток — это их сложность для программной реализации. Поэтому, как альтернатива этим методам, в задачах моделирования океана часто используются геометрические методы, такие как разбиение вдоль фрактальной кривой [13], рекурсивная бисекция [17] и др. Эти методы основаны на том, что каждый узел имеет связи со своими соседями. Геометрические методы дают разбиение хорошего качества, и это хорошая альтернатива METIS/parMETIS для многих задач.

В настоящей статье рассматривается один из геометрических методов балансировки нагрузки: метод разбиения вдоль фрактальной кривой, который уже успел зарекомендовать себя во многих работах, например в [12, 13]. В нашей работе показано, что данный метод дает хорошие результаты применительно к решению уравнений мелкой воды, а также выполнено сравнение этого метода с METIS.

2. Нелинейные уравнения мелкой воды. В произвольной ортогональной системе координат (x, y) с метрическими коэффициентами Ламе (r_x, r_y) полная система нелинейных уравнений мелкой воды записывается в виде [3]

$$\begin{aligned} \frac{\partial r_x r_y h u}{\partial t} + T_u(u, v) - F_u(u, v) - h r_x r_y l v + r_y h g \frac{\partial \zeta}{\partial x} + r_x r_y \frac{g n^2}{h^{1/3}} u \sqrt{u^2 + v^2} &= R H S_u, \\ \frac{\partial r_x r_y h v}{\partial t} + T_v(u, v) - F_v(u, v) + h r_x r_y l u + r_x h g \frac{\partial \zeta}{\partial y} + r_x r_y \frac{g n^2}{h^{1/3}} v \sqrt{u^2 + v^2} &= R H S_v, \\ \frac{\partial h}{\partial t} + \frac{1}{r_x r_y} \left(\frac{\partial u r_y h}{\partial x} + \frac{\partial v r_x h}{\partial y} \right) &= 0, \end{aligned} \quad (1)$$

где u, v — компоненты осредненного по глубине вектора горизонтальной скорости; r_x, r_y — метрические коэффициенты; l — параметр Кориолиса; g — ускорение свободного падения; n — коэффициент донного трения, оператор которого выписан в форме, предложенной в японской модели TUNAMI [4]; ζ — отклонение уровня моря относительно невозмущенного состояния; $h = H + \zeta$ — полная глубина океана; H — глубина океана в состоянии покоя. Операторы переноса T_u, T_v записываются в криволинейной системе координат в дивергентной форме:

$$\begin{aligned} T_u(u, v, h) &= \frac{\partial h r_y u u}{\partial x} + \frac{\partial h r_x v u}{\partial y} - h \left(v \frac{\partial r_y}{\partial x} - u \frac{\partial r_x}{\partial y} \right) v, \\ T_v(u, v, h) &= \frac{\partial h r_y u v}{\partial x} + \frac{\partial h r_x v v}{\partial y} + h \left(v \frac{\partial r_y}{\partial x} - u \frac{\partial r_x}{\partial y} \right) u. \end{aligned} \quad (2)$$

С вычислительной точки зрения дивергентная форма обладает полезными свойствами:

- она сохраняет интеграл переносимой величины по замкнутой области при условии непротекания на твердых границах и выполнения уравнения неразрывности;
- она допускает простую конечноразностную аппроксимацию.

Преимущества дивергентной формы записи уравнений изложены в [5].

Операторы вязкости F_u, F_v записываются как дивергенция тензора напряжений:

$$\begin{aligned} F_u(u, v) &= \frac{1}{r_y} \frac{\partial}{\partial x} (r_y^2 K D_T h) + \frac{1}{r_x} \frac{\partial}{\partial y} (r_x^2 K D_S h), \\ F_v(u, v) &= -\frac{1}{r_x} \frac{\partial}{\partial y} (r_x^2 K D_T h) + \frac{1}{r_y} \frac{\partial}{\partial x} (r_y^2 K D_S h). \end{aligned} \tag{3}$$

Здесь K — коэффициент вязкости, а D_T и D_S — компоненты тензоров напряжений сжатия–растяжения и сдвига соответственно:

$$D_T = \frac{r_y}{r_x} \frac{\partial}{\partial x} \left(\frac{u}{r_y} \right) - \frac{r_x}{r_y} \frac{\partial}{\partial y} \left(\frac{v}{r_x} \right), \quad D_S = \frac{r_x}{r_y} \frac{\partial}{\partial y} \left(\frac{u}{r_x} \right) + \frac{r_y}{r_x} \frac{\partial}{\partial x} \left(\frac{v}{r_y} \right). \tag{4}$$

Использование вязкости в форме (3) и (4) позволяет сохранять свойство бездиссипативности при твердотельном вращении жидкости.

В общем случае в правых частях RHS_u, RHS_v рассчитываются градиенты атмосферного давления и напряжения трения ветра:

$$RHS_u = P_x + \tau_x^{\text{surf}}, \quad RHS_v = P_y + \tau_y^{\text{surf}}. \tag{5}$$

Здесь P_x, P_y — компоненты градиента атмосферного давления на поверхности океана, $\tau_x^{\text{surf}}, \tau_y^{\text{surf}}$ — напряжения трения ветра на морской поверхности. В правых частях также могут рассчитываться и приливные силы, рассчитываемые через приливной потенциал. На берегах для скорости задаются граничные условия непротекания и свободного скольжения.

Именно в виде (1)–(5) нелинейные уравнения мелкой воды представлены в новой версии сигма-модели общей циркуляции океана INMOM [2]. Эта система уравнений возникает в модели при разрешении быстрых баротропных гравитационных волн.

Важно отметить, что все переменные в данной постановке задачи двумерные. В противном случае правые части должны содержать интегралы по глубине от нелинейного взаимодействия трехмерных величин, в частности адвективных слагаемых.

3. Описание численной реализации.

3.1. Дискретизация по пространству. При дискретизации по пространству нелинейных уравнений мелкой воды используется сетка, в общем случае нерегулярная по долготе и широте. Разобьем область $\{x \in [x_0, x_{\max}], y \in [y_0, y_{\max}]\}$, содержащую область, на которой рассматривается система уравнений (1), на элементарные ячейки, имеющие форму прямоугольников: $\{(x, y) : x_{m-1} < x < x_m, y_{n-1} < y < y_n\}$.

Для решения системы уравнений (1) применяется техника построения разностных аппроксимаций по пространству второго порядка точности на разнесенной C-сетке по классификации Аракавы [6, 7]. На рис. 1 показаны распределения переменных в каждой сеточной ячейке. В центре ячейки задаются ζ и H . На серединах боковых сторон ячейки задаются компоненты вектора скорости (u, v) и потоковые переменные. Параметр Кориолиса l определяется в угловых точках.

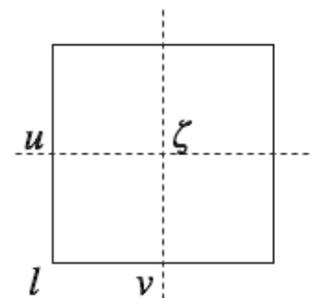


Рис. 1. Распределение переменных на ячейке модельной сетки

При построении разностных схем особое место уделяется тому, чтобы в разностных аналогах дифференциальных операторов сохранялись свойства симметрии, которые выполняются для исходной дифференциальной задачи. Это позволяет в разностной задаче автоматически удовлетворять энергетическим соотношениям, справедливым для дифференциальной. Методика построения пространственных разностных аппроксимаций изложена в [5, 8].

3.2. Схема по времени. При дискретизации нелинейных уравнений мелкой воды в качестве схемы по времени используется схема “чехарда со средней точкой” (leapfrog). Для определения решения на шаге $n + 1$ используются решения на шагах n и $n - 1$.

Рассмотрим простейшее уравнение адвекции: $\frac{dU}{dt} = F(U)$. Применяя численную схему по времени “чехарда со средней точкой”, получим разностную схему

$$\frac{U^{n+1} - U^{n-1}}{2\tau} = F(U^n).$$

У такой схемы по времени есть основной недостаток: расщепления решения по нечетным и четным временным шагам [5]. Поэтому на каждом временном шаге n делается фильтрация [9]:

$$U^s = U^n + \frac{a}{2} (U^{n+1} - 2U^n + U^{n-1}).$$

Затем при переходе на следующий шаг по времени значение U^s присваивается $(n - 1)$ -у шагу, а значение U^{n+1} присваивается n -у. Параметр a для фильтрации выбран равным 0.05 [9].

Для донного трения в системе уравнений (1) используется неявная схема Эйлера по времени, что повышает устойчивость численного алгоритма.

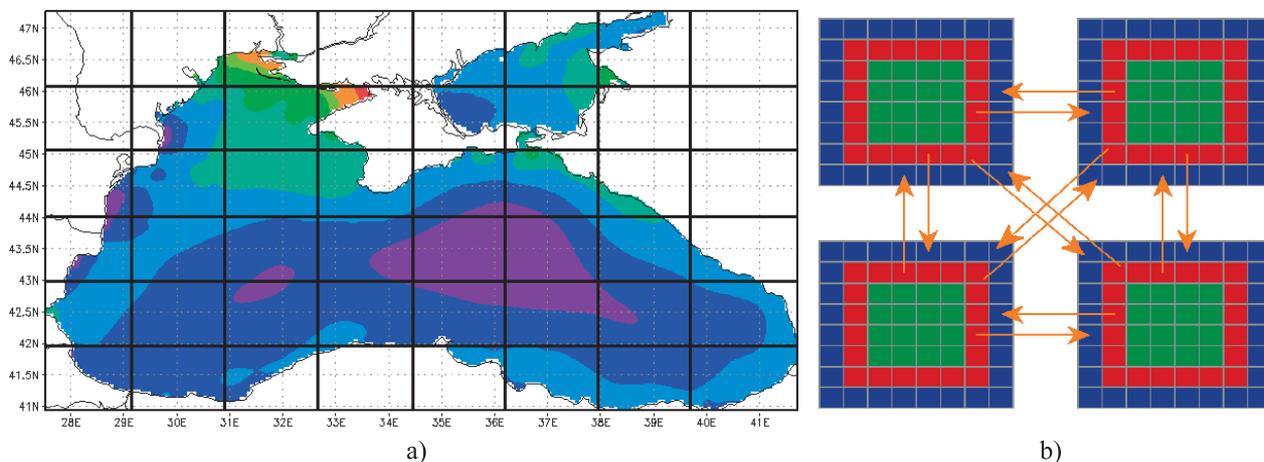


Рис. 2. Метод декомпозиции области, равномерное разбиение на прямоугольные подобласти (а); синхронизация процессоров, красные точки — это граница подобласти, синие — внерасчетная граница (б)

4. Метод декомпозиции области. В качестве основной идеи параллельного алгоритма решения нелинейных уравнений мелкой воды (1) используется метод декомпозиции области. Суть его в следующем: расчетная область разбивается на подобласти, которые ставятся в соответствие каждому процессору. Для каждой подобласти добавляется внерасчетная граница толщиной в одну узловую точку. В начале каждого расчета происходит синхронизация между процессорами для обновления значений на внерасчетных границах. При синхронизации пограничные блоки процессора помещаются во внерасчетные границы своих соседних процессоров (рис. 2). Когда процессор в своих вычислениях подходит к границе своей подобласти и ему требуются данные с соседней подобласти, он берет эти данные со своей внерасчетной границы и продолжает вычисления [2, 10, 11].

В качестве базового алгоритма разбиения расчетной области используется равномерное разбиение на прямоугольные подобласти (см. рис. 2). Рассмотрим расчетную область, состоящую из множества точек $\{(i, j) : 1 \leq i \leq n_x, 1 \leq j \leq n_y\}$, где n_x — количество точек по x и n_y — количество точек по y . Пусть используется p процессоров, образующих сетку $p = p_x \times p_y$, где p_x — количество процессоров по x и p_y — количество процессоров по y .

Тогда при равномерном разбиении каждому процессору с координатами (p_1, p_2) ставится в соответствие прямоугольная подобласть из множества точек $\{(i, j) : n_{x,start} \leq i \leq n_{x,end}, n_{y,start} \leq j \leq n_{y,end}\}$, где

$$\begin{aligned} n_{x,start} &= \left\lfloor \frac{n_x}{p_x} \right\rfloor (p_1 - 1) + 1, & n_{x,end} &= \left\lfloor \frac{n_x}{p_x} \right\rfloor (p_1), \\ n_{y,start} &= \left\lfloor \frac{n_y}{p_y} \right\rfloor (p_2 - 1) + 1, & n_{y,end} &= \left\lfloor \frac{n_y}{p_y} \right\rfloor (p_2). \end{aligned} \quad (6)$$

Кроме того, модель поддерживает декомпозицию на более сложные области: произвольные прямоугольные многоугольники. Для этого в модели был реализован блочный подход. Суть его в следующем: исходная область равномерно разбивается на прямоугольные блоки, границы блоков рассчитываются по формулам (6). Далее, каждому процессору ставится в соответствие некоторое количество блоков, которые и формируют расчетную подобласть процессора. Примеры таких разбиений будут показаны ниже при рассмотрении балансировки нагрузки области. У блочного подхода есть несколько преимуществ. Во-первых, это возможность проводить декомпозицию на более сложные области, обеспечивающую балансировку нагрузки на процессоры. Во-вторых, выбирая оптимальные размеры блока, можно получить ускорение модели за счет попадания каждого блока в кэш-память. В-третьих, данный подход довольно несложно программно реализуем и его можно встроить в сложные коды, такие как модель океана, без существенных изменений. Для этого достаточно основные циклы программы на языке Fortran

```
do m = nx_start, nx_end
  do n = ny_start, ny_end
    var(m, n) = <some computations>
  enddo
enddo
```

заменить на

```
do k = 1, blocks_count
  call set_block_boundary(k)
  do m = nx_start, nx_end
    do n = ny_start, ny_end
      var(m, n) = <some computations>
    enddo
  enddo
enddo
```

Здесь var — это некоторый массив данных, который может содержать рассчитанное отклонение уровня или компоненты вектора скорости и др. Процедура set_block_boundary(k) устанавливает границы текущего блока k и смещает указатели на массивы данных в соответствии с ними; таким образом, все основные переменные “смотрят” в текущий блок и структура циклов полностью сохраняется.

Синхронизация внерасчетных границ блоков реализована неблокирующими вызовами MPI_Isend и MPI_Irecv, причем, если соседние блоки расположены на одном процессоре, то вызовов MPI-процедур не происходит — вместо этого происходит обычное копирование с границы одного блока на внерасчетную границу другого блока. Кроме того, возможно использование гибридного подхода MPI + OpenMP, где каждый OpenMP-поток обрабатывает по одному блоку. Похожий подход уже зарекомендовал себя в таких моделях океана, как Parallel Ocean Program (POP) [13, 15, 20] и HIROMB (High Resolution Operational Model for the Baltic Sea) [16].

5. Балансировка нагрузки вычислений.

5.1. Метод разбиения вдоль фрактальной кривой. Метод разбиения вдоль фрактальной кривой основывается на фрактальных кривых, заполняющих пространство (Space-Filling Curve, SFC). Такие кривые преобразуют d -мерное пространство в одномерное с сохранением свойства локальности, т.е. соседние элементы в d -мерном пространстве преобразуются в соседние элементы в одномерном пространстве. Существует множество кривых, заполняющих пространство [14]. На практике используются кривые Пеано и их частные случаи — кривые Гильберта и кривые Мортон [12, 13, 20]. В рамках применения метода разбиения вдоль фрактальной кривой к решению нелинейных уравнений мелкой воды мы будем рассматривать кривые Гильберта в двумерном пространстве.

Кривые Гильберта используются для преобразования двумерной области с размерами $nb_x \times nb_y$ в кривую, где $nb_x \times nb_y = 2^n$ и n — это целое число, которое называется индексом кривой Гильберта. Рис. 3 показывает пример кривых Гильберта с индексами 2, 4, 8.

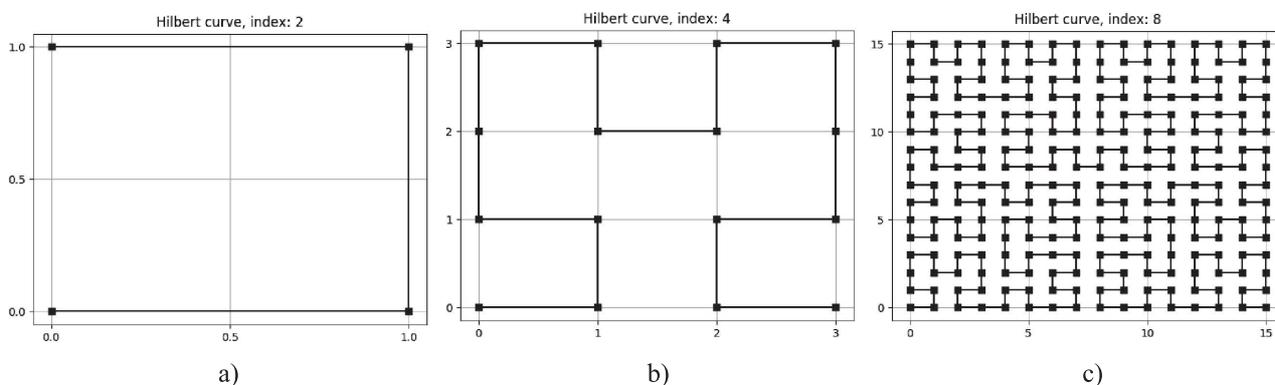


Рис. 3. Кривые Гильберта с индексами 2 (а), 4 (б) и 8 (с) соответственно

Алгоритм балансировки нагрузки с помощью кривых Гильберта следующий. Предварительно вся расчетная область с размерами $n_x \times n_y$ точек равномерно разбивается на сетку с размерами $nb_x \times nb_y = 2^n$, состоящую из прямоугольных блоков. Здесь в качестве алгоритма разбиения берется базовый алгоритм равномерного разбиения на прямоугольные подобласти (6), только в этом случае разбиение расчетной

области идет не для сетки процессоров, а для сетки блоков. При таком разбиении получается, что размер каждого блока будет $nb_1 \times nb_2$, где $nb_1 = \left\lceil \frac{n_x}{nb_x} \right\rceil$ и $nb_2 = \left\lceil \frac{n_y}{nb_y} \right\rceil$.

Для каждого блока рассчитывается значение загруженности блока w_i как сумма всех точек, которые не лежат на суше. Далее, на сетке блоков проводится кривая Гильберта, которая переводит двумерное пространство блоков в одномерное. Разбиение на подобласти происходит вдоль кривой и причем таким образом, чтобы подобласти имели примерно одинаковую сумму загруженности блоков. Затем каждому процессору ставится в соответствие его подобласть, на которой он проводит вычисления. Блоки, которые полностью состоят из точек на суше (т.е. загруженность которых $w_i = 0$), в распределении по процессорам и в дальнейших вычислениях не участвуют. На рис. 4 наглядно показаны шаги описанного алгоритма.

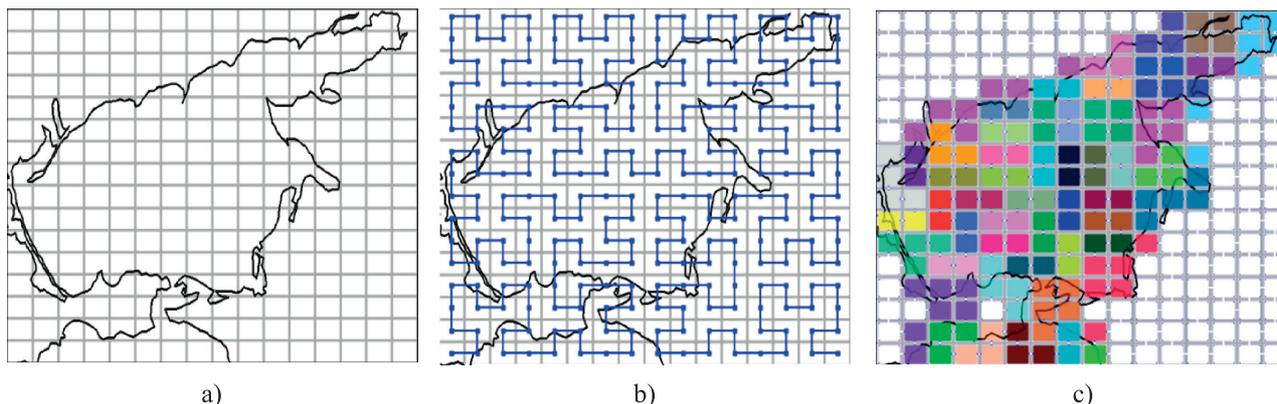


Рис. 4. Алгоритм балансировки нагрузки с помощью кривых Гильберта: а) равномерное разбиение на блоки, б) на сетке блоков проводится кривая Гильберта, в) блоки объединяются в подобласти вдоль кривой Гильберта. Каждому процессору ставится в соответствие его подобласть

Пример такого разбиения в сравнении с равномерным разбиением без балансировки нагрузки и с разбиением, полученным с помощью библиотеки METIS, приведен на рис. 5 для акватории Азовского моря с размерами 1525×1115 точек, сетка блоков 32×32 . Черным цветом на рисунке изображены блоки, состоящие только из точек на суше и которые не участвуют в вычислениях.

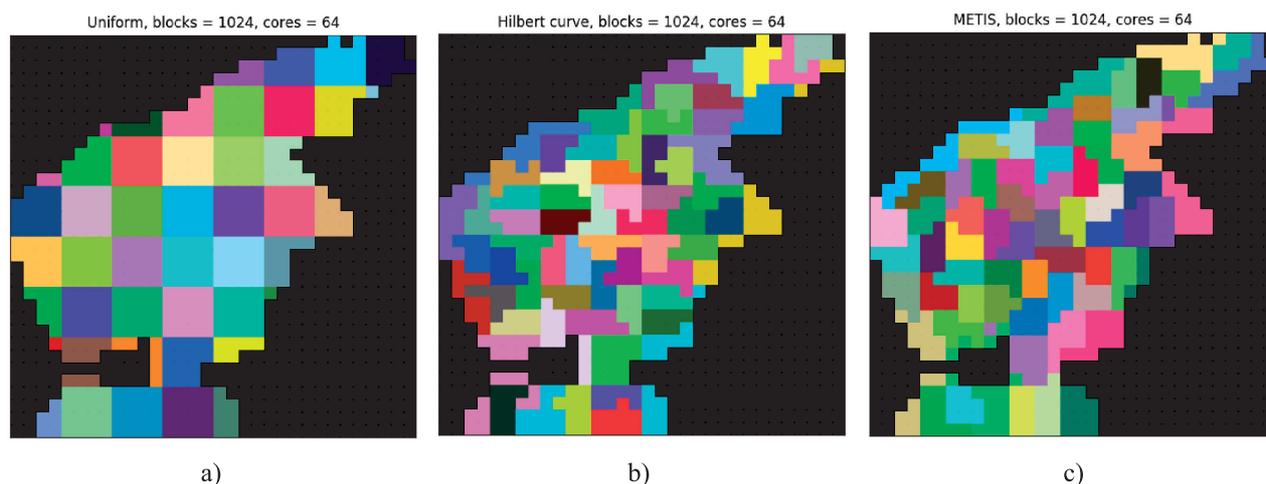


Рис. 5. Различные разбиения для 64 процессоров, 1024 блоков: а) равномерное разбиение, б) метод разбиения с использованием кривой Гильберта, в) разбиение, полученное с помощью METIS

Следует отметить, что у описанного метода балансировки нагрузки вычислений с использованием кривых Гильберта есть несколько недостатков: ограничение на сетку блоков (сетка должна быть размерами $nb_x \times nb_y = 2^n$); большое количество пересылок при большом количестве блоков на процессоре.

5.2. Метрики качества разбиения. Для того чтобы понять качество полученного разбиения, полезно иметь некоторые метрики [12, 13]. В этой части мы введем пару метрик качества разбиения. Предположим, что разбиение происходит на k подобластей для p процессоров. Введем первую метрику LB ,

которая будет отвечать за сбалансированность разбиения с точки зрения нагрузки вычислений на процессоры:

$$LB = \frac{\max_{1 \leq i \leq k} W_i}{\frac{1}{p} \sum_{i=1}^k W_i},$$

где $\max_{1 \leq i \leq k} W_i$ — максимальная загруженность i -й подобласти, $\sum_{i=1}^k W_i$ — полная загруженность всей расчетной области. Эта величина показывает отношение максимальной загруженности подобласти в разбиении к оптимальной загруженности. Значение $LB = 1$ соответствует идеально сбалансированному разбиению.

Введем также вторую метрику r_M , которая будет отвечать за качество разбиения с точки зрения коммуникаций между соседними подобластями:

$$r_M = \max_{1 \leq i \leq k} \frac{e_i}{s_i},$$

где e_i — количество узлов i -й подобласти, соседних для некоторой другой подобласти; s_i — количество всех узлов i -й подобласти. Эта величина показывает максимальное отношение периметра к объему подобласти. Для разбиений с завышенной величиной r_M время на коммуникации между процессорами становится доминирующим. Библиотека METIS по умолчанию пытается построить разбиение, минимизирующее именно такую метрику.

6. Численные эксперименты. Тестирование описанного метода балансировки нагрузки вычислений применительно к параллельному алгоритму решения уравнений мелкой воды проводилось на суперкомпьютере “Ломоносов”. Тестирование проводилось на группе вычислительных узлов regular4, состоящих из 8-ядерных процессоров Intel Xeon X5570 2.93GHz и с 12 Гб оперативной памяти [18].

В качестве расчетной области для тестирования была выбрана акватория Азовского моря с пространственным разрешением ~ 250 м и с размером сеточной области 1525×1115 точек. Эта область интересна тем, что включает в себя значительную часть суши (см. рис. 5), и описанный метод балансировки нагрузки здесь будет особенно актуален. В табл. 1–3 показаны метрики качества различных разбиений для этой области. Видно, что метрика LB , отвечающая за сбалансированность разбиения с точки зрения вычислений, может отличаться более чем в два раза для равномерного разбиения и для разбиения, полученного методом балансировки нагрузки вычислений.

Таблица 1
Метрики LB и r_M для метода разбиения с использованием кривой Гильберта

Cores	4			16		
Blocks per core	4	16	64	4	16	64
LB	1.0755	1.0535	1.0160	1.2012	1.0655	1.0173
$r_M, \%$	0.985%	1.279%	1.158%	1.975%	2.558%	2.966%
Cores	64			256		
Blocks per core	4	16	64	4	16	64
LB	1.2063	1.0640	1.0153	1.2890	1.0651	1.2846
$r_M, \%$	5.003%	5.417%	5.736%	10.13%	10.88%	12.98%

Таблица 2
Метрики LB для различных разбиений

Cores	4	16	64	128	256
Blocks per core	16	16	16	8	16
Uniform	1.5278	2.3872	2.4025	2.4025	2.4025
Hilbert	1.0535	1.0655	1.0640	1.2012	1.0651
METIS	1.0384	1.0511	1.1080	1.1714	1.1006

Таблица 3

Метрики r_M для различных разбиений

Cores	4	16	64	128	256
Blocks per core	16	16	16	8	16
Uniform	0.311%	1.248%	2.501%	3.554%	5.003%
Hilbert	1.279%	2.558%	5.417%	7.300%	10.88%
METIS	0.716%	2.501%	5.185%	8.719%	12.13%

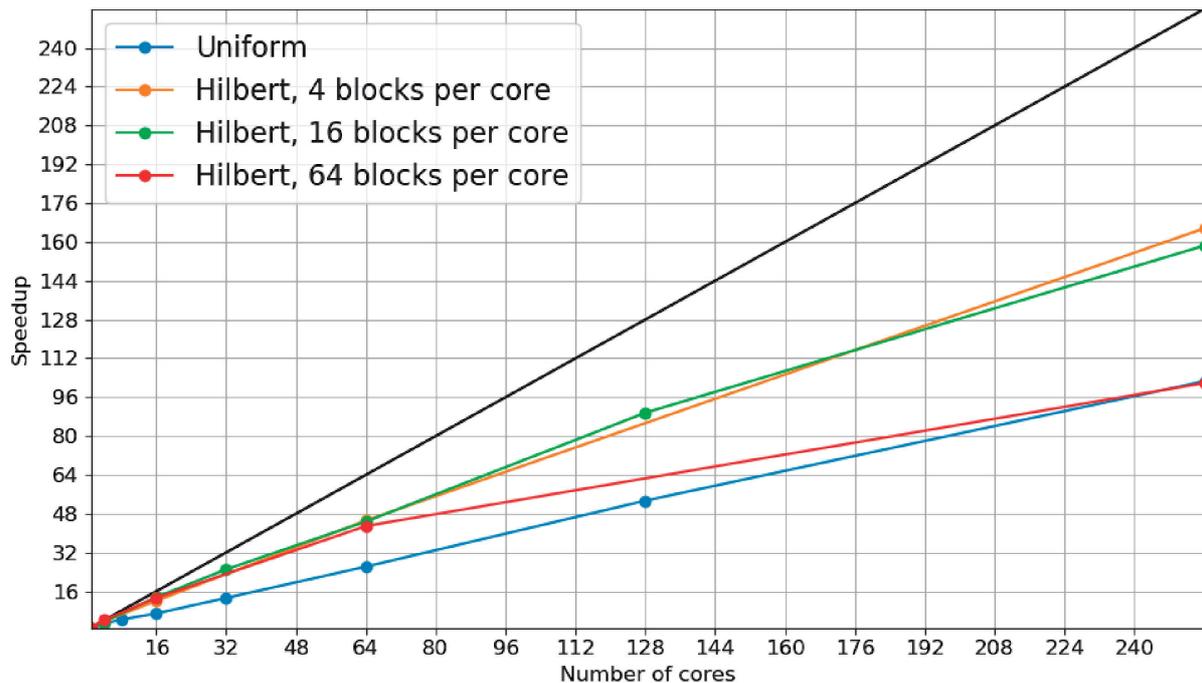


Рис. 6. Ускорение метода разбиения с использованием кривых Гильберта в сравнении с равномерным разбиением. Черная линия соответствует линейному ускорению

На рис. 6 показано ускорение параллельной версии программы с методом балансировки, использующим кривые Гильберта, в сравнении с равномерным разбиением без балансировки. Метод балансировки применялся к различным размерам используемой сетки блоков, рассматривались следующие сетки: по 4, 16 и 64 блока на ядро. Из рисунка видно, что для данной задачи наиболее оптимальные сетки блоков — это по 4 и 16 блоков на каждое ядро, в табл. 1 представлены метрики качества различных разбиений. Видно, что в среднем с ростом количества блоков на ядро значения метрики сбалансированности вычислений LB уменьшаются и приближаются к единице, что соответствует идеально сбалансированному разбиению. Однако для разбиения с 64 блоками на ядро видно заметное уменьшение ускорения на рис. 6 — это связано с тем, что коммуникационные задержки на пересылки между блоками становятся существенными, хотя и подобласти при таком разбиении могут быть более сбалансированными. Из полученных результатов видно, что при выборе размера сетки блоков важно соблюдать баланс между хорошей сбалансированностью разбиения, с одной стороны, и коммуникационными задержками на синхронизацию блоков, с другой.

Было также проведено сравнение метода балансировки нагрузки вычислений, использующего кривую Гильберта, с библиотекой METIS. На рис. 7 показано ускорение параллельной программы с использованием этих методов, в табл. 2 и 3 приведены метрики качества полученных разбиений. На рис. 5 показаны разбиения для 64 ядер. Методы рассматривались на сетках блоков: по 8 блоков на ядро для 32 и 128 ядер и по 16 блоков на ядро для всех остальных. Видно, что метод балансировки с использованием кривых Гильберта дает разбиения, очень близкие к разбиениям METIS как по значениям метрик качества разбиения, так и по значениям полученного ускорения параллельной программы. Для 256 ядер видно даже, что по всем метрикам качества разбиение, полученное с помощью кривых Гильберта, немного лучше, чем METIS. Поэтому можно сделать вывод, что для данной задачи метод балансировки нагрузки вычислений

с использованием кривой Гильберта — это хорошая альтернатива библиотеке METIS.

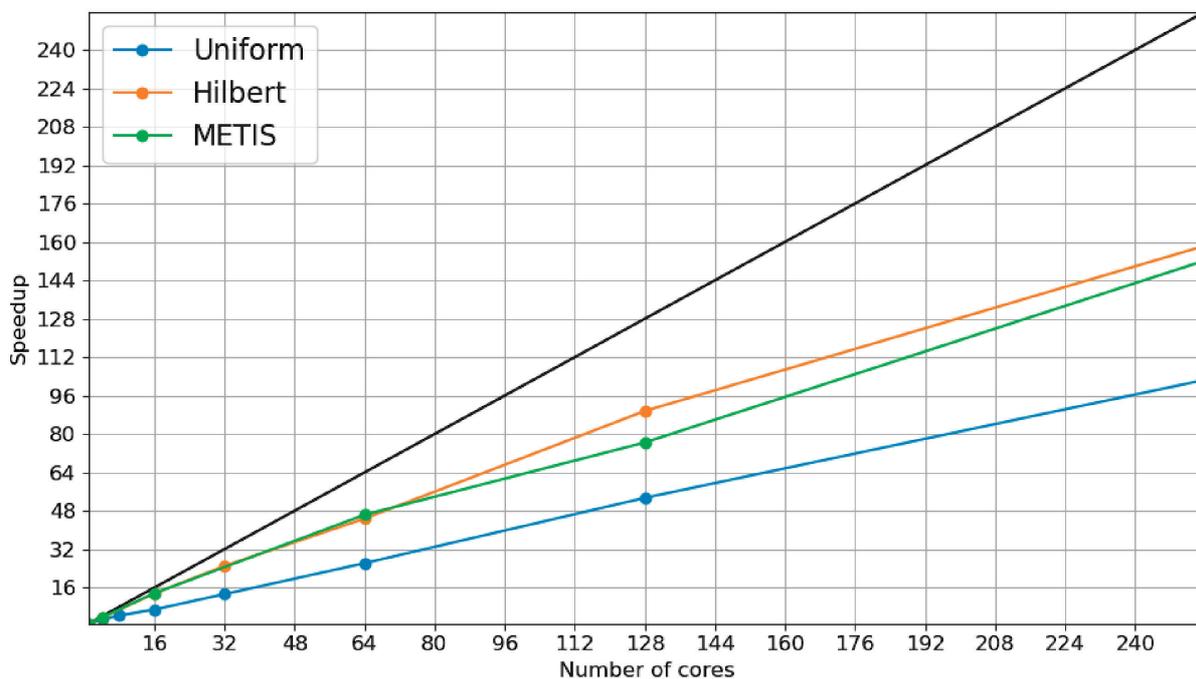


Рис. 7. Ускорение метода разбиения с использованием кривых Гильберта в сравнении с равномерным разбиением и METIS. Черная линия соответствует линейному ускорению

Отметим, что метод балансировки с использованием кривых Гильберта строит разбиения в несколько раз быстрее METIS, но для рассматриваемой задачи это незначительно, так как расчетное время на порядок больше, чем время построения разбиения. Однако алгоритм балансировки с использованием кривых Гильберта будет давать существенные преимущества по сравнению с METIS, когда вычислительные затраты будут существенно зависеть от временных затрат на построение разбиения, например на очень большой расчетной области и с адаптивной сеткой. Кроме того, описанный алгоритм балансировки предпочтительнее, чем METIS, так как он реализован непосредственно в модели и его реализация довольно простая в отличие от того, что реализовано в METIS. METIS — это библиотека для разбиений вообще произвольного вида, в ней реализованы довольно громоздкие методы, основанные на графах, тогда как балансировка с использованием кривых Гильберта — это геометрический метод, который значительно проще и очень хорошо подходит именно для рассматриваемой задачи. При использовании описанного алгоритма балансировки нагрузки у модели не будет привязки к сторонней библиотеке, поэтому дальнейшая разработка, поддержка и улучшение этого инструмента будут легче, чем работа с METIS.

7. Моделирование шторма на Азовском море. С помощью рассмотренного параллельного алгоритма решения уравнений мелкой воды было проведено моделирование экстремального шторма на Азовском море, произошедшего 24 марта 2013 г., с помощью вышеописанной модели мелкой воды с разрешением 250 метров. Атмосферные данные были взяты из расчетов по WRF (Weather Research and Forecasting Model) из работы [23]. Моделирование проводилось на 3 месяца с шагом по времени 1 секунда, использовался коэффициент донного трения $n = 0.025$. Было проведено сравнение с экспериментальными данными и сравнение результатов расчетов по нелинейным (1) и линеаризованным уравнениям мелкой воды (рис. 8) [21]. Из рисунка видно, что обе модели хорошо согласуются с данными наблюдений. Показано, что вклад нелинейности для этой задачи незначителен.

8. Заключение. Реализована балансировка нагрузки вычислений по процессорам с использованием кривых Гильберта применительно к алгоритму решения уравнений мелкой воды. Был реализован блочный подход, который позволяет без особых трудностей внедрить рассмотренный алгоритм в сложные программы, например такие, как модель океана. Показана большая эффективность разработанного метода в сравнении с равномерным разбиением без балансировки нагрузки, ускорение более чем на 35%. Однако, чтобы получить хорошее ускорение, важно соблюдать баланс при выборе размера сетки блоков: чем больше приходится блоков на процессор, тем более оптимальным становится сбалансированность разбиения, с одной стороны, но возрастают коммуникационные задержки на синхронизацию блоков, с другой.

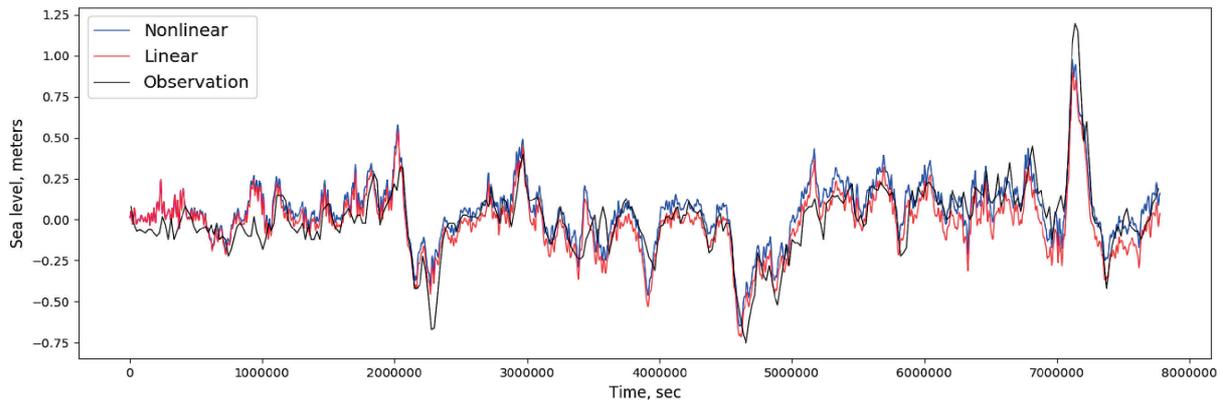


Рис. 8. Воспроизведение экстремального шторма от 24.03.2013 в акватории Азовского моря. Показано сравнения уровня (м) для поста “Ейск”. Синий — нелинейные уравнения мелкой воды; красный — линейаризованные уравнения; черный — наблюдения

Показано, что метод балансировки, использующий кривую Гильберта, дает разбиение, очень близкое по характеристикам к разбиениям, полученным с помощью библиотеки METIS. Параллельный алгоритм был применен к расчету баротропной циркуляции Азовского моря. При воспроизведении одного из самых экстремальных штормов на Азовском море было показано, что модель хорошо согласуется с данными наблюдений. Важно отметить, что оптимальность разбиения для мелкой воды точно соответствует оптимальности для трехмерной сигма-модели INMOM, так как количество расчетных уровней по глубине в ней одинаково для всех точек сетки по горизонтали. Это позволяет без особых изменений реализовать предложенный алгоритм балансировки нагрузки вычислений на процессорах для модели общей циркуляции океана INMOM.

Работа выполнена при поддержке Российского научного фонда (грант № 17-77-30001) и с использованием оборудования Центра коллективного пользования сверхвысокопроизводительными вычислительными ресурсами МГУ им. М. В. Ломоносова.

СПИСОК ЛИТЕРАТУРЫ

1. Дианский Н.А. Моделирование циркуляции океана и исследование его реакции на короткопериодные и долгопериодные атмосферные воздействия. М.: Физмалит, 2012.
2. Чаплыгин А.В. Параллельная реализация общей модели циркуляции океана INMOM // Сборник тезисов лучших выпускных квалификационных работ факультета ВМК МГУ 2017. Москва: МАКС Пресс, 2017. 27–28.
3. Сухов В.Б. О решении некоторых задач моделирования крупномасштабной динамики океана. Диссертация на соиск. уч. ст. к.ф.-м.н., 2009.
4. Imamura F., Yalciner A.C., Ozyurt G. Tsunami modelling manual (TUNAMI model). Sendai: Tohoku Univ. Press, 2006.
5. Рочк П. Вычислительная гидродинамика. М: Мир, 1980.
6. Arakawa A., Lamb V.R. Computational design of the basic dynamical processes of the UCLA general circulation model // Methods in Computational Physics. Vol. 17. New York: Academic Press, 1977. 173–265.
7. Mesinger F., Arakawa A. Numerical methods used in atmospheric models. Vol. 1. Geneva: GAPP Publ., 1976.
8. Марчук Г.И. Методы вычислительной математики. М: Наука, 1989.
9. Mellor G.L. User's guide for a three-dimensional, primitive equation, numerical ocean model. Princeton: Princeton Univ. Press, 1993.
10. Wang G., Qiao F., Xia C. Parallelization of a coupled wave-circulation model and its application // Ocean Dynamics. 2010. 60, N 2. 331–339.
11. Чаплыгин А.В., Дианский Н.А., Гусев А.В. Параллельное моделирование нелинейных уравнений мелкой воды // Труды 60-й Всероссийской научной конференции МФТИ. Долгопрудный: МФТИ, 2017. 192–194.
12. Liu H., Wang K., Yang B., Yang M., He R., Shen L., Zhong H., Chen Z. Load balancing using Hilbert space-filling curves for parallel reservoir simulations. ArXiv preprint: 1708.01365v1 [cs.DC]. Ithaca: Cornell Univ. Library, 2017. <https://arxiv.org/abs/1708.01365>.
13. Dennis J.M. Inverse space-filling curve partitioning of a global ocean model // IEEE International Parallel and Distributed Processing Symposium. New York: IEEE Press, 2007. doi 10.1109/IPDPS.2007.370215
14. Sagan H. Space-filling curves. New York: Springer, 1994.

15. *Smith R., Jones P., Briegleb B., et al.* The parallel ocean program (POP) reference manual: ocean component of the Community Climate System Model (CCSM) and Community Earth System Model (CESM). <http://www.cesm.ucar.edu/models/cesm1.0/pop2/doc/sci/POPRefManual.pdf>.
16. *Wilhelmsson T.* Parallelization of the HIROMB ocean model. <https://pdfs.semanticscholar.org/ee95/bela6bb90becdc31c84f83c343ca8daf5bdc.pdf>.
17. *Rantakokko J.* A framework for partitioning domains with inhomogeneous workload // *Parallel Algorithms and Applications*. 1998. **13**, N 2, 135–151.
18. *Воеводин В.В., Жуматий С.А., Соболев С.И., Антонов А.С., Брызгалов П.А., Никитенко Д.А., Стефанов К.С., Воеводин Вад.В.* Практика суперкомпьютера “Ломоносов” // *Открытые системы*. 2012. № 7. 36–39.
19. *Karypis G.* METIS: a software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices. Version 5.1.0. Minneapolis: Univ. of Minnesota, 2013.
20. *van Werkhoven B., Maassen J., Kliphuis M., Dijkstra H.A., Brunmabend S.E., van Meersbergen M., Seinstra F.J., Bal H.E.* A distributed computing approach to improve the performance of the parallel ocean program // *Geosci. Model Dev*. 2014. **7**, N 1. 267–281.
21. *Фомин В.В., Дианский Н.А., Чаплыгин А.В.* Расчет экстремальных нагонов в Таганрогском заливе и использование моделей циркуляции атмосферы и океана различного пространственного разрешения // *Труды VI Международной научно-практической конференции “Морские исследования и образование: MARESEDU-2017”*. <https://www.maresedu.com/materials>
22. *Володин Е.М., Дианский Н.А., Гусев А.В.* Воспроизведение и прогноз климатических изменений в XIX–XXI веках с помощью модели земной климатической системы ИВМ РАН // *Изв. РАН. Физика атмосферы и океана*. 2013. **49**, № 4. 379–400.
23. *Фомин В.В., Дианский Н.А.* Расчет экстремальных нагонов в Таганрогском заливе с использованием моделей циркуляции атмосферы и океана // *Метеорология и гидрология*. 2018. № 12. 69–80.
24. *Shchepetkin A.F., McWilliams J.C.* The regional oceanic modeling system (ROMS): a split-explicit, free-surface, topography-following-coordinate oceanic model // *Ocean Modelling*. 2005. **9**, N 4. 347–404.
25. *Ушаков К.В., Ибраев Р.А., Калмыков В.В.* Воспроизведение климата Мирового океана с помощью массивно-параллельной численной модели // *Известия РАН. Физика атмосферы и океана*. 2015. **51**, № 4. 416–436.

Поступила в редакцию
11.02.2019

Load Balancing Using Hilbert Space-Filling Curves for Parallel Shallow Water Simulations

A. V. Chaplygin¹, N. A. Diansky², and A. V. Gusev³

¹ *Lomonosov Moscow State University, Faculty of Computational Mathematics and Cybernetics; Leninskie Gory, Moscow, 119992, Russia; Student, e-mail: achaplygin99@gmail.com*

² *Lomonosov Moscow State University, Faculty of Physics; Leninskie Gory, Moscow, 119991, Russia; Dr. Sci., Principal Scientist, e-mail: nikolay.diansky@gmail.com*

³ *Marchuk Institute of Numerical Mathematics, Russian Academy of Sciences; ulitsa Gubkina 8, Moscow, 119333, Russia; Ph.D., Senior Scientist, e-mail: anatoly.v.gusev@gmail.com*

Received February 11, 2019

Abstract: This paper presents a method of load balancing using Hilbert space-filling curves applied to a parallel algorithm for solving shallow water equations. We consider the system of shallow water equations in the form presented in the ocean general circulation sigma-model INMOM (Institute of Numerical Mathematics Ocean Model). This system of equations is one of the basic blocks of the model. Due to land points in the computational grid, the load balancing is an especially urgent task. The method of load balancing using Hilbert space-filling curves is chosen as one of such methods. The paper demonstrates the greater efficiency of this method in comparison with the uniform partitioning without load balancing. It is shown that this method is a good alternative to the METIS standard library. Moreover, the optimality of the implemented partition for the shallow water equations exactly corresponds to the optimality for the INMOM three-dimensional sigma-model due to the same number of vertical levels in the entire computational domain.

Keywords: shallow water equations, domain decomposition method, parallel computing, load balancing, Hilbert space-filling curves.

References

1. N. A. Diansky, *Modeling of Ocean Circulation and Analysis of Its Reaction on Short- and Long-Term Atmospheric Effects* (Fizmatlit, Moscow, 2013) [in Russian].
2. A. V. Chaplygin, "Parallel Implementation of the INMOM General Model of Ocean Circulation," in *Collection of the Best Qualification Works in the Faculty of Computational Mathematics and Cybernetics, Moscow State University* (Maks Press, Moscow, 2017), pp. 27–28.
3. V. B. Sukhov, *Solving Some Problems of Modeling the Large-Scale Dynamics of Ocean*, Candidate's Dissertation in Mathematics and Physics (Moscow State Univ., Moscow, 2009).
4. F. Imamura, A. C. Yalciner, and G. Ozyurt, *Tsunami Modelling Manual (TUNAMI Model)* (Tohoku Univ. Press, Sendai, Japan, 2006).
5. P. Roach, *Computational Fluid Dynamics* (Hermosa Publ., Albuquerque, 1972; Mir, Moscow, 1980).
6. A. Arakawa and V. R. Lamb, "Computational Design of the Basic Dynamical Processes of the UCLA General Circulation Model," in *Methods in Computational Physics: Advances in Research and Applications* (Academic Press, New York, 1977), Vol. 17, pp. 173–265.
7. F. Mesinger and A. Arakawa, *Numerical Methods Used in Atmospheric Models* (GAPR Publ., Geneva, 1976), Vol. 1.
8. G. I. Marchuk, *Methods of Numerical Mathematics* (Nauka, Moscow, 1989; Springer, New York, 1982).
9. G. L. Mellor, *User's Guide for a Three-dimensional, Primitive Equation, Numerical Ocean Model* (Princeton Univ. Press, Princeton, 1993).
10. G. Wang, F. Qiao, and C. Xia., "Parallelization of a Coupled Wave-Circulation Model and Its Application," *Ocean Dyn.* **60** (2), 331–339 (2010).
11. A. V. Chaplygin, N. A. Diansky, and A. V. Gusev, "Parallel Modeling of Nonlinear Shallow Water Equation," in *Proc. 60th All-Russia Conf. on Applied Mathematics and Informatics, Moscow Institute of Physics and Technology, Dolgoprudny, Russia, November 20–26, 2017* (Moscow Inst. Phys. Technol., Dolgoprudny, 2017), pp. 192–194.
12. H. Liu, K. Wang, B. Yang, et al., *Load Balancing Using Hilbert Space-Filling Curves for Parallel Reservoir Simulations*, arXiv preprint: 1708.01365v1 [cs.DC] (Cornell Univ. Library, Ithaca, 2017), available at <https://arxiv.org/abs/1708.01365>.
13. J. M. Dennis, "Inverse Space-Filling Curve Partitioning of a Global Ocean Model," in *IEEE Int. Parallel and Distributed Processing Symposium, Rome, Italy, March 26-30, 2007* (IEEE Press, New York, 2007), doi 10.1109/IPDPS.2007.370215
14. H. Sagan, *Space-Filling Curves* (Springer, New York, 1994).
15. R. Smith, P. Jones, B. Briegleb, et al., *The Parallel Ocean Program (POP) Reference Manual: Ocean Component of the Community Climate System Model (CCSM) and Community Earth System Model*. <http://www.cesm.ucar.edu/models/cesm1.0/pop2/doc/sci/POPRefManual.pdf>. Cited February 28, 2019.
16. T. Wilhelmsson, *Parallelization of the HIROMB Ocean Model*. <https://pdfs.semanticscholar.org/ee95/be1a6bb90becdc31c84f83c343ca8daf5bdc.pdf>. Cited February 28, 2019.
17. J. Rantakokko, "A Framework for Partitioning Domains with Inhomogeneous Workload," *Parallel Algorithms Appl.* **13** (2), 135–151 (1998).
18. Vl. V. Voevodin, S. A. Zhumatii, S. I. Sobolev, et al., "Practice of Lomonosov Supercomputer," *Otkrytye Sistemy*, No. 7, 36–39 (2012).
19. G. Karypis, *METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices* (Univ. of Minnesota, Minneapolis, 2013), Version 5.1.0.
20. B. van Werkhoven, J. Maassen, M. Kliphuis, et al., "A Distributed Computing Approach to Improve the Performance of the Parallel Ocean Program," *Geosci. Model Dev.* **7** (1), 267–281 (2014).
21. V. V. Fomin, N. A. Dianskii, and A. V. Chaplygin, "Simulation of Extreme Surges in the Taganrog Bay and the Usage of Atmosphere and Ocean Circulation Models of Various Spatial Resolution," in *Proc. VI Int. Conf. on Marine Studies and Education, Moscow, Russia, October 30–November 2, 2017* (PoliPRESS, Tver, 2017), <https://www.maresedu.com/materials>. Cited February 28, 2019.
22. E. M. Volodin, N. A. Diansky, and A. V. Gusev, "Simulation and Prediction of Climate Changes in the 19th to 21st Centuries with the Institute of Numerical Mathematics, Russian Academy of Sciences, Model of the Earth's Climate System," *Izv. Akad. Nauk, Fiz. Atmos. Okeana* **49** (4), 379–400 (2013) [*Izv., Atmos. Ocean. Phys.* **49** (4), 347–366 (2013)].
23. V. V. Fomin and N. A. Diansky, "Simulation of Extreme Surges in the Taganrog Bay with Atmosphere and Ocean Circulation Models," *Meteorol. Gidrol.*, No. 12, 69–80 (2018) [*Russ. Meteorol. Hydrol.* **43** (12),

843–851 (2018)].

24. A. F. Shchepetkin and J. C. McWilliams, “The Regional Oceanic Modeling System (ROMS): A Split-Explicit, Free-Surface, Topography-Following-Coordinate Oceanic Model,” *Ocean Model.* **9** (4), 347–404 (2005).

25. K. V. Ushakov, R. A. Ibrayev, and V. V. Kalmykov, “Simulation of the World Ocean Climate with a Massively Parallel Numerical Model,” *Izv. Akad. Nauk, Fiz. Atmos. Okeana* **51** (4), 416–436 (2015) [*Izv., Atmos. Ocean. Phys.* **51** (4), 362–380 (2015)].