

УДК 519.63

doi 10.26089/NumMet.v20r445

MPI+OpenMP РЕАЛИЗАЦИЯ МЕТОДА BiCGStab С ЯВНЫМ ПРЕДОБУСЛОВЛИВАНИЕМ ДЛЯ РЕШЕНИЯ РАЗРЕЖЕННЫХ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ

И. Е. Капорин¹, О. Ю. Милюкова²

Для предобусловливания несимметричной положительно определенной разреженной матрицы рассматривается ее приближенная обратная, представленная в виде произведения нижнетреугольной и верхнетреугольной матриц. Предлагается новый способ предобусловливания положительно определенной разреженной матрицы — метод блочного Якоби неполного обратного LU-разложения. Описан алгоритм параллельной реализации метода BiCGStab с предложенным предобусловливанием с применением MPI+OpenMP-технологии. Проводится сравнение времени решения тестовых задач из коллекции разреженных матриц SuiteSparse (ранее известной как коллекция университета Флориды) методом BiCGStab с предложенным предобусловливанием и с предобусловливанием Якоби, а также с предобусловливанием блочного Якоби в сочетании с неполным треугольным разложением без заполнения. При этом используются разработанные параллельные реализации на основе MPI- или MPI+OpenMP-подходов.

Ключевые слова: итерационное решение систем линейных уравнений, разреженные матрицы, неполное обратное треугольное разложение, параллельное предобусловливание, стабилизированный метод бисопряженных градиентов (BiCGStab).

1. Введение. Рассмотрим задачу приближенного решения системы линейных алгебраических уравнений (СЛАУ)

$$Ax = b \quad (1)$$

с положительно определенной разреженной $(n \times n)$ -матрицей A общего вида. Проблема построения соответствующих эффективных численных методов решения СЛАУ (1) сохраняет свою актуальность, так как во многих важных прикладных областях продолжают возникать новые постановки таких задач. При этом наблюдается тенденция к росту порядка матриц n , а также к увеличению их заполненности ненулевыми элементами. Ввиду значительных вычислительных затрат, решение задач с матрицами большого размера требует применения параллельных компьютеров.

В настоящей работе для решения СЛАУ (1) большого размера используется предобусловленный стабилизированный метод бисопряженных градиентов (BiCGStab: Biconjugate Gradient Stabilized) [12] с явными предобусловливателями, итерации которого осуществляются до выполнения условия

$$\|Mr_k\| \leq \varepsilon \|Mr_0\|, \quad (2)$$

где $r_k = b - Ax_k$. Здесь и далее через $M \approx A^{-1}$ обозначается матрица предобусловливания. Будут рассмотрены итерационные методы решения задачи (1) с положительно определенной (вообще говоря, несимметричной) матрицей A , использующие матрицу предобусловливания $M = H^T G$, где H и G — нижнетреугольные матрицы с ненулевой диагональю. Такое предобусловливание сокращенно будем называть IILU-разложением (Incomplete Inverse LU-factorization):

$$A = G^{-1} H^{-T} + Z, \quad (3)$$

где Z — матрица погрешности. В методе IILU-BiCGStab значительная доля вычислительной работы приходится на повторные умножения разреженной матрицы на вектор, а операции решения систем с треугольными матрицами отсутствуют. Поэтому реализующие его алгоритмы хорошо приспособлены к параллельной реализации с использованием MPI+OpenMP-подхода.

¹ Вычислительный центр им. А. А. Дородницына РАН, ул. Вавилова, 40, 119333, Москва; гл. науч. сотрудник, e-mail: igorkaporin@mail.ru

² Институт прикладной математики им. М. В. Келдыша РАН (ИПМ РАН), Миусская пл., 4А, 125047, Москва; вед. науч. сотрудник, e-mail: olgamilyukova@mail.ru

В работах [1–5] для предобусловливания симметричной положительно определенной разреженной матрицы использовались ее приближенные обратные матрицы, представленные в виде произведения двух взаимно сопряженных разреженных треугольных матриц. В работах [3–5] предложен новый метод предобусловливания — метод блочного Якоби неполного обратного треугольного разложения, а работах [4, 5] — способ параллельной реализации метода сопряженных градиентов с некоторыми явными предобусловливателями с использованием MPI+OpenMP-подхода. Расчеты рассмотренных тестовых задач из коллекции SuiteSparse (ранее коллекция университета Флориды) [6] показали, что использование OpenMP-технологии при параллельной реализации этих методов позволяет значительно ускорить вычисления при решении задач с достаточно большими разреженными матрицами на умеренном числе процессоров.

В постановке задачи (1) предполагается, что матрица A уже переупорядочена и отмасштабирована:

$$A = A_{SP} = D_L A_P D_R = D_L P \tilde{A} P^T D_R,$$

где P — матрица перестановки, а \tilde{A} представляет собой матрицу коэффициентов исходной задачи после симметризации структуры разреженности (дополнения нулевыми значениями в соответствующих позициях). Здесь можно использовать переупорядочения, применяемые для симметричных матриц. В настоящей работе применяются переупорядочения, уменьшающие среднюю ширину ленты матрицы, предложенные в работах [3, 7] для матриц с симметричной структурой разреженности. Подход, предложенный в этих работах, позволяет одновременно произвести разбиение области расчета на подобласти. В настоящей работе применяется двухстороннее масштабирование с использованием алгоритма [8]: $A_{SP} = D_L A_P D_R$, где масштабирующие диагональные матрицы D_L и D_R таковы, что L_1 -нормы всех строк и столбцов близки к единице. Заметим, что искомое решение первоначальной СЛАУ $\tilde{A}\tilde{x} = \tilde{b}$ с заданным вектором правой части $\tilde{b} = P^T D_L^{-1} b$ имеет вид $\tilde{x} = P^T D_R x$.

В настоящей работе мы используем предобусловливание положительно определенной матрицы A посредством неполного обратного треугольного разложения (3), предложенного в [9], так что предобусловливатель имеет вид

$$M = H^T G \approx A^{-1}. \tag{4}$$

Здесь H и G — разреженные нижние треугольные матрицы. Структуру разреженности этих матриц можно выбирать, например, совпадающей со структурой разреженности нижнетреугольной части матрицы A (или содержащейся в ней). В настоящей работе предлагается новый способ предобусловливания положительно определенной разреженной матрицы, в котором используется предобусловливание блочного Якоби (сокращенно, BJ), а для каждого блока предобусловливания блочного Якоби, соответствующего вычислениям на своем процессоре, используется ILU-предобусловливание [9]. Заметим, что если матрица положительно определена, то матрицы, являющиеся блоками на главной диагонали, т.е. подматрицы блочного предобусловливания, тоже являются положительно определенными. Такое предобусловливание сокращенно будем называть BJILU. При построении этого предобусловливателя и для его применения не требуется выполнять обмен информацией между процессорами.

В настоящей работе также рассмотрен метод BiCGStab с предобусловливанием блочного метода Якоби в сочетании с неполным треугольным разложением без заполнения BJILU0-BiCGStab. Напомним, что предобусловливатель неполного треугольного разложения без заполнения ILU0 для произвольной матрицы A определяется следующим образом [10]:

$$\begin{aligned} (A - (I + L)D^{-1}(I + U))_{i,j} &= 0, & A_{i,j} &\neq 0, \\ ((I + L)D^{-1}(I + U))_{i,j} &= 0, & A_{i,j} &= 0, \end{aligned}$$

где $(I + L)D^{-1}(I + U)$ — предобусловливатель, D — диагональная матрица, L — строго нижнетреугольная матрица со структурой заполненности строго нижнетреугольной части матрицы A , а U — строго верхнетреугольная матрица со структурой заполненности строго верхнетреугольной части матрицы A . Заметим, что эффективная реализация вычисления предобусловливателя BJILU0 и его обращения с использованием MPI+OpenMP-подходов сталкивается с серьезными затруднениями из-за существенно рекурсивного характера вычислений.

В недавней статье [11] (см. также цитированную там литературу) предлагается способ решения СЛАУ с несимметричной матрицей, который позволяет использовать MPI+OpenMP-технологии для реализации алгоритма. В [11] представлено предобусловливание, основанное на ILU-разложении матрицы коэффициентов с последующим построением приближенных обратных для соответствующих нижнего и верхнего треугольных сомножителей.

В настоящей работе описывается метод предобусловливания положительно определенной разреженной матрицы ИЛУ. Предлагается новый метод предобусловливания положительно определенной разреженной матрицы ВЖИЛУ. Проводится теоретическое исследование сходимости итераций при использовании ВЖ (блочного Якоби) предобусловливания положительно определенных матриц. Описывается алгоритм параллельной реализации метода ВИЛУ-BiCGStab с применением технологии MPI+OpenMP. Проводится сравнение времени решения тестовых задач из коллекции SuiteSparse [6] (ранее коллекция университета Флориды) с использованием только MPI- и MPI+OpenMP-подходов предложенным новым методом ВЖИЛУ-BiCGStab и методом с предобусловливанием Якоби (J-BiCGStab), в котором $M = D_A^{-1}$, где D_A — диагональная часть матрицы A , а также методом BiCGStab с предобусловливанием ВЖИЛУ0.

2. Стабилизированный метод бисопряженных градиентов (BiCGStab). Требуется решить СЛАУ (1). Пусть для положительно определенной матрицы A построена приближенная обратная вида (4). Алгоритм метода BiCGStab [12] с левым предобусловливанием (4) можно представить в следующем виде.

Алгоритм 1: BiCGStab с левым предобусловливанием для решения системы $Ax = b$

```

 $x_0 = 0$ 
 $r_0 = H^T G b$ 
 $p_1 = r_0$ 
for  $i = 1, 2, \dots, i_{\max}$  :
     $\rho_{i-1} = r_0^T r_{i-1}$ 
    if ( $i \geq 2$ ) then
         $\beta_{i-1} = \frac{\rho_{i-1} \alpha_{i-1}}{\rho_{i-2} \omega_{i-1}}$ 
         $p_i = r_{i-1} + (p_{i-1} - v_{i-1} \omega_{i-1}) \beta_{i-1}$ 
    end if
     $v_i = H^T G A p_i$ 
     $\alpha_i = \rho_{i-1} / r_0^T v_i$ 
     $x_{i-1/2} = x_{i-1} + p_i \alpha_i$ 
     $r_{i-1/2} = r_{i-1} - v_i \alpha_i$ 
     $s_i = H^T G A r_{i-1/2}$ 
     $\omega_i = s_i^T r_{i-1/2} / s_i^T s_i$ 
     $x_i = x_{i-1/2} + r_{i-1/2} \omega_i$ 
     $r_i = r_{i-1/2} - s_i \omega_i$ 
    проверка на сходимость:  $\|r_i\| \leq \varepsilon \|r_0\|$ 
end for

```

Здесь $0 < \varepsilon \ll 1$ и $r_k = H^T G(b - Ax_k)$ — предобусловленная невязка. Оценки сходимости метода BiCGStab можно найти в [13], см. также цитированные там источники. Метод может требовать числа матрично-векторных умножений примерно до 2 раз больше, чем GMRES (обобщенный метод минимальных невязок, Generalized Minimal Residual) [14], однако минимальные затраты на векторные операции делают его одним из наиболее конкурентоспособных итерационных решателей для несимметричных систем уравнений. Алгоритм 1 использует лишь операции умножения разреженных матриц на вектор, операции вычисления скалярных произведений и элементарные векторные операции. Поэтому принципиальная возможность его эффективной параллельной реализации с использованием MPI+OpenMP-подхода не вызывает сомнений даже в случае большого числа процессоров.

3. Неполное обратное разложение положительно определенных матриц. Опишем алгоритм построения нижнетреугольных невырожденных матриц G и H , впервые представленный в работе [9]. Здесь и далее предположим, что $G, H \in \Omega$, где Ω — множество нижнетреугольных разреженных матриц, ненулевые элементы которых могут быть расположены только в позициях $i, j_i(p)$, где

$$1 \leq j_i(1) < \dots < j_i(m_i) = i, \quad i = 1, 2, \dots, n.$$

Вводя ортопроекторы

$$P_i = \sum_{p=1}^{m_i} e_{j_i(p)} e_{j_i(p)}^T,$$

где через e_k обозначается k -й столбец единичной матрицы I , и определив матрицы

$$A_i = I - P_i + P_i A P_i, \quad i = 1, 2, \dots, n,$$

положим строки $g_i^T = e_i^T G$ матрицы G и столбцы $h_i = H^T e_i$ матрицы H^T равными

$$g_i^T = (e_i^T A_i^{-1} e_i)^{-1/2} e_i^T A_i^{-1}, \quad h_i = A_i^{-1} e_i (e_i^T A_i^{-1} e_i)^{-1/2}.$$

Рассмотрим вопрос о возможной близости матрицы $G^{-1}H^{-T}$ к матрице A . Через $\|R\| = \sqrt{\text{trace}(R^T R)}$ обозначим евклидову норму матрицы R . Можно показать [9], что среди всех матриц вида

$$A(L_1, L_2) = G^{-1}H^{-T} + G^T L_1 + L_2^T H, \quad \text{где } L_1, L_2 \in \Omega,$$

ближайшей в евклидовой норме к A является именно построенная выше матрица $A(0, 0) = G^{-1}H^{-T}$. Кроме того, если наложить специальное условие на структуру разреженности матриц G и H , а именно потребовать, чтобы

$$\Omega + \Omega^T = \Omega^T \Omega, \tag{5}$$

где $\Omega + \Omega^T$ и $\Omega^T \Omega$ — множества матриц вида $L_1 + L_2^T$ и $L_2^T L_1$ соответственно и $L_1, L_2 \in \Omega$, то справедливы равенства

$$(A)_{ij} = (G^{-1}H^{-T})_{ij},$$

где (i, j) — позиция любого ненулевого элемента матрицы из $\Omega + \Omega^T = \Omega^T \Omega$. Заметим, что условие (5) эквивалентно требованию отсутствия заполнения при выполнении разреженного исключения Гаусса над любой матрицей из $\Omega + \Omega^T$, переупорядоченной в инверсном порядке $(n, n-1, \dots, 1)$.

Практическое вычисление i -х столбцов матриц G^T и H^T можно осуществить, вычислив для каждого i треугольное разложение

$$A_i = P_i^T L_i U_i P_i, \quad \text{Diag}(L_i) = \text{Diag}(U_i) > 0, \tag{6}$$

(т.е., возможно использование перестановки P_i , при условии, что она оставляет на месте i -ю позицию, так что $P_i e_i = e_i$) по формулам

$$g_i = L_i^{-T} e_i, \quad h_i = U_i^{-1} e_i.$$

Таким образом, достаточно выполнить треугольное разложение матрицы A_i (которая фактически имеет порядок $m_i \ll n$, так как в позициях, отличных от $(i, j_i(p))$, она совпадает с единичной матрицей I_n) и затем решить две системы (тоже размера m_i) с полученными треугольными матрицами. Если это целесообразно, то для вычисления треугольного разложения с перестановками (6) можно использовать имеющуюся разреженность подматриц A_i , привлекая известные алгоритмы, см., например, [15].

4. Предобусловливание при помощи блочного метода Якоби в сочетании с неполным обратным разложением. Пусть матрица A отмасштабирована, переупорядочена и разбита на блоки, причем на блочной диагонали расположены p квадратных блоков размера n_s , $1 \leq s \leq p$, и $\sum_{s=1}^p n_s = n$.

Обозначим $k_s = n_1 + \dots + n_s$ и определим прямоугольные матрицы

$$W_s = [e_{k_{s-1}+1} \mid \dots \mid e_{k_s}],$$

столбцы которых являются единичными n -векторами, а $k_{s-1} + 1, \dots, k_s$ представляют собой индексы s -го блока. Рассмотрим матрицы размеров $n_s \times n_s$ вида $W_s^T A W_s = A_s$, где $s = 1, \dots, p$, являющиеся диагональными блоками матрицы A . Построим неполные обратные треугольные разложения для этих матриц: $H_s^T G_s \approx A_s^{-1}$. В качестве предобусловливателя будем использовать блочно-диагональную матрицу

$$M = \sum_{s=1}^p W_s H_s^T G_s W_s^T \approx (\text{BlockDiag}(A))^{-1}. \tag{7}$$

Предобусловливание (7) будем называть “блочное Якоби — неполное обратное треугольное разложение” (BJILU). Вычисление элементов матриц $G_s, H_s, s = 1, \dots, p$, осуществляется по алгоритму, описанному выше в разделе 3, где вместо матрицы A используются матрицы A_s . Заметим, что “точное” блочное предобусловливание Якоби имеет следующий вид: $M_{BJ} = \sum_{s=1}^p W_s A_s^{-1} W_s^T = (\text{BlockDiag}(A))^{-1}$, а предобусловливание BJILU0 может быть записано в форме

$$M_{BJILU0} = \sum_{s=1}^p W_s (I_s + U_s)^{-1} D_s (I_s + L_s)^{-1} W_s^T \approx (\text{BlockDiag}(A))^{-1}.$$

5. Сходимость итераций при ВJ-предобусловливании положительно определенных матриц. Пусть матрица A предварительно симметрично переупорядочена и отмасштабирована подходящим образом (см. выше). Напомним, что ВJ-предобусловливание получается при использовании обратной к матрице $B = \text{BlockDiag}(A)$, получаемой удалением всех блочно-внедиагональных элементов из матрицы A . Предположим, что матрица A является положительно определенной, т.е.

$$A_0 = \frac{A + A^T}{2} > 0.$$

Очевидно, что в этом случае положительно определенной является также и матрица B , т.е.

$$B_0 = \frac{B + B^T}{2} > 0.$$

Тогда, согласно основному результату работы [16], имеет место локализация спектра матрицы $B^{-1}A$, обеспечивающая сходимость предобусловленного метода GMRES, а тогда и метода BiCGStab [13]. В частности, для невязок $r_k = b - Ax_k$ показано, что если величина

$$\gamma = \frac{1}{4} \left\| A_0^{-1/2} (B - A) B_0^{-1/2} \right\|^2$$

удовлетворяет условию $0 < \gamma < 2.261$, то справедлива оценка

$$\frac{\|r_k\|}{\|r_0\|} \leq \text{const} \cdot \left(\frac{4\xi}{1 + 3\xi^2 + 5\xi^4} \right)^k,$$

где $\xi = (\gamma/(4 + 4\gamma))^{1/2}$. Видно, что оценка скорости сходимости имеет тенденцию к улучшению при уменьшении величины нормы $\|A - B\|$ блочно-внедиагональной части матрицы A . Таким образом, если матрица B достаточно мало отличается от A , можно ожидать, что ВJ-предобусловливание будет достаточно эффективным. Незначительное отличие матрицы B от A может иметь место при достаточно хорошем блочном разбиении матрицы A , когда вне блочной диагонали оказывается не слишком много ее ненулевых элементов и они сравнительно невелики по модулю. В настоящей работе используется разбиение [3], при котором количество ненулевых элементов вне блочной диагонали относительно невелико. Следует отметить, что замена точного обращения диагональных блоков, образующих матрицу B , на их приближения по алгоритму ИЛУ соответствует замене ВJ-предобусловливателя B на соответствующее приближение \tilde{B} и может привести к некоторому увеличению γ . Тем не менее, при достаточно хорошем качестве таких приближений можно рассчитывать на сходимость итераций метода BiCGStab в случае положительной определенности матрицы A и сохранения этого свойства для \tilde{B} . Заметим, что если диагональные элементы положительно определенной матрицы A недостаточно велики относительно внедиагональных элементов, то возможны случаи, когда ИЛУ-предобусловленная матрица $M = GAH^T$ уже не является положительно определенной и непосредственное использование результатов [16] невозможно. Так, в работе [17] приведен пример параметризованной матрицы размера 3×3 , где $A = \text{tridiag}(-a, 1, a)$, являющейся положительно определенной, однако при этом матрица $M = GAH^T$ будет положительно определенной только в случае, если в матрице A есть достаточно выраженное диагональное преобладание. С другой стороны, невыполнение достаточных условий, использованных в [16], еще не означает, что применение рассматриваемого предобусловливания сильно ухудшило свойства матрицы. В примере из [17], для предобусловленной матрицы всегда получаем вещественный спектр положительных собственных значений, расположенный на положительной полуоси, что можно считать улучшением (по сравнению с комплексным спектром исходной матрицы) с точки зрения использования итерационных методов.

6. Алгоритм параллельной реализации. При построении предобусловливателя в каждом процессоре с номером $s = 1, \dots, p$ будем строить матрицы G_s и H_s , содержащие соответствующие n_s строк матриц G и H . Сначала в каждом процессоре создадим матрицы $A_s = W_s^T A W_s$ размера $n_s \times n_s$, которые будут содержать все элементы матрицы A , необходимые для построения всех строк матриц G_s и H_s . При вычислении этих матриц в каждом процессоре не требуется информации, хранящейся в других процессорах, и поэтому их вычисление происходит одновременно во всех процессорах. Кроме того, каждая строка матриц G_s и H_s вычисляется независимо от других строк этих матриц с использованием OpenMP-технологии. Для распределения вычисления значений элементов матриц G_s и H_s по строкам используются простые циклы с опцией **schedule (static)**, задающей, каким образом итерации цикла распределяются по нитям. Заметим, что, как показали расчеты тестовых задач, использование опций

schedule (dynamic) или **schedule (guided)** не привело к существенному ускорению расчетов. Затем во всех процессорах одновременно формируются матрицы H_s^T . Как показывают расчеты задач, приведенные в [4], построение транспонированной матрицы занимает ничтожно малое время по сравнению с временем вычисления предобусловливателя и выполнением итерационного процесса.

При параллельной реализации вычислений типа $z = H^T Gr$ все процессоры выполняют работу одновременно и независимо, пересылок не требуется. При этом в каждом процессоре с номером s вычисляются $z_s = H_s^T G_s r_s$ в 2 этапа: $u_s = G_s r_s$, $z_s = H_s^T u_s$. Для распределения вычисления значений элементов векторов u_s , z_s по строкам i используется директива **do** с опцией **schedule static**. Матрица A хранится в памяти в стандартном CRS-формате (Compressed Row Storage) [15]. Параллельная реализация умножения матрицы на вектор в этом случае хорошо известна. Остальные этапы параллельной реализации предобусловленного метода BiCGStab тоже хорошо известны и не представляют труда. Заметим, что алгоритм параллельной реализации метода BJILU-BiCGStab аналогичен алгоритму параллельной реализации ВЈІІС-СG [5].

Заметим, что при параллельной реализации построения и обращения матрицы предобусловливания метода BJILU0 в настоящей работе использовался только MPI-подход. На этих этапах вычислений не требуется обмена информацией между процессорами. При умножении матрицы на вектор, вычислении скалярных произведений и выполнении векторных операции в алгоритме BJILU0-BiCGStab используются OpenMP+MPI-подходы.

7. Результаты расчетов. Все программы, реализующие использование методов BJILU-BiCGStab, J-BiCGStab и BJILU0-BiCGStab для решения системы уравнений $Ax = b$, были написаны на языке FORTRAN с использованием MPI+OpenMP, расчеты производились на многопроцессорном вычислительном кластере K60, установленном в ИПМ им. М.В. Келдыша РАН. Для тестирования рассматриваемых параллельных методов использовался набор матриц из коллекции разреженных матриц SuiteSparse (ранее коллекция университета Флориды). Перечислим имена используемых тестовых матриц и укажем источник их происхождения:

- tmt_unsym** — моделирование электромагнитной задачи;
- xenon2** — моделирование упругих свойств сложного кристаллического соединения;
- sme3dc** — моделирование трехмерной конструкции;
- sme3db** — моделирование трехмерной конструкции;
- ecl32** — моделирование полупроводникового устройства;
- raefsky3** — вычислительная гидродинамика, моделирование обтекания и температуры в условиях турбулентности.

В табл. 1 приведены некоторые свойства этих матриц: n — порядок матрицы, $nz(A)$ — число ее ненулевых элементов, nz_{\min} и nz_{\max} — минимальное и максимальное число ненулевых элементов в строке матрицы A соответственно. Решалось уравнение (1) с правой частью $b = [1 \dots 1]^T$. Начальное приближение x_0 выбиралось равным нулю, итерации производились до выполнения условия (2), где $\epsilon = 10^{-8}$. Для разбиения области расчета использовался метод [3] с 4 итерациями.

Таблица 1
Структурные свойства набора тестовых матриц

Матрица	n	$nz(A)$	nz_{\min}	nz_{\max}
tmt_unsym	917825	4584801	3	5
xenon2	157464	3866688	1	27
sme3dc	42930	3148656	24	405
sme3db	29067	2081063	24	345
ecl32	51993	380415	1	23
raefsky3	21200	1488769	24	80

Таблица 2
Число итераций и время счета задачи tmt_unsym разными методами на p процессорах с использованием MPI или MPI+OpenMP

$p =$	8	16	32	64	96
Jacobi	12224, 58.92 8832, 23.25(4)	12282, 29.57 7904, 19.65(8)	13060, 14.07 9400, 12.45(8)	13060, 7.07 11240, 8.15(4)	13608, 5.39 8774, 5.07(4)
BJILU	6100, 48.98 4476, 19.29(4)	4978, 20.44 4696, 17.76(8)	5070, 12.04 5252, 13.40(4)	4944, 4.89 4706, 5.32(4)	5484, 3.73 5080, 4.33(4)
BJILU0	3198, 20.74(4)	3260, 15.90(8)	3094, 6.65(4)	3214, 3.49(4)	4178, 3.13(4)

В табл. 2–7 приведены числа итераций и времена счета задач на p процессорах (в секундах) методом J-BiCGStab, состоящие из времени счета итерационного процесса, и методами BJILU-BiCGStab и BJILU0-

BiCGStab, состоящие из времени счета предобусловливателя и времени счета итерационного процесса. В скобках приведено число нитей, при котором время счета было минимальным. В первой и третьей строках приведены результаты расчетов при использовании только MPI.

При использовании метода BJILU–BiCGStab или BJILU0–BiCGStab может наблюдаться некоторый (не всегда монотонный) рост числа итераций с ростом числа процессоров. Немонотонность роста числа итераций, возможно, связана с особенностями разбиения области. Кроме того, наблюдалось изменение числа итераций при использовании нитей и от запуска к запуску, что объясняется погрешностями вычислений, возможно, из-за разного порядка слагаемых при вычислении скалярных произведений.

Таблица 3

Число итераций и время счета задачи xepo2 разными методами на p процессорах с использованием MPI или MPI+OpenMP

$p =$	8	16	32	64	96
Jacobi	2218, 4.05 1898, 1.90(4)	2304, 2.29 1790, 1.96(4)	2562, 1.47 1852, 1.31(4)	2336, 0.722 2042, 0.853(4)	2400, 0.57 1986, 0.65(4)
BJILU	720, 3.13 628, 1.50(4)	778, 1.8 652, 1.57(8)	754, 0.977 720, 0.976(4)	730, 0.461 732, 0.523(4)	766, 0.34 744, 0.39(4)
BJILU0	> 20000(4)	> 20000(8)	> 20000(4)	> 20000(4)	> 20000(4)

Таблица 4

Число итераций и время счета задачи sme3dc разными методами на p процессорах с использованием MPI или MPI+OpenMP

$p =$	8	15	19	36
Jacobi	18796, 27.95 10336, 8.56(8)	14812, 11.83 12496, 10.93(4)	10440, 7.66 12530, 10.46(4)	18580, 9.33 8354, 6.15(8)
BJILU	5488, 19.94 3576, 6.16(8)	4874, 9.32 3800, 7.39(16)	4494, 7.88 3816, 6.80(4)	5948, 6.5 3468, 4.81(8)
BJILU0	3270, 8.18(8)	3600, 9.53(16)	4136, 6.52(4)	3758, 4.23(8)

Таблица 5

Число итераций и время счета задачи sme3db разными методами на p процессорах с использованием MPI или MPI+OpenMP

$p =$	8	14 или 15	19	36
Jacobi	14974, 15.54 8576, 5.1(8)	7080, 5.57 5678, 2.75(4)	7674, 4.46 6018, 4.01(4)	14038, 5.44 6360, 3.62(8)
BJILU	2398, 7.50 2182, 3.77(16)	4320, 8.74 2530, 3.26(4)	3398, 5.17 2654, 4.01(4)	3692, 3.23 3408, 3.36(4)
BJILU0	2420, 4.41(16)	2246, 4.17(4)	2556, 3.29(4)	>20000(4)

Как видно из табл. 2–4, использование предобусловливания BJILU позволяет существенно уменьшить число итераций по сравнению с расчетами с предобусловливателем Jacobi. С другой стороны, обращение предобусловливателя BJILU на каждой итерации, а в ряде случаев и его построение, занимает немало времени, что может увеличить время счета. Как видно из таблиц и рисунков, при использовании только MPI метод BiCGStab с предобусловливанием BJILU в большинстве случаев считает быстрее, чем с предобусловливанием Якоби. Исключение составляет задача с матрицей es132.

На рис. 1–6 приведены графики (в логарифмическом масштабе) зависимости времени счета рассматриваемых задач методами J-BiCGStab, BJILU-BiCGStab и BJILU0-BiCGStab от числа процессоров с применением только MPI (сплошные линии) и MPI+OpenMP-подходов (пунктирные линии).

Как видно из табл. 2–5, 7 и рис. 1–4, 6, использование OpenMP-технологии при решении задач с матрицами tmt_unsym, xepo2, raefsky3 методом BJILU-BiCGStab приводит к ускорению счета на K60 при использовании менее 32 процессоров, в случае решения задачи с матрицей sme3dc — при использовании

Таблица 6
Число итераций и время счета задачи esl32 разными методами на p процессорах с использованием MPI или MPI+OpenMP

$p =$	8	16	32	64
Jacobi	606, 0.26 604, 0.17(4)	594, 0.15 602, 0.24(4)	594, 0.16 602, 0.14(4)	592, 0.11 592, 0.11(4)
BJILU	406, 0.34 356, 0.19(4)	446, 0.21 366, 0.23(4)	424, 0.19 362, 0.12(4)	466, 0.15 402, 0.10(4)
BJILU0	280, 0.22(4)	318, 0.20(4)	314, 0.12(4)	336, 0.08(4)

Таблица 7
Число итераций и время счета задачи gaefsky3 разными методами на p процессорах с использованием MPI или MPI+OpenMP

$p =$	8	16	32	64
Jacobi	11042, 7.23 10926, 3.85(4)	11046, 3.92 11226, 4.90(4)	10986, 2.22 11110, 3.04(4)	11078, 1.45 11056, 2.22(4)
BJILU	3432, 5.39 3168, 2.33(4)	3110, 2.57 3004, 2.40(4)	3032, 1.34 2962, 1.44(4)	3592, 0.93 3444, 1.10(4)
BJILU0	834, 1.10(4)	994, 0.93(4)	1436, 0.65(4)	2100, 0.59(4)

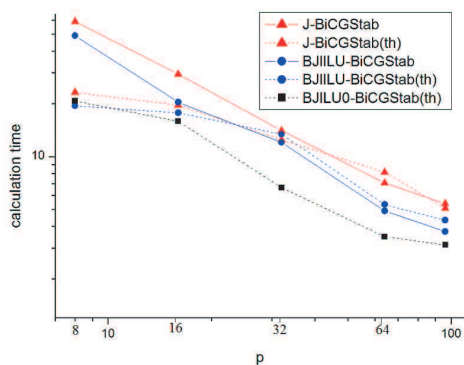


Рис. 1. Времена счета задачи с матрицей tmt_unsum

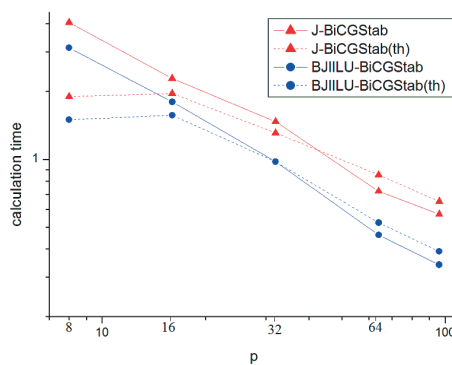


Рис. 2. Времена счета задачи с матрицей xepom2

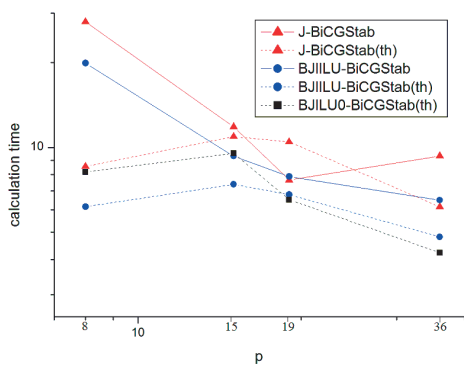


Рис. 3. Времена счета задачи с матрицей sme3dc

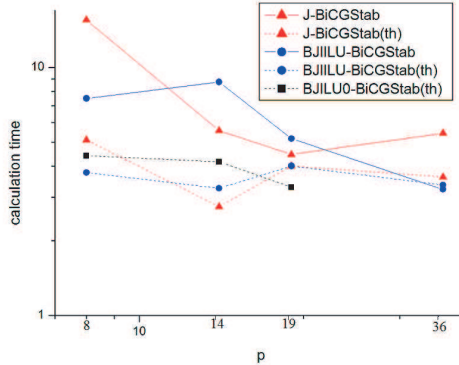


Рис. 4. Времена счета задачи с матрицей sme3db

до 36 процессоров, а с матрицей *sme3db* — менее 36 процессоров. Как видно из табл. 6 и рис. 5, использование OpenMP-технологии при решении задачи с матрицей *esl32* методом *BJILU-BiCGStab* приводит к ускорению счета на K60 при использовании любого рассмотренного числа процессоров кроме $p = 16$.

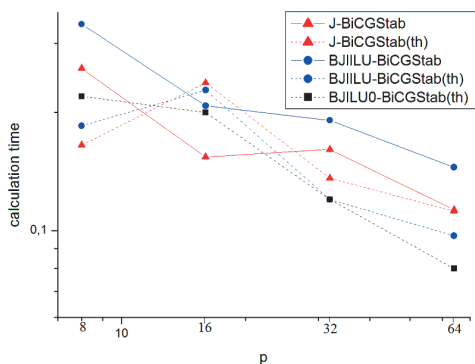


Рис. 5. Времена счета задачи с матрицей *esl32*

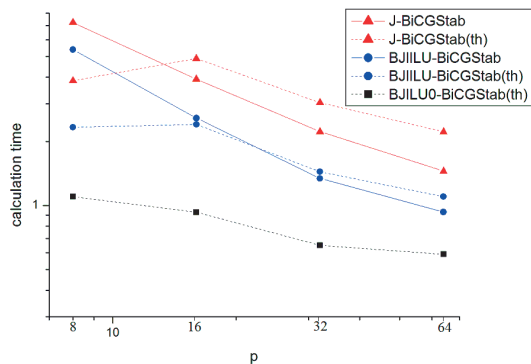


Рис. 6. Времена счета задачи с матрицей *raefsky3*

Уменьшение эффекта от использования OpenMP-технологии с увеличением числа процессоров (см. рис. 1, 2, 6) объясняется уменьшением числа строк матрицы, приходящихся на каждый процессор. Заметим, что в настоящей работе в качестве тестовых матриц использовались матрицы относительно небольшого размера. При расчетах реальных физических задач размеры матриц, как правило, значительно больше. Следует ожидать, что для таких задач потеря эффективности от использования OpenMP-технологии наступит при значительно большем числе процессоров.

Как видно из рис. 1–6, в подавляющем большинстве случаев решение рассмотренных тестовых задач методом *BJILU-BiCGStab* с использованием MPI+OpenMP-подхода на многопроцессорном вычислительном кластере K60 происходило быстрее, чем методом *J-BiCGStab*. На рис. 7 представлены графики зависимости ускорения счета по сравнению со счетом на 8 процессорах с использованием только MPI для метода *BJILU-BiCGStab* при решении некоторых указанных выше достаточно больших задач. Сплошная линия соответствует расчетам с использованием только MPI, штриховая — MPI+OpenMP. Как видно из рис. 7, наблюдается хорошее ускорение счета с ростом числа процессоров для их разумного числа. При не слишком большом числе процессоров использование MPI+OpenMP-подхода при расчетах указанных задач наблюдается сверхлинейное ускорение. В ряде случаев сверхлинейное ускорение получено при использовании только MPI, что связано с изменением числа полученных в расчете итераций и, возможно, с особенностями разбиения на подобласти (так как сравнение происходит с временем счета на 8 процессорах).

Обсудим теперь результаты использования предобуславливания *BJILU0*. Таблица 3 показывает, что при решении задачи с матрицей *xenon2* методом *BJILU0-BiCGStab* на основе MPI+OpenMP-подхода был достигнут предел числа итераций, причем на 20000-й итерации точность была около 0.01. Аналогичный пример неудачного применения *BJILU0-BiCGStab* демонстрирует табл. 5, причем для наибольшего числа процессоров. Произведем сравнение времени счета остальных тестовых задач методом *BJILU0-BiCGStab* с использованием MPI+OpenMP-подхода и методом *BJILU-BiCGStab* с использованием MPI+OpenMP-подхода.

Как видно из табл. 2, 4–6 и рис. 1, 3–5, решение задач с матрицами *sme3db* и *sme3dc* методом *BJILU0-BiCGStab* происходило медленнее, чем методом *BJILU-BiCGStab* при $p \leq 15$, решение задач

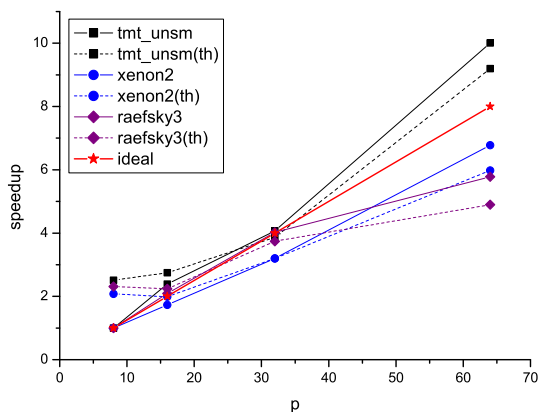


Рис. 7. Ускорение счета для задач большего размера

с матрицами `esl32`, `tmt_unsm` методом `VJILU0-BiCGStab` происходило медленнее, чем методом `VJILU-BiCGStab` при $p \leq 32$ и при $p \leq 8$ соответственно. Заметим, что число итераций при решении задачи с матрицей `tmt_unsm` методом `VJILU0-BiCGStab` примерно в 1.5 раза меньше, чем методом `VJILU-BiCGStab`. Как видно из табл. 7 и рис. 6, решение задачи с матрицей `raefsky3` методом `VJILU0-BiCGStab` происходило быстрее, чем методом `VJILU-BiCGStab` при всех использованных значениях p . Заметим, что число итераций метода `VJILU0-BiCGStab` при решении этой задачи сильно растет с ростом числа процессоров, но значительно меньше, чем число итераций метода `VJILU-BiCGStab`. Заметим, что число итераций при решении задач с матрицами `sme3db`, `sme3dc`, `esl32` этими двумя методами отличается несильно.

Что касается времени счета, лишь для одной из пяти задач предложенный метод сильно проигрывает методу `VJILU0-BiCGStab`. Для задач с сопоставимым числом итераций для небольшого числа процессоров, наоборот, наблюдается выигрыш времени счета предложенным методом по сравнению с методом `VJILU0-BiCGStab`. В отношении надежности метод `VJILU0-BiCGStab` явно проигрывает обоим методам `VJILU-BiCGStab` и `J-BiCGStab`, так как в двух из шести задач этот метод не обеспечивает сходимость итераций. Это объясняется тем, что предобусловливание `ILU0` не гарантирует корректность вычислений при отсутствии достаточного диагонального преобладания матрицы A , в отличие от предобусловливаний `ILU` или Якоби, для применимости которых достаточно положительной определенности A .

8. Заключение. В настоящей статье рассмотрен метод предобусловливания `ILU` и предложен новый метод предобусловливания `VJILU`. Кроме того, предложен способ параллельной реализации метода `VJILU-BiCGStab` с использованием `MPI+OpenMP`-подхода. Расчеты ряда тестовых задач из коллекции университета Флориды показали, что использование `OpenMP`-технологии при решении задач методом `VJILU-BiCGStab` позволяет ускорить вычисления для не слишком большого числа процессоров.

Авторы благодарны рецензенту за полезные замечания, позволившие существенно улучшить изложение материала статьи.

СПИСОК ЛИТЕРАТУРЫ

1. Капорин И.Е. О предобусловливании метода сопряженных градиентов при решении дискретных аналогов дифференциальных задач // Дифференциальные уравнения. 1990. **26**, № 7. 1225–1236.
2. Kaporin I.E. New convergence results and preconditioning strategies for the conjugate gradient method // Numer. Linear Algebra Appl. 1994. **1**, N 2. 179–210.
3. Капорин И.Е., Милюкова О.Ю. Неполное обратное треугольное разложение в параллельных алгоритмах предобусловленного метода сопряженных градиентов. Препринт № 037 ИПМ им. М.В. Келдыша. М., 2017.
4. Капорин И.Е., Милюкова О.Ю. `MPI+OpenMP` параллельная реализация метода сопряженных градиентов с некоторыми явными предобусловливателями. Препринт № 008 ИПМ им. М.В. Келдыша. М., 2018.
5. Капорин И.Е., Милюкова О.Ю. `MPI+OpenMP` реализация метода сопряженных градиентов с факторизованными явными предобусловливателями // Вопросы атомной науки и техники. Серия математическое моделирование физических процессов. 2018. Вып. 4. 57–69.
6. Davis T.A., Hu Y. The university of Florida sparse matrix collection // ACM Trans. on Math. Software. 2011. **38**, N 1. doi 10.1145/2049662.2049663.
7. Капорин И.Е., Милюкова О.Ю. Массивно-параллельный алгоритм предобусловленного метода сопряженных градиентов для численного решения систем линейных алгебраических уравнений // Сб. Трудов отдела проблем прикладной оптимизации ВЦ РАН. М.: Изд-во ВЦ РАН, 2011. 132–157.
8. Knight P.A. The Sinkhorn–Knopp algorithm: convergence and applications // SIAM J. Matrix Anal. Appl. 2008. **30**, N 1. 261–275.
9. Капорин И.Е. Итерационное решение систем линейных уравнений с использованием неполной обратной треугольной факторизации // Прямые и обратные задачи математической физики. М.: Изд-во МГУ, 1991. 71–77.
10. Kershaw D.S. The incomplete Cholesky-conjugate gradient method for the iterative solution of systems of linear equations // Journal of Computational Physics. 1978. **26**, N 1. 43–65.
11. Anzt H., Huckle T.K., Brückle J., Dongarra J. Incomplete sparse approximate inverses for parallel preconditioning // Parallel Computing. 2018. **71**. 1–22.
12. Van der Vorst H.A. Bi-CGStab: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems // SIAM J. Sci. Stat. Comput. 1992. **13**, N 2. 631–644.
13. Chan T.F., Szeto T. Composite step product methods for solving nonsymmetric linear systems // SIAM Journal on Scientific Computing. 1996. **17**, N 6. 1491–1508.
14. Saad Y., Schultz M.H. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems // SIAM Journal on Scientific and Statistical Computing. 1986. **7**, N 3. 856–869.
15. Джорджс А., Лю Джс. Численное решение больших разреженных систем уравнений. М.: Мир, 1984.

16. Капорин И.Е. Локализация собственных значений пучка положительно-определенных матриц // Ж. вычисл. матем. и матем. физ. 2008. **48**, № 11. 1923–1931.
17. Капорин И.Е., Милукова О.Ю. MPI+OpenMPI реализация метода BiCGStab с факторизованным явным преобусловливателем. Препринт № 047 ИПМ им. М.В. Келдыша. М., 2019.

Поступила в редакцию
10.07.2019

MPI+OpenMP Implementation of the BiCGStab Method with Explicit Preconditioning for the Numerical Solution of Sparse Linear Systems

I. E. Kaporin¹ and O. Yu. Milyukova²

¹ *Dorodnitsyn Computing Center, Russian Academy of Sciences; ulitsa Vavilova 40, Moscow, 119333, Russia; Dr. Sci., Principal Scientist, e-mail: igorkaporin@mail.ru*

² *Keldysh Institute of Applied Mathematics, Russian Academy of Sciences; Miusskaya ploshchad' 4, Moscow, 125047, Russia; Dr. Sci., Leading Scientist, e-mail: olgamilyukova@mail.ru*

Received July 10, 2019

Abstract: A preconditioner for a large sparse nonsymmetric positive definite matrix is considered on the basis of its approximate inverse in the form of product of a lower triangular sparse matrix by an upper triangular matrix. For the class of matrices being considered, a new preconditioning based on the approximate block Jacobi with incomplete inverse LU-factorization preconditioning is proposed. For a parallel implementation of the corresponding preconditioned BiCGStab algorithm, the MPI+OpenMP techniques are used. The timing results obtained for the MPI+OpenMP and MPI implementations of the proposed preconditioning and for the Jacobi preconditioning with the BiCGStab are compared using several test problems from the SuiteSparse collection (formerly known as the University of Florida sparse matrix collection).

Keywords: iterative solution of linear systems, sparse matrices, incomplete inverse triangular factorization, parallel preconditioning, BiCGStab method.

References

1. I. E. Kaporin, "A Preconditioned Conjugate-Gradient Method for Solving Discrete Analogs of Differential Problems," *Differ. Uravn.* **26** (7), 1225–1236 (1990) [*Differ. Equ.* **26** (12), 897–906 (1990)].
2. I. E. Kaporin, "New Convergence Results and Preconditioning Strategies for the Conjugate Gradient Method," *Numer. Linear Algebra Appl.* **1** (2), 179–210 (1994).
3. I. E. Kaporin and O. Yu. Milyukova, *Incomplete Inverse Triangular Factorization in Parallel Algorithms of Preconditioned Conjugate Gradient Methods*, Preprint No. 037 (Keldysh Institute of Applied Mathematics, Moscow, 2017).
4. I. E. Kaporin and O. Yu. Milyukova, *MPI+OpenMP Parallel Implementation of Explicitly Preconditioned Conjugate Gradient Method*, Preprint No. 008 (Keldysh Institute of Applied Mathematics, Moscow, 2018).
5. I. E. Kaporin and O. Yu. Milyukova, "MPI+OpenMP Parallel Implementation of the Conjugate Gradient Method with Factorized Explicit Preconditioners," *Voprosy Atomn. Nauki Tekhniki, Ser.: Mat. Mod. Fiz. Proc.*, No. 4, 57–69 (2018).
6. T. A. Davis and Y. Hu, "The University of Florida Sparse Matrix Collection," *ACM Trans. Math. Softw.* **38** (2011). doi 10.1145/2049662.2049663
7. I. E. Kaporin and O. Yu. Milyukova, "Massively Parallel Preconditioned Conjugate Gradient Algorithm for the Numerical Solution of Linear Algebraic Equations," in *Optimization and Applications* (Comput. Center Ross. Akad. Nauk, Moscow, 2011), Issue 2, pp. 132–157.
8. P. A. Knight, "The Sinkhorn–Knopp Algorithm: Convergence and Applications," *SIAM J. Matrix Anal. Appl.* **30** (1), 261–275 (2008).
9. I. E. Kaporin, "Iterative Solution of Linear Systems Using the Incomplete Inverse Triangular Decomposition," in *Direct and Inverse Problems of Mathematical Physics* (Mosk. Gos. Univ., Moscow, 1991), pp. 71–77.
10. D. S. Kershaw, "The Incomplete Cholesky-Conjugate Gradient Method for the Iterative Solution of Systems of Linear Equations," *J. Comput. Phys.* **26** (1), 43–65 (1978).

11. H. Anzt, T. K. Huckle, J. Bräckle, and J. Dongarra, “Incomplete Sparse Approximate Inverses for Parallel Preconditioning,” *Parallel Comput.* **71**, 1–22 (2018).
12. H. A. van der Vorst, “Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems,” *SIAM J. Sci. Stat. Comput.* **13** (2), 631–644 (1992).
13. T. F. Chan and T. Szeto, “Composite Step Product Methods for Solving Nonsymmetric Linear Systems,” *SIAM J. Sci. Comput.* **17** (6), 1491–1508 (1996).
14. Y. Saad and M. H. Schultz, “GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems,” *SIAM J. Sci. Stat. Comput.* **7** (3), 856–869 (1986).
15. A. George and J. W. H. Liu, *Computer Solution of Large Sparse Positive Definite Systems* (Prentice-Hall, Englewood Cliffs, 1981).
16. I. E. Kaporin, “Localization of the Eigenvalues of a Pencil of Positive Definite Matrices,” *Zh. Vychisl. Mat. Mat. Fiz.* **48** (11), 1923–1931 (2008) [*Comput. Math. Math. Phys.* **48** (11), 1917–1926 (2008)].
17. I. E. Kaporin and O. Yu. Milyukova, *MPI+OpenMP Implementation of BiCGStab Method with Factorized Explicit Preconditioner*, Preprint No. 047 (Keldysh Institute of Applied Mathematics, Moscow, 2019).