

В. П. Ильин, Г. Ю. Казанцев

ИТЕРАЦИОННОЕ РЕШЕНИЕ СЛАУ С СЕДЛОВОЙ ТОЧКОЙ

§1. ВВЕДЕНИЕ

Рассматривается проблема о решении систем линейных алгебраических уравнений (СЛАУ) вида

$$Ku \equiv \begin{bmatrix} A & B^T \\ B & -C \end{bmatrix} \begin{bmatrix} u_a \\ u_c \end{bmatrix} = \begin{bmatrix} f_a \\ f_c \end{bmatrix} \equiv f, \quad (1)$$

где A, C – симметричные положительно определенная (с.п.о.) и неотрицательно определенная квадратные матрицы порядков N_1 и N_2 соответственно, $K \in \mathcal{R}^{N,N}$, $N = N_1 + N_2$. Матрица K является невырожденной, если C есть с.п.о.-матрица на ядре B^T , и вырожденной, когда C и B^T имеют общее нуль-пространство. Вопросы данного типа актуальны во многих электрофизических, гидро-газодинамических и других приложениях, особенно, возникающих при аппроксимации многомерных смешанных постановок для краевых задач методами конечных разностей, конечных объемов, конечных элементов и разрывными алгоритмами Галеркина, см. [1]. В частности, самостоятельный интерес представляют СЛАУ вида (1) при $C = 0$, называемыми системами с седловой точкой, имеющими происхождение из задач оптимизации.

Более конкретно, в качестве системы (1) мы рассматриваем СЛАУ, возникающие из конечно-элементной аппроксимации двумерной или трехмерной задачи Неймана для двухфазной фильтрации в смешанной постановке с помощью базисных функций Равьяра–Тома на регулярной прямоугольной сетке в соответствии с работой [2]. При этом матрица A является трехдиагональной, а матрица B^T имеет одномерное ядро с вектором $e = \{1\}$, состоящим из единичных компонент.

Ключевые слова: уравнения фильтрации, базис Равьяра–Тома, седловая задача, алгебраическая система, итерационный метод Узавы, предобуславливающая матрица, подпространства Крылова, численные эксперименты.

Работа поддержана грантом РФФИ No. 18-01-00295 (решение абстрактных СЛАУ с седловой точкой) и грантом РНФ No. 19-11-00048 (приближения к задачам фильтрации в смешанной постановке).

Матрица C может быть нулевой или иметь специальный вид (см. ниже). Если СЛАУ (1) является вырожденной, то вектор правой части f предполагается ортогональным ядру для обеспечения совместности алгебраической системы.

Методы решения СЛАУ с седловой точкой представляют содержательную часть вычислительной алгебры и имеют достаточно большую историю. Анализ литературы по данным вопросам можно найти в обзорной статье Дм. Голуба с коллегами [3], в монографии [4], а также в недавних работах [5]–[8]. Основные подходы здесь заключаются в построении легко обрабатываемой предобуславливающей матрицы M , обеспечивающей значительное улучшение обусловленности произведения $M^{-1}K$ по сравнению с исходной матрицей K , а также в формировании итерационных процессов в подпространствах Крылова

$$\mathcal{K}_{n+1}(K, M, r^0) = \text{Span}\{r^0, M^{-1}r^0, \dots, (M^{-1})^n r^0\}, \quad r^0 = f - Ku^0.$$

При реализации данных алгоритмов на гетерогенных многопроцессорных вычислительных системах (МВС) с архитектурой кластерного типа большое практическое значение имеет достижение высокой производительности компьютерного эксперимента, включая масштабируемое распараллеливание методов и их отображение на компьютерные архитектуры с распределенной и иерархической общей памятью. Для этого, в общем случае, используются средства гибридного программирования с инструментами передачи межузловых сообщений и многопоточковых вычислений на многоядерных процессорных устройствах CPU (системы MPI и OPENMP соответственно), а также векторизации операций на основе системы команд типа AVX и применения сверхбыстрых графических ускорителей (GPGPU или Intel Phi). Главная цель здесь заключается в сокращении потерь от коммуникационных операций, которые являются не только самыми медленными, но и экстремально энергозатратными. В этих вопросах важное значение имеют структуры данных для разреженных матриц высоких порядков (до 10^{10} и более), которые возникают в актуальных приложениях. Эти и другие технологические проблемы исследовались в многочисленных работах по программному обеспечению для задач вычислительной алгебры, см. обзоры в [9], [10].

Содержание данной работы заключается в сравнительном экспериментальном исследовании некоторых алгоритмов для решения трехмерных задач фильтрации с модельными исходными данными. В §2 мы

рассматриваем ряд современных предобусловленных блочных методов для решения седловых задач. §3 посвящен алгоритмическим особенностям решения рассматриваемых краевых задач, включая вопросы эффективности распараллеливания итерационных процессов. §4 содержит результаты численных экспериментов, демонстрирующих эффективность предлагаемых методов. В заключении обсуждаются практические выводы по перспективам повышения производительности алгоритмов для решения седловых задач и возможные направления дальнейших исследований.

§2. БЛОЧНЫЕ ПРЕДОБУСЛОВЛЕННЫЕ МЕТОДЫ В ПОДПРОСТРАНСТВАХ КРЫЛОВА

С использованием дополнения Шура

$$S = C + BA^{-1}B^T \quad (2)$$

матрица системы (1) факторизуется в виде

$$K = \begin{bmatrix} A & 0 \\ B & -S \end{bmatrix} \begin{bmatrix} I & A^{-1}B^{-1} \\ 0 & I \end{bmatrix}. \quad (3)$$

Если матрицы A, S в (3) заменить их приближениями (предобуславливателями) M_a и M_s , то получаем предобуславливатель матрицы K в виде неполной блочной факторизации

$$M_1 = \begin{bmatrix} M_a & 0 \\ B & -M_s \end{bmatrix} \begin{bmatrix} I & M_a^{-1}B^T \\ 0 & I \end{bmatrix}. \quad (4)$$

Несколько более грубая аппроксимация, получаемая при использовании в (4) только одного (левого или правого) множителя, приводит к неполному блочно-треугольному предобуславлианию с матрицей

$$M_2 = \begin{bmatrix} M_a & B^T \\ 0 & -M_s \end{bmatrix} \quad (5)$$

или к неполному предобуславливателю Узавы

$$M_3 = \begin{bmatrix} M_a & 0 \\ B & -M_s \end{bmatrix}. \quad (6)$$

Реализация каждого шага соответствующих итерационных процессов может быть представлена несколькими стадиями, на которых пересчитывается только одна блочная компонента искомого решения (поэтому эти методы называются иногда сегрегационными). В несколько обобщенной форме такой стационарный алгоритм (пока без крыловского ускорения) представляется следующими тремя стадиями, см. [5]:

$$\begin{aligned}\hat{u}_a^{n+1} &= u_a^n + Q_a^{(1)}(f_a - Au_a^n - B^T u_c^n), \\ u_c^{n+1} &= u_c^n - M_s^{-1}(f_c - B\hat{u}_a^{n+1} + Cu_c^n), \\ u_a^{n+1} &= \hat{u}_a^{n+1} + Q_a^{(2)}(f_a - A\hat{u}_a^{n+1} - B^T u_c^{n+1}).\end{aligned}\quad (7)$$

Здесь $Q_a^{(1)}, Q_a^{(2)}$ – некоторые аппроксимации обратной или обобщенной обратной к предобуславителю M_a . В частности, если $Q_a^{(1)} = M_a^{-1}$, $Q_a^{(2)} = 0$ или $Q_a^{(1)} = 0$, $Q_a^{(2)} = M_a^{-1}$, то из (7) мы получаем или алгоритм Узавы с предобуславлителем M_3 из (6) (при этом третья стадия опускается, т.е. $u_a^{n+1} = \hat{u}_a^{n+1}$), или неполное блочно-треугольное предобуславливание с матрицей M_2 из (5) (при этом первая стадия в (7) опускается и $u_a^{n+1} = u_a^n$).

Если матрица $Q_a = Q_a^{(1)} + Q_a^{(2)} - Q_a^{(2)} A Q_a^{(1)}$ является невырожденной, то итерационному процессу (7) соответствует предобуславливатель

$$M = \begin{pmatrix} I & 0 \\ BQ_a^{(1)} & I \end{pmatrix} \begin{pmatrix} Q_a^{-1} & 0 \\ 0 & -M_s \end{pmatrix} \begin{pmatrix} I & Q_a^{(2)} B^T \\ 0 & I \end{pmatrix}. \quad (8)$$

В частном случае $Q_a^{(1)} = Q_a^{(2)} = M_a^{-1}$, из (8) следует так называемый симметризованный неполный метод Узавы с предобуславливающей матрицей

$$M = \begin{pmatrix} I & 0 \\ BM_a^{-1} & I \end{pmatrix} \begin{pmatrix} M_a(2M_a - A)^{-1}M_a & 0 \\ 0 & -M_s \end{pmatrix} \begin{pmatrix} I & M_a^{-1}B^T \\ 0 & I \end{pmatrix}. \quad (9)$$

Рассмотрим теперь случай, когда матрицы B^T и C имеют общее нуль-пространство. Пусть $V \in \mathcal{R}^{N_2, N_k}$, $N_k \leq N_2$, – прямоугольная матрица, N_k столбцов которой составляют ортогональный базис ядра $\mathcal{N}(B^T)$. При этом для совместимости СЛАУ (1) подвектор f_c должен быть ортогонален данному ядру (т.е. $V^T f_c = 0$), а матрица C будет положительно определенной только на множестве $\mathcal{N}(B^T) \setminus \mathcal{R}(V)$.

Для данной ситуации рассмотрим итерационный процесс вида (7), в котором C и M_s^{-1} заменены соответственно на

$$\bar{C} = C + VV^T, \quad \bar{M}_s^{-1} = (I - VV^T) + VV^T, \quad (10)$$

а для получаемых при этом последовательных приближений \bar{u}_a^n , \bar{u}_c^n начальные значения выбираются в виде

$$\bar{u}_a^0 = u_a^0, \quad \bar{u}_c^0 = (I - VV^T)u_c^0. \quad (11)$$

Соответствующий итерационный метод будем называть регуляризованным. При этом матрица

$$\bar{S} = \bar{C} + BA^{-1}B^T = S + VV^T \quad (12)$$

будет положительно определенной, \bar{C} будет с.п.о.-матрица на нуль-пространстве B^T , а \bar{M}_s является с.п.о.-матрицей одновременно с M_s .

В силу симметрии матрицы C и выполнения соотношений

$$CV = 0, \quad V^TC = 0, \quad B^TV = 0, \quad V^TB = 0,$$

по индукции нетрудно показать, что для всех n

$$\bar{u}_a^n = u_a^n, \quad \bar{u}_c^n = (I - VV^T)\bar{u}_c^n.$$

Таким образом, в точной арифметике исходный итерационный алгоритм (7) и его регуляризованный вариант будут сходиться одновременно.

Содержательный класс итерационных алгоритмов решения СЛАУ седлового типа основан на введении предобусловленных матриц блочно-диагонального вида. В работе [7], в частности, показано, что при использовании предобуславливателя вида

$$M = \begin{bmatrix} A + B^TL^{-1}B & 0 \\ 0 & L \end{bmatrix}, \quad (13)$$

где $L \in \mathcal{R}^{N_2, N_2}$ – произвольная невырожденная матрица, спектр произведения $M^{-1}K$ содержит только два различных значения: $\lambda_+ = 1$, $\lambda_- = -1$, где первое из них имеет кратность N_1 , а второе – N_2 .

Отметим также, что для решения СЛАУ (1) при $C = 0$ можно предложить метод, достаточно близкий в алгоритмическом плане к выбору предобуславливателя вида (13). Он заключается в построении линейной комбинации блочных строк системы (1):

$$\begin{bmatrix} A + B^TL^{-1}B & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} u_a \\ u_c \end{bmatrix} = \begin{bmatrix} f_a + B^TL^{-1}f_c \\ f_c \end{bmatrix}, \quad (14)$$

где L , как и в (13), есть произвольная с.п.о.-матрица. Отсюда следует алгоритм, который можно назвать регуляризованным методом Узавы:

$$\begin{aligned}\bar{K}u_c &\equiv B\bar{A}^{-1}B^T u_c = B\bar{A}^{-1}(f_0 + B^T L^{-1}f_c) - f_c, \\ \bar{A} &= A + B^T L^{-1}B.\end{aligned}\quad (15)$$

В формулах (14) и (15) заложены широкие возможности конструирования конкретных алгоритмов за счет выбора матрицы L . Такой подход иногда называется градиентной регуляризацией, поскольку матрица B^T происходит, как правило, из аппроксимации оператора градиента, см. формулы (25) ниже.

Для случая симметричных исходной и предобуславливающей матриц K, M запишем итерационные методы сопряженных направлений для решения СЛАУ (1) следующим образом [11]:

$$\begin{aligned}r^0 &= f - Ku^0, \quad p^0 = M^{-1}r^0, \quad n = 0, 1, \dots : \\ u^{n+1} &= u^n + \alpha_n p^n, \quad \alpha_n = \sigma_n / \rho_n, \quad \sigma_n = (K^\gamma M^{-1}r^n, r^n), \\ r^{n+1} &= r^n - \alpha_n Kp^n, \quad \rho_n = (Kp^n, K^\gamma p^n), \\ p^{n+1} &= M^{-1}r^{n+1} + \beta_n p^n, \quad \beta_n = \sigma_{n+1} / \sigma_n.\end{aligned}\quad (16)$$

Здесь $\gamma = 0$ и $\gamma = 1$ соответствуют методам сопряженных градиентов и сопряженных невязок, которые минимизируют функционал

$$\Phi_n^{(\gamma)} = (K^{\gamma-1}r^n, r^n)$$

в подпространствах Крылова

$$\mathcal{K}_{n+1}(K, M, r^0) = \text{Span}\{p^0, Kp^0, \dots, K^n p^0\}.$$

§3. ОСОБЕННОСТИ РЕАЛИЗАЦИИ АЛГОРИТМОВ НА ПРЯМОУГОЛЬНОЙ СЕТКЕ

Рассмотрим сначала более подробно структуры матриц и особенности реализации алгоритмов для случая $C = 0$ в (1) и прямоугольной расчетной области. При использовании равномерной сетки с шагами h_x, h_y

$$x_i = x_0 + ih_x, \quad i = 1, 2, \dots, N_x, \quad y_j = y_0 + jh_y, \quad j = 1, 2, \dots, N_y, \quad (17)$$

с базисными функциями Равьяра–Тома низшей степени A является блочно-диагональной матрицей вида

$$A = \begin{bmatrix} A_x & 0 \\ 0 & A_y \end{bmatrix}, \quad A_x = \text{block} - \text{diag}\{A_j^x\}, \quad A_y = \text{block} - \text{diag}\{A_i^y\}.$$

Здесь N_x, N_y равны числу ячеек сетки по соответствующей координате,

$$A_x \in \mathcal{R}^{\hat{N}_1, \hat{N}_1}, \quad \hat{N}_1 = (N_x - 1)N_y, \quad \hat{A}_y \in \mathcal{R}^{\hat{N}_2, \hat{N}_2}, \quad \hat{N}_2 = N_x(N_y - 1),$$

а A_j^x и A_i^y – трехдиагональные матрицы со строгим диагональным преобладанием порядков $N_x - 1, N_y - 1$ соответственно (в данном варианте размерности матриц приведены для задачи Дирихле). В простейшем модельном случае их ненулевые элементы, с точностью до нормировки, имеют вид $\{1, 4, 1\}$. Матрица B^T при этом является двухдиагональной и имеет следующую структуру:

$$B^T = \begin{bmatrix} B_x^T \\ B_y^T \end{bmatrix}, \quad B_x^T \in \mathcal{R}^{\hat{N}_1, N}, \quad B_y^T \in \mathcal{R}^{\hat{N}_2, N}, \quad N = N_x N_y,$$

а ненулевые элементы $B_x = \{B_i^x\}$ и $B_y = \{B_j^y\}$ равны $\{-1, 1\}$ с точностью до нормировки. Компоненты подвекторов u_a и u_c в алгебраической системе (1) можно сопоставить с реберными и объемными типами точек сетки, обозначенными на рис. 1 символами “ \times ” и “ \circ ” соответственно и имеющими физический смысл потоков субстанции и давления (там же указаны существующие связи между разными типами узлов).

В трехмерном случае для регулярной сетки с конечными элементами в форме параллелепипедов структуры матриц A и B естественным образом обобщаются:

$$A = \begin{bmatrix} A_x & 0 & 0 \\ 0 & A_y & 0 \\ 0 & 0 & A_z \end{bmatrix}, \quad B^T = \begin{bmatrix} B_x^T \\ B_y^T \\ B_z^T \end{bmatrix}, \quad A_z = \text{block} - \text{diag}\{A_k^z\}. \quad (18)$$

Здесь, естественно, предполагается, что определение двумерной сетки (17) дополняется дискретизацией третьей координаты: $z_k = z_0 + kh_z, k = 1, 2, \dots, N_z$.

Исключая из первой блочной строки системы (1) подвектор

$$u_a = A^{-1}(f_a - B^T u_c),$$

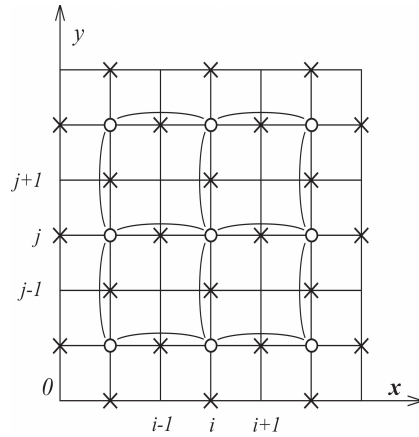


Рис. 1. Структура межузловых связей для двумерной прямоугольной сетки.

можно свести алгоритм Узавы к решению редуцированной системы порядка N_1

$$Su_c = f, \quad S = BA^{-1}B^T \in \mathcal{R}^{N_1, N_1}, \quad f = BA^{-1}f_a - f_c.$$

Если матрица A имеет вид (18), а элементарные блоки A_i^x , A_j^y , A_k^z являются трехдиагональными, то реализация данного метода по формулам сопряженных направлений (16) (здесь в данном случае надо заменить A на S и положить $M = I$) легко осуществляется с помощью алгоритмов прогонки. Однако при переходе к регуляризованному методу Узавы (17) умножение на матрицу \bar{A}^{-1} (даже при диагональной матрице L в простейшем случае) приводит к необходимости формирования двухуровневых итерационных процессов. То же самое относится и к алгоритмам с применением предобуславливателя вида (13).

Для случая вырожденной системы (15) рассмотрим применение итерационного алгоритма с дефляционным вариантом метода сопряженных градиентов. Пусть для общности $V \in \mathcal{R}^{N, m}$ есть прямоугольная матрица, столбцы которой составляют базис некоторого m -мерного дефляционного пространства. В простейшем случае $m = 1$, $V = e$ представляет собой одномерное нуль-пространство матрицы B^T .

Первый этап дефляционного алгоритма заключается в выборе начального приближения u_c^0 по некоторым условиям ортогонолизации. Для произвольного вектора u_c^{-1} определяем

$$\begin{aligned} u_c^0 &= u_c^{-1} + V\bar{c}, \quad \bar{c} \in \mathcal{R}^m, \\ r^0 &= r^{-1} - AV\bar{c}, \quad r^{-1} = f - Au^{-1}. \end{aligned} \quad (19)$$

Вектор коэффициентов \bar{c} в (19) будем искать из переопределенной системы

$$AV\bar{c} = r^{-1}, \quad (20)$$

получаемой формально из (19) по условию $r^0 = 0$. Умножая обе части уравнения (20) на транспонированную матрицу V^T , получаем

$$\bar{c} = (V^T AV)^+ V^T r^{-1}. \quad (21)$$

Здесь $(V^T AV)^+ \in \mathcal{R}^{m,m}$ – обобщенная обратная матрица, которая может быть вычислена, например, по формуле Гревилля [12] или с помощью QR-разложения (если матрица A невырожденная, то $(V^T AV)^+ = (V^T AV)^{-1}$). Определяя при этом начальный вектор направлений p^0 по формуле

$$p^0 = r^0 - V(V^T AV)^+ V^T Ar^0 = B_d r^0, \quad B_d = I - V(V^T AV)^+ V^T A, \quad (22)$$

получаем следующие дефляционные соотношения ортогональности:

$$\begin{aligned} V^T r^0 &= 0, \quad V^T A p^0 = 0, \\ V^T A B_d &= 0, \quad B_d V = 0, \end{aligned} \quad (23)$$

где B_d есть дефляционная предобуславливающая матрица.

Для обеспечения свойств ортогональности на последующих приближениях итерационный процесс выполняется по формулам дефляционного метода сопряженных градиентов, см. [13]:

$$\begin{aligned} u^{n+1} &= u^n + \alpha_n p^n, \quad \alpha_n = \sigma_n / \rho_n, \\ r^{n+1} &= r^n - \alpha_n A p^n, \quad \rho_n = (A p^n, p^n), \\ p^{n+1} &= B_d^{-1} r^{n+1} + \beta_n p^n, \quad \beta_n = \sigma_{n+1} / \sigma_n, \quad \sigma_n = (B_d^{-1} r^n, r^n). \end{aligned} \quad (24)$$

При этом будут удовлетворяться условия

$$V^T r^n = 0, \quad V^T A p^n = 0.$$

Отметим, что если применяются два предобуславливателя вида M и B_d из (16) и (24), то требуется переходить к методологии мульти-предобусловленных итерационных процессов в подпространствах Крылова, см. [14].

Масштабируемое распараллеливание рассматриваемых итерационных процессов осуществляется на многопроцессорной вычислительной системе (МВС) кластерной архитектуры с распределенной и иерархической общей памятью, на основе применения гибридного программирования, включающего средства передачи межпроцессорных сообщений (система MPI), многопоточковых вычислений на многоядерных CPU с помощью системы OpenMP и векторизации операций с применением системы команд типа AVX. Использование графических ускорителей типа GPGPU или INTEL Phi мы для простоты не рассматриваем.

Количественно производительность параллельных вычислений на p арифметических устройствах, или ядрах, при решении задачи или реализации алгоритма A характеризуется двумя коэффициентами – ускорения и эффективности использования процессора:

$$S_p(A) = T_1(A)/T_p(A), \quad E_p(A) = S_p(A)/p.$$

Здесь $T_p(A)$ – время работы МВС с p процессорами, которое складывается из длительности выполнения арифметических действий и передачи данных, т.е.

$$T_p(A) = T_p^a(A) + T_p^c(A).$$

В грубом приближении, последние величины для $p = 1$ можно оценить с помощью формул

$$T_1^a(A) = \tau_a Q_a, \quad T_1^c = \tau_0 + \tau_c Q_c,$$

где τ_a – среднее время одного арифметического действия, τ_0 – время задержки (настройки) одной межпроцессорной пересылки информации, τ_c – время передачи одного числа, а Q_a и Q_c – объемы выполняемых арифметических и коммуникационных операций. Поскольку для реальных МВС имеют место соотношения $\tau_a \ll \tau_c \ll \tau_0$, то из качественных соображений следует, что для повышения производительности вычислений следует, в первую очередь, сокращать как количество передач, так и общий объем пересылаемой информации. Здесь еще

необходимо иметь в виду, что коммуникационные операции потребляют намного больше электроэнергии, чем арифметические, и этот фактор очень существенен с точки зрения эксплуатационных расходов суперЭВМ. Для более точных оценок времени исполнения программ необходимо учитывать существующий большой разброс в значениях τ_a для различных арифметических действий, особенно, в длительности обменов данными между различными уровнями памяти (регистры CPU, кеши разных уровней, общая оперативная память вычислительного узла). Сложность реальных архитектур приводит к тому, что главным инструментом исследований сравнительной производительности программных реализаций алгоритмов является машинный эксперимент.

Технологические принципы распараллеливания численных решений больших СЛАУ рассмотрены в [9]. Основным инструментом достижения высокой производительности здесь является аддитивный метод декомпозиции областей [14], существующий как в геометрическом, так и в алгебраическом вариантах, основанный на сбалансированном разбиении сеточной расчетной области на подобласти, в каждой из которых формируется своя алгебраическая подсистема. Для этих вспомогательных СЛАУ организуются соответствующие MPI-процессы (Message Passing Interface) на узловых кластерах с разделенной памятью. Общая вычислительная схема, как правило, представляет собой двухуровневый блочный метод Шварца–Якоби в подпространствах Крылова. Алгебраические подсистемы в подобластях решаются синхронно на многоядерных CPU (Central Processor Unit) с общей памятью, а “внутреннее” распараллеливание при этом обеспечивается средствами многопоточковых вычислений (с помощью систем типа Open MP). При учете тонких особенностей структуры данных дополнительное ускорение может быть получено за счет оптимизации обменов между регистрами арифметического устройства и кеш-памятью низшего уровня, путем использования системы команд типа AVX (Advanced Vector Extension). Следует сказать, что существенное повышение производительности программного кода может быть достигнуто путем квалифицированного использования готовых программных операций из SPARSE BLAS, активно применяемых в INTEL MKL [15].

§4. ПРИМЕРЫ ЧИСЛЕННЫХ ЭКСПЕРИМЕНТОВ

Для иллюстрации эффективности рассматриваемых итерационных алгоритмов мы остановимся на описанной в [2] математической модели фильтрации двумерной несжимаемой жидкости, в которой суммарная скорость потока \vec{w} и давление p , в пренебрежении силы тяжести, удовлетворяют в ограниченной области Ω соотношениям

$$\vec{w} + \varkappa \operatorname{grad} p = 0, \quad \operatorname{div} \vec{w} = 0. \quad (25)$$

Здесь коэффициент \varkappa характеризует материальные свойства среды, связанные с её проницаемостью, пористостью и другими параметрами, которые могут быть разрывными и сильно контрастными. На границе расчетной области предполагаются заданными некоторые граничные условия для скорости, за деталями которых мы отсылаем к [2], а скалярная функция давления определяется с точностью до постоянной. После сведения соотношений (25) к вариационной постановке с базисными функциями Равьяра–Тома низшего порядка мы приходим к (1), где матрицы A и B для модельных исходных данных принимают простой вид в соответствии с блочной структурой (18). Элементарные блоки вида A_i^x, B_j^x этих матриц несколько различаются в зависимости от типов краевых условий на границах. Например, для задачи Неймана они имеют следующее представление:

$$A_i^x = \frac{1}{6} \begin{bmatrix} 4 & 1 & & 0 \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ 0 & & 1 & 4 \end{bmatrix} \in \mathcal{R}^{N_x+1, N_x+1},$$

$$B_j^x = \frac{1}{h} \begin{bmatrix} -1 & & 0 \\ 1 & \ddots & \\ & \ddots & -1 \\ 0 & & 1 \end{bmatrix} \in \mathcal{R}^{N_x+1, N_x}. \quad (26)$$

В случае задания условия Дирихле с одной или с двух сторон вдоль какой-то из координат, количества столбцов в этих матрицах уменьшаются на единицу или на двойку, а значение соответствующих угловых элементов матриц A_i^x меняется с 4 на 2. Аналогично определяются и матрицы $A_j^y, A_k^z, B_j^y, B_k^z$.

В последующих таблицах мы приводим результаты численных решений для краевых задач Дирихле и Неймана с модельными исходными данными, допускающими простое аналитическое решение. В расчетной кубической области аппроксимации исходных уравнений (25) осуществлялись на кубических сетках с параметрами разбиений $N_x, N_y, N_z = 16, 32, 64$. Коэффициенты и нормировка уравнений выбраны такими, что матрицы решаемых СЛАУ определяются в соответствии с (26). Расчеты проводились со стандартной двойной точностью представления. В качестве критериев окончания итераций выбирались условия

$$(r^n, r^n) \leq \varepsilon^2(f, f), \quad \varepsilon = 10^7$$

как для внешних, так и для внутренних итераций. Расчеты проводились для регуляризованного метода Узавы (15) с выбором матрицы $L^{-1} = \theta I$ при различных параметрах θ . Фактически осуществлялось использование формул метода сопряженных градиентов без предобуславливания (16), в которых надо положить $\gamma = 0, M = I$, а матрицу K заменить на \tilde{K} из (15).

В таблице 1 мы приводим результаты расчетов для задачи Неймана при значениях параметров $\theta = 0.0, 0.1, 0.2, 0.3, 0.6$ на сетках с числом ячеек $N = 16^3, 32^3$ и 64^3 . В каждой клетке таблицы верхнее значение означает количество внешних итераций метода сопряженных градиентов, а нижнее – общее число внутренних итераций.

Как видно из приведенных данных, в методе Узавы без регуляризации количества и внешних, и внутренних итераций растут примерно пропорционально увеличению значения N_x . Введение регуляризации значительно сокращает время вычислений, однако лишь до определенной величины параметра θ (при $\theta > 0.6$ процесс ведет себя непредсказуемо).

Таблица 1. Результаты расчетов для задачи Неймана

$N \setminus \theta$	0	0.1	0.2	0.3	0.6
16^3	23 258	5 80	4 64	4 64	3 48
32^3	47 565	5 160	5 160	4 128	3 96
64^3	100 1228	4 256	4 256	4 256	3 192

Аналогичные данные для задачи Дирихле на тех же сетках приводятся в таблице 2.

Таблица 2. Результаты расчетов для задачи Дирихле

$N \setminus \theta$	0	0.1	0.2	0.9
16^3	21 282	5 80	4 64	3 47
32^3	45 618	5 160	4 128	3 95
64^3	96 1318	5 320	4 256	3 191

В данной ситуации, как и для задачи Неймана, при использовании классического (нерегуляризованного) метода Узавы число итераций растет примерно пропорционально с увеличением N_x . Оптимальное значение параметра равно приблизительно $\theta \approx 0.9$ на всех сетках. В целом, для задачи Дирихле число внешних итераций меньше, чем в предыдущем случае. Отметим, что в этих таблицах в столбцах, соответствующих значению итерационного параметра $\theta = 0$, количества внутренних итераций приведены из чисто методического интереса, поскольку в данном случае матрица A является трехдиагональной, и её обращение легко осуществляется с помощью метода прогонки.

§5. ЗАКЛЮЧЕНИЕ

Конечную цель исследований по алгоритмам решения СЛАУ седлового типа мы рассматриваем как разработку высокопроизводительных программных реализаций для моделирования реальных фильтрационных процессов на современных МВС. Настоящая работа содержит сравнительный анализ ряда перспективных предложений по конструированию специализированных предобусловленных методов в подпространствах Крылова, обсуждение вопросов масштабируемого распараллеливания алгоритмов при решении ресурсоемких многомерных задач фильтрации, а также предварительное экспериментальное рассмотрение результатов численного решения для модельных примеров. Дальнейшие планы включают комплексные исследования как путей

повышения теоретической скорости сходимости итерационных методов, так и технологических аспектов производительности программных разработок, которые осуществляются авторами в рамках библиотеки алгебраических решателей KRYLOV [10], концептуально представляющий собой интегрированное инструментальное окружение для задач вычислительной алгебры (в составе базовой системы моделирования БСМ).

Здесь значительный интерес представляет как сравнительный анализ эффективности различных имеющихся методов, так и апробация их на реальных задачах с многочисленными особенностями искомого решения.

СПИСОК ЛИТЕРАТУРЫ

1. В. П. Ильин, *Математическое моделирование. Часть 1. Непрерывные и дискретные модели*. Новосибирск, изд. СО РАН, 2017.
2. М. И. Иванов, И. А. Кремер, Ю. М. Лаевский, *Об одной противоточковой схеме решения задачи фильтрации*. — Сиб. электрон. мат. изв. **16** (2019), 757–776.
3. M. Benzi, G. H. Golub, J. Liesen, *Numerical solution of saddle point problems*. — Acta Numer. **14** (2005), 1–137.
4. Ю. В. Бычков, Е. В. Чижонков, *Итерационные методы решения седловых задач*. М., БИНОМ. Лаборатория знаний, 2010.
5. Y. Notay, *Convergence of some iterative methods for symmetric saddle point linear systems*. — Matrix Anal. Appl. SIAM **40**, No. 1 (2018), 122–146.
6. R. Estrin, C. Greif, *SPMR: A family of saddle-point minimum residual solvers*. — SIAM J. Sci. Comput. **40**, No. 3 (2018), 1884–1914.
7. C. Greif, M. Wathen, *Conjugate gradient for nonsingular saddle-point systems with a maximally rank-deficient leading block*. — J. Comput. Appl. Math. **358** (2019), 1–11.
8. Р. Е. Попов, А. А. Калинкин, *The method of separation of variables in a problem with a saddle point*. — Russ. J. Numer. Anal. Math. Model. **23**, No. 1 (2008), 97–106.
9. В. П. Ильин, *О проблемах параллельного решения больших СЛАУ*. — Зап. научн. семин. ПОМИ **439** (2015), 112–127.
10. V. P. Il'in, *On an integrated computational environment for numerical algebra*, Springer Nature Switzerland AG 2019 L. Sokolinsky and M. Zumbler (Eds.), Proc. PCT 2019, CCIS 1063 (2019), 91–106.
11. В. П. Ильин, *Методы и технологии конечных элементов*. Новосибирск, ИВ-МиМГ СО РАН, 2007.
12. Ф. Р. Гантмахер, *Теория матриц*. М., Наука, 1958.
13. Я. Л. Гурьева, В. П. Ильин, *О методах грубосеточной коррекции в подпространствах Крылова*. — Зап. научн. семин. ПОМИ **463** (2017), 44–57.
14. V. P. Il'in, *Multi-Preconditioned Domain Decomposition Methods in the Krylov Subspaces*. — LNCS **10187** (2017), 95–106.

15. Intel Math Kernel Library. Reference Manual, <http://software.intel.com/sites/products/documentation/hpc/composerxe/enus/mklxe/mk-manual-win-mac/index.html>

Ильин В. П., Казанцев Г. Ю. Iterative solution of saddle-point SLAEs.

The paper considers preconditioned iterative methods in Krylov subspaces for solving systems of linear algebraic equations (SLAEs) with a saddle point arising from grid approximations of three dimensional boundary-value problems of various types describing filtration flows of a two-phase incompressible fluid. A comparative analysis of up-to-date approaches to block preconditioning of SLAEs under consideration, including issues of scalable parallelization of algorithms on multiprocessor computing systems with distributed and hierarchical shared memory using hybrid programming tools, is presented. A regularized Uzawa algorithm using a two-level iterative process is proposed. Results of numerical experiments for the Dirichlet and Neumann model boundary-value problems are provided and discussed.

Институт вычислительной математики
и математической геофизики СО РАН
E-mail: ilin@sccc.ru

Поступило 14 октября 2019 г.

Новосибирский государственный университет
E-mail: kazancev.grigorij@gmail.com