

ON THE ITERATIVE SOLUTION OF THE
STOKES PROBLEMА.М. ГУРИН¹, В.П. ИЛЬИН², Д.И. КОЗЛОВ^{2,3}, Е.А. КУЗЬМИН^{2,3}¹) Институт гидродинамики имени М. А. Лаврентьева СО РАН,²) Институт вычислительной математики и математической
геофизики СО РАН,³) Новосибирский государственный университет
guralisk@mail.ru

Abstract: Block preconditioned iterative conjugate gradient methods for solving the three-dimensional Stokes problem are investigated. A formulation with a computational domain in the form of a parallelepiped is considered. “Standard” approximations on a cubic staggered grid, seven-point and two-point for the Laplace and derivative operators are used. The resulting saddle-type SLAE is regularized to ensure the uniqueness of the solution. The block preconditioner is constructed by the method of incomplete factorization with diagonal compensation and using band approximations for matrices inverse to the Schur complement and the grid Laplace operator, as well as using an algebraic multigrid algorithm. Examples of numerical experiments on a representative series of methodological problems using parallel algorithms on different numbers of processors are given. The issues of generalizing the proposed approaches to broader classes of problems are considered.

Keywords: Stokes problem, large sparse SLAEs, iterative preconditioned methods, Krylov subspace, Schur complement.

1 Введение

Задача Стокса, получаемая из системы уравнений Навье-Стокса при малых скоростях флюида, является актуальной во многих приложениях гидромеханики, решению которых посвящено огромное количество исследований, см. монографии и обзорные работы [1] - [7]. Мы рассмотрим её в классической бездивергентной постановке для трёхмерного случая:

$$\frac{\partial \rho u}{\partial t} - \mu \Delta \bar{u} + \nabla p = f(\vec{r}), \quad \operatorname{div} \bar{u} = 0, \quad \bar{r} \in \Omega, \quad \bar{u} = (u_1, u_2, u_3), \quad (1)$$

где ρ, μ – предполагаемые для простоты постоянными плотность и динамическая вязкость, а для вектора скорости \bar{u} на границе Γ и для давления в области Ω выполняются условия

$$\bar{u}|_{\Gamma} = \bar{g}(\bar{r}), \quad \int_{\Omega} p d\bar{r} = 0, \quad (2)$$

обеспечивающие существование единственного решения в пространствах $\bar{u} \in (H_0^1(\Omega))^3$, $p \in L_0^2(\Omega)$ [1]. Отметим, в частности, что на границе жидкости с твердой неподвижной стенкой задается условие прилипания $\bar{g}(\bar{r}) = 0$.

Для простоты расчётную область предполагаем параллелепипедом. Исходную задачу (1) - (2) аппроксимируем на кубической смещённой сетке с шагом h , в которой компоненты вектора скорости определены в центрах перпендикулярных им сеточных граней, а давление – в центрах сеточных ячеек ($x_i = ih, y_j = jh, z_k = kh$):

$$\begin{aligned} u_x &= u_1(x_i, y_{j+1/2}, z_{k+1/2}), & u_y &= u_2(x_{i+1/2}, y_j, z_{k+1/2}), \\ u_z &= u_3(x_{i+1/2}, y_{j+1/2}, z_k), & p &= p(x_{i+1/2}, y_{j+1/2}, z_{k+1/2}), \\ i &= 0, 1, \dots, N_1; & j &= 0, 1, \dots, N_2; & k &= 0, 1, \dots, N_3. \end{aligned}$$

После применения “стандартных” неявных конечно-разностных (или конечно-объёмных, или простейших конечно-элементных [8]) аппроксимаций задачи (1) - (2) мы получаем симметричную систему линейных алгебраических уравнений (СЛАУ) седлового типа,

которая в матричном представлении имеет вид

$$Au = \begin{bmatrix} \Delta & \nabla \\ \nabla^\top & 0 \end{bmatrix} \begin{bmatrix} \bar{u} \\ p \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix} = f, \quad \bar{u} = (u_1, u_2, u_3), \quad \bar{f} = (f_1, f_2, f_3). \quad (3)$$

Здесь и далее символы Δ и ∇ обозначают матрицы, получаемые в результате сеточных аппроксимаций соответствующих дифференциальных операторов (для Лапласа – с учётом граничных условий Дирихле (2), фактически этот оператор включает учёт коэффициента μ и аппроксимацию производной по времени, которая усиливает его диагональное преобладание). Отметим, что в общем случае сеточные функции, или подвекторы, u_1, u_2, u_3, p имеют различные размерности, т. е.

$$\begin{aligned} u_1 &\in \mathbb{R}^{N_x}, & u_2 &\in \mathbb{R}^{N_y}, & u_3 &\in \mathbb{R}^{N_z}, & p &\in \mathbb{R}^{N_p}, \\ N_x &= (N_1 - 1)N_2N_3, & N_y &= N_1(N_2 - 1)N_3, & N_z &= N_1N_2(N_3 - 1), \\ & & & & N_p &= N_1N_2N_3, \end{aligned}$$

общий порядок матрицы A при этом равен $N = N_1 + N_2 + N_3 + N_p \equiv N_u + N_p$. Заметим также, что вектор

$$u_0 = \begin{pmatrix} 0 \\ e_p \end{pmatrix}, \quad e_p = \{1\} \in \mathbb{R}^{N_p} \quad (4)$$

есть собственный вектор матрицы A , соответствующий её нулевому собственному числу, причём оно не является кратным, см. [2] - [6]. Это означает, что для согласования СЛАУ (3) необходимо условие ортогональности $(f, e) = 0$, $e = \{1\} \in \mathbb{R}^N$, а если вектор $u = \begin{pmatrix} \bar{u} \\ p \end{pmatrix}$ есть решение СЛАУ (3), и его подвектор p ортогонален соответствующему подпространству ядра матрицы A , т. е. $(p, e_p) = 0$, то такое решение единственно.

При выбираемой нами последовательно-естественной нумерации разных типов точек определения подвекторов компонент матрицы $\Delta, \nabla, \nabla^\top$ примут блочную форму, и матричное представление СЛАУ (3) записывается как

$$Aw = \begin{bmatrix} \Delta_1 & 0 & 0 & \nabla_1 \\ 0 & \Delta_2 & 0 & \nabla_2 \\ 0 & 0 & \Delta_3 & \nabla_3 \\ \nabla_1^\top & \nabla_2^\top & \nabla_3^\top & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ p \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ 0 \end{bmatrix}. \quad (5)$$

В рассматриваемых случаях матрицы Δ_l , $l = 1, 2, 3$, являются семидиагональными, симметричными и положительно определёнными (с. п. о), а ∇_l – треугольными двухдиагональными. Современные практические требования к точности численных методов

обуславливают высокие порядки алгебраических систем вида (5), до нескольких миллиардов ($N \approx 10^9$), что делает востребованным построение высокопроизводительных итерационных алгоритмов, в том числе с их распараллеливанием на многопроцессорных вычислительных системах (МВС) с распределённой и иерархической памятью.

Среди большого числа современных публикаций на данную тему (см., например, [9] - [?]) общая тенденция исследований направлена на использование предобусловленных методов в подпространствах Крылова [7]. Длительное время популярностью пользуются алгоритмы типа SIMPLE (semi-implicit method for pressure-linked equations) и варианты метода Узавы [1]. Отметим, что для решения сеточных уравнений Стокса также эффективно применяются многосеточные подходы, см. [13, 14]/

В данной работе для решения СЛАУ мы исследуем метод сопряжённых градиентов с применением дефляции [16], в целях обеспечения свойства ортогональности давления (2) на каждой итерации. Рассматривается также подход с регуляризацией седловой матрицы, аналогичный разработанному в [17] при решении вырожденной задачи Неймана. В качестве предобуславливателя используется матрица блочной неполной факторизации, в которой операторы Лапласа и дополнения Шура аппроксимируются или с помощью приближения редкой обратной матрицы (SPAI, [18]), или “чисто алгебраическим” многосеточным алгоритмом неполной факторизации, предложенным в [19].

Настоящая работа построена следующим образом. В п. 2 излагаются предобусловленные методы сопряжённых градиентов, в том числе с ускорением при помощи регуляризации СЛАУ и применении дефляции к итерационному процессу. П. 3 посвящён описанию предобуславливающих матриц с помощью подходов SPAI и AMG, а в последнем пункте обсуждаются предварительные результаты численных экспериментов с распараллеливанием алгоритмов на различном числе процессов.

2 Блочный предобусловленный метод сопряжённых градиентов для решения задачи Стокса

Определим алгебраическую регуляризованную задачу Стокса как СЛАУ, являющуюся обобщением (3):

$$A_\gamma u = \begin{bmatrix} A_{1,1} & \nabla \\ \nabla^\top & \gamma A_{2,2} \end{bmatrix} \begin{bmatrix} \bar{u} \\ p \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix}, \quad (6)$$

где $A_{1,1} = \Delta + \nabla K \nabla^\top$, $A_{2,2} = ME$, $E = \{1\} \in R^{N_p, N_p}$ есть матрица с единичными компонентами, $K \in \mathcal{R}^{N_u, N_u}$ и $M \in \mathcal{R}^{N_p, N_p}$ – некоторые с.п.о. - матрицы, а γ – вещественный параметр регуляризации. В силу свойств матрицы градиента $\nabla e_p = 0$ можно показать следующие два результата.

Теорема 1. *Матрица A_γ при достаточно большом значении $\gamma \geq \gamma_0 > 0$ имеет положительные собственные числа, за исключением одного (нулевого), соответствующего собственному вектору u_0 из (4). СЛАУ (6) является согласованной при условии $(f, u_0) = 0$.*

Теорема 2. *Алгебраическая система (6) в условиях теоремы 1 имеет единственное решение, которое обладает условием ортогональности для давления $(p, e_p) = 0$, и это решение не зависит от γ .*

Замечание 1. *Вместо условия ортогональности давления p ядру матрицы A_γ , однозначность решения СЛАУ (6) можно обеспечить, фиксируя значение (например, нулевое) давления p в одном из узлов сетки. Однако можно ожидать, по аналогии с вырожденной задачей Неймана для уравнения Пуассона, что при этом значительно вырастет число обусловленности матрицы A_γ . Более строгий анализ данного вопроса требует специальных исследований.*

Замечание 2. *Наличие регуляризующего параметра γ в матрице A_γ формально даёт возможность минимизировать её число обусловленности и количества итераций при решении СЛАУ (6). Теоретически этот вопрос пока не исследован, а приближённая оптимизация γ проводится экспериментально.*

Метод сопряжённых градиентов с предобуславливающей матрицей B для решения СЛАУ (5), запишем в следующем виде [16]:

$$\begin{aligned} r^0 &= f - A_\gamma u^0, & q^0 &= B^{-1}r^0, & \alpha_n &= \sigma_n/\rho_n, \\ u^{n+1} &= u^n + \alpha_n q^n, & r^{n+1} &= r^n - \alpha_n A_\gamma q^n, & q^{n+1} &= B^{-1}r^{n+1} + \beta_n q^n, \\ \sigma_n &= (r^n, q^n) = (B^{-1}r^n, r^n), & \rho_n &= (A_\gamma q^n, q^n), & \beta_n &= \sigma_{n+1}/\sigma_n, \end{aligned} \quad (7)$$

где вектор начального приближения предполагается ортогональным ядру матрицы A_γ , т. е. $(u^0, u_0) = 0$, для чего достаточно обеспечить условие $\sum_{l=1}^{N_p} p_l^0 = 0$. При этом для достижения условия сходимости

$$\|r^n\| \leq \varepsilon \|f\|, \quad \varepsilon \ll 1$$

достаточное число итераций оценивается величиной

$$n(\varepsilon) \leq \frac{1}{2} |\ln \frac{\varepsilon}{2}| (\text{cond}(B^{-1}A_\gamma)^{1/2} + 1).$$

Построение предобуславливающей матрицы B определим на основе блочно-факторизованного представления матрицы \bar{A}_γ , приближённо равной матрице A_γ из СЛАУ (6):

$$B = \bar{A}_\gamma = \begin{bmatrix} \bar{A}_{1,1} & \nabla \\ \nabla^\top & \gamma A_{2,2} \end{bmatrix} = \begin{bmatrix} I & 0 \\ \nabla^\top \bar{A}_{1,1}^{-1} & I \end{bmatrix} \begin{bmatrix} \bar{A}_{1,1} & 0 \\ 0 & \bar{S}_\gamma \end{bmatrix} \begin{bmatrix} I & \bar{A}_{1,1}^{-1} \nabla \\ 0 & I \end{bmatrix}, \quad (8)$$

где $\bar{A}_{1,1}$ – некоторая аппроксимация матрицы $A_{1,1}$, $\bar{S}_\gamma = \gamma A_{2,2} - \nabla^\top \bar{A}_{1,1}^{-1} \nabla$ есть дополнение Шура для матрицы \bar{A}_γ . Предобуславливатель конструируем с помощью одной из двух аппроксимаций матрицы $\bar{A}_{1,1}^{-1}$. Первый способ заключается в приближённом ленточном представлении обратной матрицы SPAI (см. [18] и приведённую там литературу), а второй – в применении алгебраического многосеточного метода (AMG, см. обзор в [19]). В обоих случаях предобуславливатель строится путём замены точной матрицы $A_{1,1}$ из (6) на приближённую. На каждой итерации требуется решать вспомогательную СЛАУ вида $Bq = r$, что экономично реализуется с помощью прямого и обратного хода факторизации:

$$Bq = \begin{bmatrix} \bar{A}_{1,1} & 0 \\ \nabla^\top & \bar{S}_\gamma \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}, \quad \begin{bmatrix} I & \bar{A}_{1,1}^{-1} \nabla \\ 0 & I \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}. \quad (9)$$

В компонентной форме эти соотношения записываются следующим образом:

$$\begin{aligned} v_1 &= \bar{A}_{1,1}^{-1} r_1, & \bar{S}_\gamma v_2 &= (\gamma E - \nabla^\top \bar{A}_{1,1}^{-1} \nabla) v_2 = r_2, \\ q_2 &= v_2, & q_1 &= v_1 - \bar{A}_{1,1}^{-1} \nabla q_2, \end{aligned} \quad (10)$$

где $\bar{A}_{1,1}$ означает какой-то из рассматриваемых предобуславливателей к матрице $A_{1,1}$, на которых остановимся в следующем разделе. Отметим, что в соотношениях (10) при вычислениях v_1 и q_1 необходимо дважды умножать вектор на ленточную матрицу $\bar{A}_{1,1}^{-1}$, однако нахождение v_2 требует решения СЛАУ с плотной матрицей \bar{S}_γ . Здесь, тем не менее, можно заметить два обнадеживающих фактора. Во-первых, умножение на неё делается очень экономично, поскольку $Ev = (v, e_p)e_p$. Во-вторых, по аналогии с работой [17], следует ожидать хорошей обусловленности (не зависящей от h) матрицы \bar{S}_γ , за счёт выбора параметра γ и матриц K, M . В таком случае целесообразно было бы применить эффективный алгоритм чебышёвского ускорения, не требующий плохо распараллеливаемых векторных скалярных произведений, но при этом требуется определять границы спектра матрицы \bar{S}_γ .

Отметим, что для обеспечения условия ортогональности давления p к ядру матрицы A из (3) на всех итерациях (которые выполняются с неизбежными машинными округлениями) предобусловленный метод сопряжённых градиентов следует применять с дефляцией, заключающейся в добавлении к базисам подпространств Крылова “дефляционного” вектора v , ортогональному ядру матрицы A_γ , т.е. $(v, u_0) = 0$ что приводит к следующим формулам (при этом предварительно осуществляется ортогонализация начального приближения и направляющего вектора):

$$\begin{aligned} u^0 &= u^{-1} + vc = u^{-1} + Hr^{-1}, & H &= v\hat{A}_\gamma^{-1}v^\top, \\ c &= \hat{A}_\gamma^{-1}v^\top r^{-1}, & \hat{A}_\gamma &= v^\top Av, & r^{-1} &= f - A_\gamma u^{-1}, \end{aligned}$$

где u^{-1} – произвольный вектор, удовлетворяющий условию $(u^{-1}, v) = 0$.

$$\begin{aligned} r^0 &= f - A_\gamma u^0, & q^0 &= QB^{-1}r^0, & Q &= I - HA_\gamma, & n &= 0, 1, \dots : \\ u^{n+1} &= u^n + \alpha_n q^n, & r^{n+1} &= r^n - \alpha_n A_\gamma q^n, & \alpha_n &= \sigma_n / \rho_n, \\ q^{n+1} &= QB^{-1}r^{n+1} + \beta_n q^n, & \beta_n &= \sigma_{n+1} / \sigma_n, \\ \sigma_n &= (B^{-1}r^n, r^n), & \rho_n &= (Aq^n, q^n). \end{aligned} \tag{11}$$

В этих соотношениях матрица \hat{A} введена для общности (если дефляционных векторов несколько, то $v = (v_1, \dots, v_m) \in \mathbb{R}^{N,m}$ есть дефляционная матрица, а в нашем случае $m = 1$ и \hat{A} есть фактически скаляр).

Для решения блочного СЛАУ второго порядка на основе точной LDU-факторизации (8) можно также предложить двустороннее экономичное предобуславливание следующего типа:

$$\begin{aligned}
 A_\gamma &= LDU, \quad L = \begin{bmatrix} I & 0 \\ A_{2,1}A_{1,1}^{-1} & I \end{bmatrix}, \quad U = \begin{bmatrix} I & A_{1,1}^{-1}A_{1,2} \\ 0 & I \end{bmatrix}, \quad D = \begin{bmatrix} A_{1,1} & 0 \\ 0 & S \end{bmatrix}, \\
 S_\gamma &= \gamma A_{2,2} - A_{2,1}A_{1,1}^{-1}A_{1,2}, \quad L^{-1} = \begin{bmatrix} I & 0 \\ -A_{2,1}A_{1,1}^{-1} & I \end{bmatrix}, \quad U^{-1} = \begin{bmatrix} I & -A_{1,1}^{-1}A_{1,2} \\ 0 & I \end{bmatrix}, \\
 \bar{A}\bar{u} &= L^{-1}AU^{-1}\bar{u} = L^{-1}f = \bar{f}, \quad Uu, u = U^{-1}\bar{u}, \\
 D\bar{u} &= \begin{bmatrix} A_{1,1} & 0 \\ 0 & S_\gamma \end{bmatrix} \begin{bmatrix} \bar{u}_1 \\ \bar{u}_2 \end{bmatrix} = \begin{bmatrix} \bar{f}_1 \\ \bar{f}_2 \end{bmatrix}.
 \end{aligned} \tag{12}$$

При итерационном решении СЛАУ (12) с матрицей D на каждой итерации требуется однократное решение независимых подсистем с матрицами $A_{1,1}$ и S_γ . Векторы \bar{f} , u вычисляются по одному разу, перед и после итераций соответственно.

В данном алгоритме требуется многократное решение алгебраических систем с матрицей $A_{1,1}$ в частности, при реализации двухуровневого итерационного процесса для СЛАУ с матрицей S_γ . В этих случаях наиболее целесообразно применять экономичный многосеточный метод AMG. Что касается внешних итераций, то здесь, вообще говоря, необходимо использовать предобусловленные алгоритмы с длинными рекурсиями типа FCG или GMRES. Предобуславливающую матрицу целесообразно брать на основе неполной факторизации или SSOR метода в применении к приближённому дополнению Шура

$$\bar{S}_\gamma = \gamma M - A_{2,1}\bar{A}_{1,1}^{-1}A_{1,2}, \tag{13}$$

где $\bar{A}_{1,1}^{-1}$ – ленточная аппроксимация обратной матрицы $A_{1,1}^{-1}$ (SPAI), в качестве M проще взять единичную матрицу, а параметр γ подбирается экспериментально. При этом матрица $\bar{S} = D_s + L_s + U_s$ разбивается на диагональную, нижнюю и верхнюю треугольные части, а предобуславливатель определяется формулой $B_s = (G_s + L_s)G_s^{-1}(G_s + U_s)$, подробнее о G_s – см. [7].

Для решения СЛАУ (6) с матрицей A_γ мы рассмотрим ещё построение предобуславливателя на основе классического блочного метода неполной факторизации с диагональной компенсацией, базирующегося на построении предобуславливателя следующего вида:

$$B_\gamma = \begin{bmatrix} A_{1,1} & 0 \\ A_{2,1} & \gamma A_{2,2} \end{bmatrix} \begin{bmatrix} A_{1,1}^{-1} & 0 \\ 0 & G_\gamma^{-1} \end{bmatrix} \begin{bmatrix} A_{1,1} & A_{1,2} \\ 0 & G_\gamma \end{bmatrix}, \tag{14}$$

где матрица G_γ есть фактически некоторая аппроксимация дополнения Шура, представляемая формулой

$$\begin{aligned} G_\gamma &= \gamma A_{2,2} - A_{2,1} \overline{A_{1,1}}^{-1} A_{1,2} - \theta R, \quad \theta \in (0, 1) \\ Re_2 &= (A_{2,1} A_{1,1}^{-1} A_{1,2} - A_{2,1} \overline{A_{1,1}}^{-1} A_{1,2}) e_2 = \\ &= A_{2,1} (A_{1,1}^{-1} - \overline{A_{1,1}}^{-1}) A_{1,2} e_2, \quad e_2 = \{1\} \in \mathcal{R}^{N_p}. \end{aligned} \quad (15)$$

Здесь $G \in (0, 1)$ – компенсирующий параметр, а R – диагональная матрица, а применение соотношений (14), (15) (без регуляризации, т.е. $\gamma = 0$) обеспечивает при $G = 1$ условие полной компенсации, или согласования строчных сумм

$$B_\gamma e = A_\gamma e, \quad e = \{1\} \in \mathcal{R}^N, \quad (16)$$

характеризующее в определённом смысле близость исходной и предобуславливающей матриц.

3 Построение предобуславливателей для оператора Лапласа

Самый традиционный из универсальных способов аппроксимации симметричной матрицы заключается в приближённом разложении её на треугольные множители. Если матрица $A_{1,1}$ представима в виде $A_{1,1} = D + L + U$, где D, L, U – в общем случае её блочные диагональная, нижняя и верхняя треугольные части, то предобуславливатель можно записать в виде

$$B_1 = (G + L) G^{-1} (G + U) = G + L + U + L G^{-1} U. \quad (17)$$

Поскольку в формуле (17) представление $G = D - L G^{-1} U$, обеспечивающее точную факторизацию ($B_1 = A_{1,1}$), слишком ресурсоёмко, оно заменяется одной из двух возможных диагональных “аппроксимаций”, реализующих неполную (приближённую) факторизацию. Первая основана на принципе компенсации, или согласования строчных сумм:

$$G = D - \overline{L G^{-1} U} - \theta R, \quad Re = (L G^{-1} U - \overline{L G^{-1} U}) e, \quad \theta \in [0, 1],$$

где θ – компенсирующий параметр, e – пробный вектор, обычно выбираемый из единичных компонент, а черта над матрицей означает взятие её диагональной части, т.е. $\overline{M} = \text{diag}\{m_{i,i}\}$. Здесь R есть диагональная матрица, обеспечивающая выполнение условия “полной компенсации” $B_1 e = A_{1,1} e$ при $\theta = 1$, или согласование строчных сумм [7]. Данный алгоритм будем называть CIF (Compensated Incomplete Factorization).

Второй подход заключается в использовании последовательной верхней релаксации с параметром релаксации ω , который обозначается как SSOR (Symmetric Successive Over Relaxation):

$$G = \omega^{-1}D, \quad \omega \in (0, 2). \quad (18)$$

По поводу выбора итерационных параметров θ или ω см. [20].

Для построения аппроксимации непосредственно обратной матрицы $\overline{A}_{1,1}^{-1}$ существует оригинальный подход SPAI, в котором приближение определяется в виде разреженной матрицы с тем же портретом, что и у исходной матрицы Δ . Конкретные алгоритмы при этом строятся на основе минимизации нормы Фробениуса для матричной невязки

$$\|I - A_{1,1}\overline{A}_{1,1}^{-1}\|_F = \sum_j (e_j - A_{1,1}m_j)^2, \quad (19)$$

где m_j и e_j – столбцы матриц $\overline{A}_{1,1}^{-1}$ и I с номером j . Здесь ненулевые неизвестные m_j находятся из независимого решения систем $A_{1,1}m_j = e_j$ методом наименьших квадратов [18], а получаемый при этом алгоритм естественным образом распараллеливается. Отметим, что поскольку в алгоритме приближенная обратная может получиться несимметричной даже для симметричной матрицы $A_{1,1}$, рекомендуется применять следующую процедуру симметризации:

$$(\overline{A}_{1,1}^{-1})_s = \frac{1}{2}(\overline{A}_{1,1}^{-1} + (\overline{A}_{1,1}^{-1})^\top). \quad (20)$$

В качестве дальнейшей модификации метода SPAI можно предложить использованный в (18) алгоритм диагональной компенсации, в котором приближенная обратная матрица определяется как

$$(\overline{A}^{-1})_c = (\overline{A}^{-1})_s + \theta R, \quad Re_1 = A^{-1}e_1 - (\overline{A}^{-1})_s e_1, \quad e_1 \in R_1^{N_1} \quad (21)$$

где R – диагональная матрица, обеспечивающая согласование строчных сумм $(\overline{A}^{-1})_c e_1 = A^{-1}e_1$.

Наиболее перспективным представляется подход, основанный на “чисто алгебраической” версии многосеточного метода AMG, предложенной в [19] и основанного на неполной факторизации с диагональной компенсацией предобуславливателя блочно-треугольного вида, формируемого из иерархического определения матричного графа решаемой на вложенных сетках СЛАУ.

Данный подход базируется на построении рекурсивной сеточной и соответствующей алгебраической структур. Как обычно, многосеточные алгоритмы трактуются как рекурсивное применение

двухсеточных, определение которых проведём на простом, но достаточно общем, примере следующим образом.

Пусть исходная СЛАУ (3) аппроксимируется на кубической сетке Ω^h с шагом h , содержащей последовательность вложенных сеток Ω_l^h , $l = 1, \dots, m$, каждая с числом узлов N_l , таких, что $\Omega^h = \Omega_1^h \supset \Omega_2^h \supset \dots \supset \Omega_m^h$, причём граница Γ проходит по линиям (или плоскостям) самой редкой сетки Ω_m^h , а каждая более густая сетка Ω_{l-1}^h , $l = 2, \dots, m$, получается из Ω_l делением шага предыдущей сетки пополам. Рассматриваемый далее алгоритм для решения систем с матрицей $A_{1,1}$ фактически основывается на применении классического алгебраического метода неполной факторизации при специальной перенумерации сеточных узлов и соответствующих векторных компонент.

Каждое из соответствующих множеств сеточных узлов и определённых в них векторных компонент разобьём на два подмножества, в результате чего получаем следующие рекурсивные представления исходной сетки и векторов:

$$\begin{aligned} \Omega_1^h &= \bar{\Omega}_1^h \cup \Omega_2^h = \bar{\Omega}_1^h \cup \bar{\Omega}_2^h \dots \cup \bar{\Omega}_{m-1}^h \cup \Omega_m^h, \\ u &= u^{(1)} = ((\bar{u}^{(1)})^\top, (u^{(2)})^\top)^\top = \\ &((\bar{u}^{(1)})^\top, (\bar{u}^{(2)})^\top, \dots, \bar{u}^{(m-1)})^\top, u^{(m)})^\top. \end{aligned} \quad (22)$$

Здесь каждая блочная векторная компонента $u^{(l)}$ уровня $l = 1, \dots, m-1$ разбивается на две части $(\bar{u}^{(l)}$ и $u^{(l+1)})$, вторая из которых представляет следующий уровень и тоже делится надвое так, что вторая часть $(l+1)$ -го уровня совпадает с целым $(l+2)$ -ым уровнем, и т.д. Нумерация сеточных узлов и соответствующих векторных компонент осуществляется последовательно: сначала для $\bar{\Omega}^h$ и $\bar{u}^{(1)}$, затем для $\bar{\Omega}_2^h$, $\bar{u}^{(2)}$, и т. п. В этом случае для двухсеточного варианта матрица $A_{1,1}$ приводится к блочному виду второго порядка. В трёхсеточном методе правый нижний блок матрицы аналогично, в свою очередь, сводится к блочному виду второго порядка, а в многосеточном подходе эта процедура повторяется многократно.

При более детальной классификации узлов, которые мы будем использовать, матрица $A_{1,1}$ приводится к блочному четвёртому порядку. Для этого при $l = 1$ узлы Ω_l^h разобьём на четыре подмножества: $\Omega_l^h = \Omega_l^1 \cup \Omega_l^2 \cup \Omega_l^3 \cup \Omega_l^4$, $\Omega_l^4 = \Omega_{l+1}^h$, которые относятся последовательно к центрам объёмов и граней, к серединам ребер

и узлам вложенной редкой сетки Ω_{l+1}^h . Для сокращения дальнейшего изложения рекурсивного подхода мы здесь, вплоть до формулы (24), при описании исходной матрицы $A_{1,1} = A_{1,1}^{(1)}$ используем индекс l , подразумевая его значение $l = 1$. Обозначая через $\bar{u}_1^{(l)}, \bar{u}_2^{(l)}, \bar{u}_3^{(l)}, \bar{u}_4^{(l)}$ и $\bar{f}_1^{(l)}, \bar{f}_2^{(l)}, \bar{f}_3^{(l)}, \bar{f}_4^{(l)}$ ассоциированные подвекторы (с размерностями $N_1^{(l)}, N_2^{(l)}, N_3^{(l)}, N_4^{(l)}$) в l -й СЛАУ, запишем её в соответствующем блочном виде:

$$A_{1,1}^{(l)} u^{(l)} = \begin{bmatrix} D_{1,1}^{(l)} & A_{1,2}^{(l)} & 0 & 0 \\ A_{2,1}^{(l)} & D_{2,2}^{(l)} & A_{2,3}^{(l)} & 0 \\ 0 & A_{3,2}^{(l)} & D_{3,3}^{(l)} & A_{3,4}^{(l)} \\ 0 & 0 & A_{4,3}^{(l)} & D_{4,4}^{(l)} \end{bmatrix} \begin{bmatrix} \bar{u}_1^{(l)} \\ \bar{u}_2^{(l)} \\ \bar{u}_3^{(l)} \\ \bar{u}_4^{(l)} \end{bmatrix} = \begin{bmatrix} \bar{f}_1^{(l)} \\ \bar{f}_2^{(l)} \\ \bar{f}_3^{(l)} \\ \bar{f}_4^{(l)} \end{bmatrix}. \quad (23)$$

Здесь можно обозначить $\bar{u}_4^{(l)} = u^{(l+1)}$, $\bar{f}_4^{(l)} = \bar{f}^{(l+1)}$, так как соответствующие подвекторы представляют структуры следующего сеточного уровня. Заметим, что блоки $D_{1,1}^{(l)}, \dots, D_{4,4}^{(l)}$ в (23) (напомним, что пока $l = 1$) являются диагональными матрицами.

Отметим, что если рассмотреть уравнения узлового типа на вложенных трёхмерных сетках, составленных из тетраэдров, то множество узлов густой сетки Ω_1^h представляется состоящим только из двух подмножеств: $\bar{\Omega}_1^h$ – подмножество середин рёбер редкой сетки и само подмножество узлов Ω_2^h . Поэтому блочная структура СЛАУ в данном случае имеет второй порядок. В целом рассмотренные матричные представления двухсеточных, или двухуровневых, СЛАУ можно определить как k -блочные структуры, где $k = 2, 3, 4$ обозначает блочный порядок алгебраической системы (23) соответственно.

Для матрицы $A^{(l)}$ предобуславливатель определяем в факторизованной форме:

$$\begin{aligned}
 B^{(l)} &= \begin{bmatrix} G_1^{(l)} & 0 & 0 & 0 \\ A_{2,1}^{(l)} & G_2^{(l)} & 0 & 0 \\ 0 & A_{3,2}^{(l)} & G_3^{(l)} & 0 \\ 0 & 0 & A_{4,3}^{(l)} & G_4^{(l)} \end{bmatrix} (G^{(l)})^{-1} \begin{bmatrix} G_1^{(l)} & A_{1,2}^{(l)} & 0 & 0 \\ 0 & G_2^{(l)} & A_{2,3}^{(l)} & 0 \\ 0 & 0 & G_3^{(l)} & A_{3,4}^{(l)} \\ 0 & 0 & 0 & G_4^{(l)} \end{bmatrix}, \\
 G_1^{(l)} &= D_{1,1}^{(l)}, \quad G_2^{(l)} = D_{2,2}^{(l)} - (A_{2,1}^{(l)}(G_1^{(l)})^{-1}A_{1,2}^{(l)})_1 - \theta S_2^{(l)}, \\
 S_2^{(l)} e_2 &= \left[A_{2,1}^{(l)}(G_1^{(l)})^{-1}A_{1,2}^{(l)} - (A_{2,1}^{(l)}(G_1^{(l)})^{-1}A_{1,2}^{(l)})_1 \right] e_2, \\
 G_3^{(l)} &= D_{3,3}^{(l)} - \left(A_{3,2}^{(l)}(G_2^{(l)})^{-1}A_{2,3}^{(l)} \right)_1 - \theta S_3^{(l)}, \\
 S_3^{(l)} e_3 &= \left[A_{3,2}^{(l)}(G_2^{(l)})^{-1}A_{2,3}^{(l)} - \left(A_{3,2}^{(l)}(G_2^{(l)})^{-1}A_{2,3}^{(l)} \right)_1 \right] e_3, \\
 G_4^{(l)} &= D_{4,4}^{(l)} - A_{4,3}^{(l)}(G_3^{(l)})^{-1}A_{3,4}^{(l)}.
 \end{aligned} \tag{24}$$

Здесь $A_{k,k}^{(l)}$, $k = 1, \dots, 4$ и $G_1^{(l)}, G_2^{(l)}, G_3^{(l)}$ – диагональные матрицы, а $G_4^{(l)}$ – семидиагональная с тем же матричным портретом, что и $A_{1,1}^{(l)}$, и именно её мы будем считать матрицей СЛАУ для сетки следующего уровня. Таким образом, при $l = 1$ и $m > 1$ в (24) следует положить $G_4^{(1)} = A_{1,1}^{(2)}$, определить для данной матрицы блочно-трёхдиагональное представление (23) и предобуславливатель $B^{(2)}$. Далее для $l = 2, \dots, m$ рекурсивное формирование $G_4^{(l)}$ продолжается.

Запись вида $(C)_1$ в (24) обозначает диагональную часть матрицы C , $\theta \in [0, 1]$ – параметр компенсации, а пробные векторы (обычно их компоненты берут равными единице) $e_2^{(l)}, e_3^{(l)}$ имеют размерности $N_2^{(l)}, N_3^{(l)}$. Для конкретизации предобуславливающей матрицы $B^{(l)}$ нам ещё необходимо представить матрицу $G_4^{(l)}$ в факторизованной форме, и при этом будем рассматривать два варианта. Если $l + 1 = \bar{m}$ есть номер последней применяемой сетки, то используется точное треугольное разложение $G_4^{(l)} = A^{(l+1)} = L^{(l+1)}U^{(l+1)}$. В противном случае матрица $G_4^{(l)}$ в (24) приближённо заменяется факторизованным предобуславливателем $B^{(l+1)}$, определяемым в соответствии с этой же формулой. При использовании факторизованного предобуславливателя B из (21) в методе сопряжённых градиентов (7) решение вспомогательной СЛАУ вида $Bq = r$ легко

реализуется по следующим формулам (точнее говоря, ниже представлен псевдокод):

$$\begin{aligned}
 G_1^{(l)} q_1^{(1)} &= r_1^{(1)}, & G_2^{(l)} q_2^{(l)} &= r_2^{(l)} - A_{2,1}^{(l)} q_1^{(1)}, & G_3^{(l)} q_3^{(l)} &= r_3^{(l)} - A_{3,2}^{(l)} q_2^{(l)}, \\
 G_4^{(l)} q_4^{(l)} &= r_4^{(l)} - A_{4,3}^{(l)} q_3^{(l)}, & q_3^{(l)} &= q_3^{(l)} - (G_3^{(l)})^{-1} A_{3,4}^{(l)} q_4^{(l)}, \\
 q_2^{(l)} &= q_2^{(l)} - (G_2^{(l)})^{-1} A_{2,3}^{(l)} q_3^{(l)}, & q_1^{(l)} &= q_1^{(l)} - (G_1^{(l)})^{-1} A_{1,2}^{(l)} q_2^{(l)}
 \end{aligned} \tag{25}$$

Применение предобуславливателя в форме (24) формально соответствует методу неполной факторизации IFM (Incomplete Factorization Method, см. [20]) с согласованием строчных сумм, т. е. по условию $B^{(l)} e^{(l)} = A^{(l)} e^{(l)}$ при $\theta = 1$. Отметим, что реализация такого подхода фактически включает этапы редукции, грубосеточной коррекции и продолжения, обычных для “классических” многосеточных алгоритмов, но без применения сглаживания, см. обзор в [23].

4 Примеры численных экспериментов

Мы проиллюстрируем эффективность предложенных подходов результатами предварительных экспериментальных исследований на представительной серии методических задач. Исходные уравнения решаются в кубической расчетной области с однородной средой на регулярных равномерных сетках с числом ячеек $32^3, 64^3, 128^3$.

Также расчёты выполнены на искусственном керне размером $120 \times 120 \times 124$ ячейки, который построен как случайная разномасштабная пористая среда. Для генерации модели создан набор структурированных кубических сеток с размером $14^3, 20^3, 40^3, 60^3$. На каждой сетке определяются случайные величины в центрах соответствующих ячеек в диапазоне от 0 до 1. Затем была произведена трикубическая интерполяция, как описано в статье [22], с этих сеток на конечную сетку $120 \times 120 \times 124$, после чего значения, полученные с грубых сеток, были просуммированы и на конечной сетке получено распределение значений от 0 до 4. Ячейки, в которых значение превышало определённое пороговое значение, были обозначены как проницаемые. Таким образом, была получена пористая структура, напоминающая реальный керн. Пороговое значение было выбрано такое, чтобы получить пористость керна 50%.

Единственное принципиальное отличие от реальных кернов со связной непроницаемой средой – это наличие непроницаемых островков, окружённых проницаемыми ячейками. Однако это отличие

не мешает тестированию решателей. Для построения окончательной расчетной сетки непроницаемые ячейки были удалены. В приведенных ниже таблицах расчеты на этой сетке обозначены как “kern”. Внешний вид расчетной сетки показан на рисунке 1.

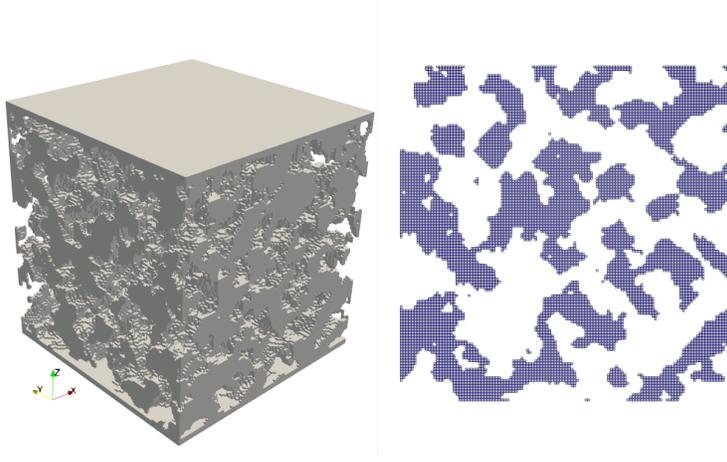


Рис. 1. Искусственный керн размером $120 \times 120 \times 124$ и центральное сечение перпендикулярное оси z .

Для всех сеток величина шага постоянна, $h = 1$, так что фактически при различных дискретизациях размеры расчётных областей отличаются. Нижняя и верхняя грани куба, перпендикулярные оси z , являются точками входа и выхода потока флюида, с заданными одинаковыми вертикальными скоростями $u_z = 1/$. Боковые поверхности куба рассматриваются как непроницаемые стенки с условиями прилипания $u_x = u_y = u_z = 0$. Вязкость среды задается постоянной и равной $\mu = 5 \cdot 10^{-3} \text{Па} \cdot \text{с}$, а плотность — $\rho = 1000/3$. Внутри расчётной области начальная скорость и давление полагаются нулевыми. Во всех таблицах результаты для сеток 32^3 , 64^4 , 128^3 относятся к задачам с однородной средой, а слово kern соответствует модели случайной пористой среды на сетке размером $120 \times 120 \times 124$.

Вычисления проводились в Yandex Cloud на узле Intel Xeon Gold 6338 с 48-ю ядрами, 96-ю потоками и с частотой 2 Ghz. В нижеследующих таблицах приводятся результаты предварительных экспериментальных исследований для модельных задач. Целью расчётов было сравнение эффективности итерационных процессов и анализ масштабируемости распараллеливания при использовании различных временных шагов Δt и разного числа параллельных процессов $P = 2, 4, 8, 16, 32, 48$. Во всех таблицах результаты приводятся для

расчёта одного (первого) шага по времени. Разработанный авторами программный код для решения задач на неструктурированных сетках реализован на платформе INMOST [26], с существенным использованием функциональных инструментов библиотеки PETSc [25].

ТАБЛИЦА 1. Числа итераций и времени расчёта для $\Delta t = 0.5$.

$N \setminus P$	2	4	8	16	32	48
32^3	40/10.1	41/6.2	41/4.1	41/3.3	41/2.9	41/2.9
64^3	81/215.8	82/123.3	82/79.7	82/60.6	82/45.7	82/41.2
128^3	-	-	-	167/1136.6	166/790.2	167/684.3
kern	42/236.8	42/122.2	42/63.2	42/37.2	42/22.8	42/18.0

ТАБЛИЦА 2. Результаты расчётов для $\Delta t = 0.5 \cdot 10^{-3}$.

$N \setminus P$	2	4	8	16	32	48
32^3	40/9.4	39/5.5	40/3.7	40/2.8	40/2.5	40/2.4
64^3	72/183.0	73/100.4	73/62.2	74/47.1	74/32.1	74/30.8
128^3	-	-	-	131/816.9	131/571.0	131/489.4
kern	44/251.0	44/121.9	44/65.0	44/38.8	44/24.5	44/19.3

ТАБЛИЦА 3. Результаты расчётов для $\Delta t = 10^{-7}$.

$N \setminus P$	2	4	8	16	32	48
32^3	10/1.9	10/1.1	10/0.6	10/0.5	10/0.4	10/0.4
64^3	10/20.9	10/10.7	10/6.0	10/3.9	10/2.5	10/2.2
128^3	10/205.4	10/105.6	10/60.0	10/37.8	10/25.7	10/21.1
kern	11/62.0	11/31.1	11/16.5	11/9.8	11/5.8	11/4.6

ТАБЛИЦА 4. Зависимость числа итераций от параметра регуляризации γ (сетка 64^3 , см. табл.2).

γ	0.05	0.01	0	-0.1	-0.2	-1	-2	-4	-8
n	∞	83	74	44	35	20	17	16	10

ТАБЛИЦА 5. Зависимость числа итераций от параметра регуляризации γ (сеточная модель kern, см. табл.2).

γ	0.005	0.001	0	-0.005	-0.05	-0.01	-0.1
n	∞	59	44	78	186	100	251

ТАБЛИЦА 6. Результаты расчёта для $\Delta t = 0.5$, при использовании диагональной компенсации (21) для аппроксимации дополнения Шура.

$N \setminus P$	2	4	8	16	32	48
32^3	23/5.4	23/3.9	23/2.7	24/2.3	23/2.0	23/1.9
64^3	28/65.3	28/41.6	28/27.7	28/22.3	28/17.7	28/16.7
128^3	31/635.4	31/387.1	31/284.8	31/229.4	31/163.1	31/136.6
kern	65/376.7	65/189.3	65/98.6	65/58.1	65/36.5	65/29.0

ТАБЛИЦА 7. Результаты расчёта для $\Delta t = 0.5 \cdot 10^{-3}$, алгоритм без предобуславливателя дополнения Шура.

$N \setminus P$	2	4	8	16	32	48
32^3	14/2.1	14/1.4	14/1.0	14/0.81	14/0.77	14/0.74
64^3	15/21.4	15/13.3	15/9.2	15/7.6	15/5.7	15/5.3
128^3	-	17/122.5	17/84.0	17/73.9	17/53.4	17/46.6
kern	-	-	490/354.2	490/208.6	490/126.1	489/104.2

Исследования осуществлялись для итерационного процесса с предобуславливанием СЛАУ по формуле (12). При этом решение вспомогательной системы с дополнением Шура осуществлялось итерационными методами FGMRES или CG, с применением предобуславливателя \bar{S}_γ (с разными значениями γ) или без него. Матрица $A_{1,1}^{-1}$ “обращалась” с помощью алгоритма BoomerAMG [24] из библиотеки Nupre, а её аппроксимация $\bar{A}_{1,1}^{-1}$ определялась методом SPAI. После этого матрица \bar{S}_γ “обращалась” (т.е. решалась соответствующая СЛАУ) также с помощью AMG.

В таблицах 1–3 приведены результаты расчётов для $\gamma = 0$ при различных значениях временного шага: $\Delta t = 0.5$, $0.5 \cdot 10^{-3}$ и 10^{-7} . Можно заметить, что число итераций и время решения растут с увеличением временного шага. Сетка “kern” ведёт себя нетипично:

число итераций немного уменьшается при увеличении шага с 0.0005 до 0.5 секунд. Причина такого поведения неясна.

Таблицы 4 и 5 демонстрируют возможность уменьшения числа итераций за счёт оптимизации параметра регуляризации γ (при $M = I$) из выражения (13). Можно заметить, что на расчетной сетке с однородным кубом 64^3 число итераций уменьшается с отрицательной γ . Однако на сетке с искусственным керном никакого улучшения не наблюдается и параметр γ выгоднее задавать равным нулю.

В таблице 6 даны результаты расчётов при $\Delta t = 0.5$, получаемые с использованием диагональной компенсации (21) при $\theta = 1$ для аппроксимации дополнения Шура. Получаемый при этом предобуславливатель \bar{S}_γ строится без регуляризации, т.е. $\gamma = 0$, а его обращение, как и в предыдущих случаях, осуществляется с помощью алгоритма BoomerAMG.

Наконец, в таблице 7 приведены результаты полученные при решении СЛАУ без предобуславливания, но с точной безматричной аппроксимацией дополнения Шура. На сетках с пустыми кубами 32^3 , 64^3 , 128^3 наблюдается наивысшая эффективность расчета. При этом на сетке “kern” скорость счета падает во много раз.

Анализ представленных выше результатов позволяет сделать следующие предварительные выводы:

- Решение сеточных начально-краевых задач для систем уравнений Стокса представляет собой достаточно трудоёмкую проблему, особенно при больших значениях временных шагов. В частности, значение $\Delta t = 0.5$ в таблицах 1, 6 фактически соответствует стационарной задаче.
- Эффекты, наблюдаемые в таблице 7, связаны с особенностью дополнения Шура для исследуемой матрицы. Оно представляет собой дивергенцию от обратного лапласиана, примененного к градиенту, что близко к единичной матрице I . Поэтому на однородных средах наблюдается высокая эффективность. На искусственном керне большую роль играют граничные условия прилипания, т.к. по всему керну распределены непроницаемые стенки и дополнение Шура далеко от матрицы I .
- Применение регуляризации показывает положительный эффект только на пустых кубах. Это связано с приближением аппроксимации дополнения Шура к единичной матрице. Можно сделать вывод, что применение такой регуляризации неэффективно для задачи о течении в порах.

- Использование диагональной компенсации при ленточной аппроксимации обратной матрицы (алгоритм SPAI) в дополнении Шура позволяет существенно повысить производительность исследуемых методов на задачах с однородными средами, но неэффективно для пористой среды.
- Эффективность распараллеливания рассматриваемых итерационных процессов можно считать вполне удовлетворительной, хотя здесь ещё есть большие резервы для ускорения за счёт учёта особенностей алгоритмов и архитектуры многопроцессорной вычислительной системы (МВС).
- Для расчетной сетки “kepi” с пористой средой из рассмотренных вариантов наибольшую производительность показывает предобуславливатель (12) без дополнительных регуляризаций и диагональной компенсации.

5 Заключение

Рассматриваемые в работе блочные методы неполной факторизации для системы сеточных уравнений Стокса, в комбинации с применением регуляризации, многосеточных подходов, алгоритмов аппроксимации обратных матриц с диагональной компенсацией, предобусловленных крыловских итерационных процессов и технологий распараллеливания вычислений на МВС с распределённой и общей памятью представляют собой широкое поле деятельности для перспективных разработок высокопроизводительного математического и программного обеспечения для решения актуальных проблем суперкомпьютерного моделирования. Следует отметить, что задача Стокса обычно представляет собой только часть больших междисциплинарных проблем. Однако рассматриваемые в работе вопросы имеют самостоятельный характер, и полученные предварительные результаты дают хорошую перспективу для дальнейших исследований в различных направлениях. Отметим, что в данной работе по техническим причинам использовался широко распространённый решатель BoomerAMG, однако предварительные расчеты показывают возможность ускорения вычислений при переходе на алгебро-геометрический многосеточный метод описанный в п. 3. Существенные перспективы видятся при использовании многосеточных методов непосредственно для задачи Стокса [13], а также при распараллеливании алгоритмов с помощью декомпозиции областей [7].

References

- [1] M.A. Olshanskii, E.E. Tyrtysnikov, *Iterative Methods for Linear Systems: Theory and Applications*, SIAM Philadelphia, 2014.
- [2] H.C. Elman, D.J. Silvester, A.J. Wathen, *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*, Oxford University Press, New-York, 2005.
- [3] Yu.V. Bychenkov, E.V. Chizhonkov, *Iterative Methods for Solving Saddle-Point Problems*, BINOM, Moscow, 2010.
- [4] F. Brezzi, M. Fortin, *Mixed and Hybrid Finite Element Method*, Springer Series in Computational Mathematics, 1991.
- [5] D. Boffi, F. Brezzi, M. Fortin, *Finite Elements for the Stokes Problem*, Lecture Notes in Mathematics. Mixed Finite Elements, Compatibility Conditions, and Applications, Springer, Berlin, Heidelberg, **1939** (2008), 45–100.
- [6] M. Benzi, G.H. Golub, J. Liesen, *Numerical Solution of Saddle Point Problems*, Acta Numerica, **14** (2005), 1–137.
- [7] V.P. Il'in, *Iterative Preconditioned Methods in Krylov Spaces: Trends of the 21st Century*, Computational Mathematics and Mathematical Physics, **61** (2021), 1750–1775.
- [8] V.P. Il'in, *Mathematical modeling Part 1. Continuous and discrete models*, SB RAS, Novosibirsk, 2017.
- [9] R. Estrin, C. Greif, *SPMR: A Family of Saddle-Point Minimum Residual Solvers*, SIAM Journal on Scientific Computing, **40**:3 (2018), 1884–1914.
- [10] F.J. Gaspar, Y. Notay, C.W. Oosterlee, C. Rodrigo, *A Simple and Efficient Segregated Smoother for the Discrete Stokes Equations*, SIAM Journal on Scientific Computing, **36**:3 (2014), A1187–A1206.
- [11] N. Gould, D. Orban, T. Rees *Projected Krylov Methods for Saddle-Point Systems*, SIAM Journal on Matrix Analysis and Applications. **35** (2014), 1329–1343.
- [12] C. Greif, D. Schoetzau *Preconditioners for Saddle Point Linear Systems with Highly Singular (1,1) Blocks*, Electronic Transactions on Numerical Analysis, **22**, (2006), 114–121.
- [13] C.W. Oosterlee, F.J.G. Lorenz, *Multigrid Methods for the Stokes System. Computing in Science and Engineering*, **8**:6 (2006), 34–43.
- [14] Y. Notay, *Algebraic Multigrid for Stokes Equations*, SIAM Journal on Scientific Computing, **39**:5 (2017), S88–S111.
- [15] M. Benzi, G.H. Golub, J. Liesen, *Numerical solution of saddle point problems* Acta Numerica, **14** (2005), 1–137.
- [16] Ya. Gurieva, V. Il'in, *On methods of conjugate directions for multiple solution of SLAE* Zapiski POMI, **496** (2020), 26–42.
- [17] M.I. Ivanov, I.A. Kremer, M.V. Urev, *Solving the Pure Neumann Problem by a Finite Element Method*, Numerical Analysis and Applications, **12** (2019), 359–371.
- [18] M. Benzi, C.D. Meyer, M. Tũma, *A Sparse Approximate Inverse Preconditioner for the Conjugate Gradient Method*, SIAM Journal on Scientific Computing, **17**:5 (1996), 1135–1149.
- [19] V.P. Il'in, *Multigrid Incomplete Factorization Methods in Krylov Subspaces*, Journal of Mathematical Sciences, **272** (2023), 523–532.

- [20] V.P. Il'in, *Finite Element Methods and Technologies*, SB RAS, Novosibirsk, 2017.
- [21] K. Stüben, *Algebraic multigrid (AMG). An introduction with applications*, Sankt Augustin, 1999.
- [22] F. Lekien, J.E. Marsden, *Tricubic interpolation in three dimensions* International Journal for Numerical Methods in Engineering, **63**:3 (2005), 455–471.
- [23] K. Stüben, *A review of algebraic multigrid*, Journal of Computational and Applied Mathematics, **128**:1-2 (2001), 281–309.
- [24] V. Henson, U. Yang, *BoomerAMG: A parallel algebraic multigrid solver and preconditioner*, Applied Numerical Mathematics. **41**:1 (2002), 155–177.
- [25] S. Balay, W.D. Gropp, L.C. McInnes, B.F. Smith, *Efficient Management of Parallelism in Object-Oriented Numerical Software Libraries*, Modern Software Tools for Scientific Computing. Birkhäuser, Boston, MA. (1997), 163–202.
- [26] I. Konshin, I. Капырин, К. Nikitin, К. Terekhov, *Application of the parallel INMOST platform to subsurface flow and transport modelling*, In: Proc. 11th International Conference on Parallel Processing and Applied Mathematics (2015), 277–286.