

Н. И. Горбенко, В. П. Ильин

АДДИТИВНЫЙ МЕТОД ПИСМАНА–РЕЧФОРДА

§1. ВВЕДЕНИЕ

Итерационный неявный метод переменных направлений Писмана–Речфорда (ADI, Alternating Direction Implicit method) известен уже около 60 лет и исследовался многими авторами, в том числе с точки зрения его распараллеливания на многопроцессорных вычислительных системах (МВС), см., например, [1–4] и цитируемые там работы. Методов типа ADI существует достаточно много, и они в основном эквивалентны по скорости сходимости итераций. Поэтому мы ограничимся исследованием только алгоритма Писмана–Речфорда, обозначая его для краткости аббревиатурой ADI. В последние годы интерес к подобному рода алгоритмам возрос в связи с их эффективным применением к решению матричных уравнений Ляпунова и Сильвестра [5].

В применении к решению СЛАУ вида

$$Au = f, \quad A = A_1 + A_2, \quad A \in \mathcal{R}^{N,N}, \quad (1)$$

рассматриваемый метод описывается формулами

$$\begin{aligned} (I + \tau_n A_1)u^{n-1/2} &= (I - \tau_n A_2)u^{n-1} + \tau_n f, \\ (I + \tau_n A_2)u^n &= (I - \tau_n A_1)u^{n-1/2} + \tau_n f, \end{aligned} \quad (2)$$

где n, τ_n и I суть номер итерации, итерационный параметр и единичная матрица соответственно. Известны также другие, эквивалентные по скорости сходимости, неявные алгоритмы переменных направлений, которые в дальнейшем будем также обозначать ADI. Здесь и в дальнейшем предполагается, что матрицы A_1 и A_2 – вещественные и положительно полуопределенные, а одна из них – положительно определена, т.е.

$$(A_k v, v) \geq \delta_k (v, v), \quad \delta_k \geq 0, \quad \delta_1 + \delta_2 > 0, \quad v \in \mathcal{R}^n. \quad (3)$$

Ключевые слова: метод Писмана–Речфорда, оптимальная последовательность параметров, перестановочные матрицы, рациональная функция, простые дроби, параллельные алгоритмы, двумерные краевые задачи.

Работа поддержана грантом РФФИ N 14-11-00485, а также грантом РФФИ N 14-07-00128.

Методы ADI обладают рекордной скоростью сходимости итераций в случае представления исходной матрицы СЛАУ суммой двух перестановочных слагаемых с известными границами спектра, которые возникают, например, при конечно-объемной или конечно-элементной аппроксимации двумерных краевых задач с разделяющимися переменными. В этой ситуации, если известны собственные числа одной из матриц $A_k, k = 1, 2$, методы ADI, при соответствующем выборе итерационных параметров τ_n , дают точное решение СЛАУ за N_k итераций (в предположении отсутствия погрешностей округления), где N_k – число узлов сетки в одном из координатных направлений, $N = N_1 N_2$. Точнее говоря, число итераций равно $\bar{N}_k \leq N_k$ – количеству различных собственных чисел матрицы A_k . Отметим, что оптимальная последовательность параметров τ_n , с точки зрения достижения заданной точности ε за минимальное число итераций $n(\varepsilon)$, в смысле выполнения условия для эвклидовой нормы невязки

$$\|r^n\| \equiv \|f - Au^n\| \leq \varepsilon \|f\|, \quad \varepsilon \ll 1, \quad (4)$$

определяется из решения задачи Золотарева [1] и характеризуется неравенством

$$n(\varepsilon) \leq C \log \varepsilon^{-1} \log(\text{cond}(A)), \quad (5)$$

где C – постоянная и $\text{cond}(A)$ – некоторая оценка числа обусловленности матрицы A . Строго говоря, если матрицы A_1, A_2 имеют различные границы спектра, то оптимальные значения итерационных параметров в (2) имеют разные значения $\tau_n^{(1)}$ и $\tau_n^{(2)}$ для первой и второй строк соответственно. Но если $\text{cond}(A_1), \text{cond}(A_2)$ являются величинами одного порядка по N , то формулы (2) позволяют получить “почти оптимальную”, или оптимальную по порядку, последовательность параметров τ_n , и мы для простоты ограничимся рассмотрением этого случая. Поскольку, например, для сеточных аппроксимаций эллиптических уравнений второго порядка (будем для краткости называть их модельными СЛАУ) имеем $\text{cond}(A) \sim O(N)$, соответствующее число итераций оказывается пропорциональным $\log N$. При этом методы ADI конкурентны с алгоритмом быстрого преобразования Фурье (БПФ), и конкретные соотношения их производительности на МВС зависят от свойств решаемой задачи, а также от технологий используемых программных параллельных реализаций.

§2. НЕКОТОРЫЕ СВОЙСТВА МЕТОДА ADI

После исключения в соотношениях (2) вспомогательного вектора $u^{n-1/2}$ рассматриваемый итерационный процесс можно записать в виде

$$(s_n I + A_1)(s_n I + A_2)u^n = (s_n I - A_1)(s_n I - A_2)u^{n-1} + 2s_n f, \quad (6)$$

где новый введенный параметр $s_n = \tau_n^{-1}$ играет роль сдвига спектра для матриц A_1, A_2 . Определим предобуславливающую матрицу

$$B_n = \frac{1}{2s_n}(s_n I + A_1)(s_n I + A_2), \quad (7)$$

тогда формулу (6) можно привести к виду

$$u^n = u^{n-1} + B_n^{-1}r^{n-1}, \quad (8)$$

откуда получаем следующие соотношения для векторов невязок:

$$\begin{aligned} r^n &= r^{n-1} - AB_n^{-1}r^{n-1} = (B_n - A)B_n^{-1}r^{n-1} \\ &= (s_n I - A_1)(s_n I - A_2)(s_n I + A_2)^{-1}(s_n I + A_1)^{-1}r^{n-1}. \end{aligned} \quad (9)$$

Предполагая теперь выполненным свойство перестановочности матриц $A_1 A_2 = A_2 A_1$ (которое пока никак не использовалось), из (8) получаем формулы

$$r^n = R_n(A_1)R_n(A_2)r^{n-1} \equiv T_n r^{n-1}, \quad (10)$$

в которых введены обозначения

$$\begin{aligned} R_n(A_k) &= P_n(A_k)Q_n^{-1}(A_k), \quad P_n(A_k) = s_n I - A_k, \\ Q_n(A_k) &= (s_n I + A_k), \quad k = 1, 2, \end{aligned} \quad (11)$$

а T_n есть матрица перехода итерационного процесса. Из (9), (10) следует выражение

$$\begin{aligned} r^n &= \bar{R}_n(A_1)\bar{R}_n(A_2)r^0, \\ \bar{R}_n(A_k) &= \bar{P}_n(A_k)\bar{Q}_n^{-1}(A_k) = \prod_{l=1}^n P_l(A_k) \prod_{l=1}^n Q_l^{-1}(A_k), \end{aligned} \quad (12)$$

где в скалярном случае $\bar{R}_n(s)$ представляет собой рациональную функцию вида

$$\bar{R}_n(s) = \frac{\bar{P}_n(s)}{\bar{Q}_n(s)} \equiv \frac{(s_1 - s) \cdots (s_n - s)}{(s_1 + s) \cdots (s_n + s)}. \quad (13)$$

Отметим, что если векторы невязок r^0, r^1, \dots, r^{n-1} известны, то векторы последовательных приближений к решению могут быть вычислены с помощью формулы, которая следует из (8):

$$u^n = u^0 + B_1^{-1}r^0 + \dots + B_n^{-1}r^{n-1}. \quad (14)$$

Вместо (13) можно также использовать соотношения, не содержащие векторы невязок. Для этого приведем равенство к виду

$$u^n = T_n u^{n-1} + g_n, \quad g_n = 2s_n Q_n(A_1)Q_n(A_2)f, \quad (15)$$

из которого следует выражение

$$u^n = T_{n,1}u^0 + T_{n,2}g_1 + \dots + T_{n,n}g_{n-1} + g_n, \quad (16)$$

в котором введены обозначения

$$T_{n,l} = T_{n,l}^{(1)}T_{n,l}^{(2)}, \quad T_{n,l}^{(k)} = \prod_{m=l}^n (s_m I + A_k)^{-1}(s_m I - A_k). \quad (17)$$

§3. АДДИТИВНЫЙ ИТЕРАЦИОННЫЙ МЕТОД ПИСМАНА–РЕЧФОРДА

Как видно из предыдущего пункта, реализация алгоритма ADI может быть выражена с помощью рациональных матричных функций вида (12), операции с которыми, с точки зрения распараллеливания, целесообразно выполнить на основе разложения на простые дроби, которые будем искать в следующем виде:

$$\bar{R}_n(A) \equiv \prod_{l=1}^n (s_l I - A)(s_l I + A)^{-1} = z_0 I + \sum_{l=1}^n z_l R_l(A). \quad (18)$$

Здесь s_1, \dots, s_n – предполагаемый заданным набор чисел, z_1, \dots, z_n – искомые коэффициенты разложения, а индекс “ k ” для краткости опущен. Умножая обе части (18) на матричный многочлен $\bar{Q}_n(A)$, приходим к равенству

$$\begin{aligned} \bar{P}_n(A) &= (s_1 I - A) \cdots (s_n I - A) = z_0 \bar{Q}_n(A) + \sum_{l=1}^n z_l \bar{S}_l(A), \\ \bar{S}_l(A) &= \prod_{k=l}^n (t_{k,l} I + A), \end{aligned} \quad (19)$$

где ради краткости используется обозначение: $t_{k,l} = -s_l$ при $k = l$ и $t_{k,l} = s_l$ в противном случае. Приравнявая в (19) коэффициенты при

одинаковых степенях A , получаем систему из n линейных алгебраических уравнений относительно неизвестных z_l , $l = 1, \dots, n$.

Предварительно запишем матричные многочлены $\bar{P}_n(A)$, $\bar{Q}_n(A)$, $\bar{S}_k(A)$ n -го порядка в виде сумм по степеням матрицы A :

$$\begin{aligned}\bar{P}_n(A) &= b_0 I - b_1 A + \dots + (-1)^n b_n A^n, \\ \bar{Q}_n(A) &= b_0 I + b_1 A + \dots + b_n A^n, \\ \bar{S}_l(A) &= b_{0,l} I + b_{1,l} A + \dots + b_{n,l} A^n.\end{aligned}\tag{20}$$

Участвующие здесь коэффициенты описываются формулами

$$\begin{aligned}b_n = 1, \quad b_{n-1} &= \sum_{k=1}^n s_k, \quad b_{n-k} = \sum_{1 \leq i_1 < \dots < i_k \leq n} \prod_{j=1}^k s_{i_j}, \quad b_0 = s_1 \dots s_n, \\ b_{n,l} = 1, \quad b_{n-1,l} &= \sum_{i=1}^n t_{l,i}, \quad b_{n-k,l} = \sum_{1 \leq i_1 < \dots < i_k \leq n} \prod_{j=1}^k t_{l,i_j}, \quad b_{0,l} = t_{1,l} \dots t_{n,l}.\end{aligned}$$

Из определения величин $t_{l,k}$ следует, что при всех $l = 1, \dots, n$ имеет место равенство $b_{0,l} = -b_0$. Далее из (19) получаем

$$\begin{aligned}b_0 z_0 - \sum_{l=1}^n b_{0,l} z_l &= b_0, \\ b_k z_0 - \sum_{l=1}^n b_{k,l} z_l &= (-1)^k b_k, \quad k = 1, \dots, n.\end{aligned}\tag{21}$$

Из системы (21) для удобства можно исключить z_0 , после чего получаем

$$\begin{aligned}z_0 &= 1 - \sum_{l=1}^n z_l, \\ \sum_{l=1}^n \left(b_k + \sum_{l=1}^n b_{k,l} \right) z_l &= [(-1)^k - 1] b_k, \quad k = 1, \dots, n.\end{aligned}$$

Таким образом, для вектора неизвестных $z = \{z_k\}$ имеем СЛАУ n -го порядка

$$\begin{aligned}C z &= g, \\ C &= \{c_{k,l} = 1 + b_{k,l}/b_k\}, \quad g = \{g_k = (-1)^k - 1\}.\end{aligned}\tag{22}$$

Если в соответствии с формулами (18)–(22) найти коэффициенты $z_0^{(k)}, z_1^{(k)}, \dots, z_n^{(k)}$, $k = 1, 2$, для разложений рациональных функций $\bar{R}_n(A_1)$ и $\bar{R}_n(A_2)$, то для вектора невязки r^n из (12) получим формулу

$$r^n = \left[z_0^{(1)} I + \sum_{l=1}^n z_l^{(1)} R_l(A_1) \right] \left[z_0^{(2)} I + \sum_{l=1}^n z_l^{(2)} R_l(A_2) \right] r^0. \quad (23)$$

Можно представить и другой способ представления рациональной функции простых дробей. Для этого вместо (18) запишем равенство

$$n\bar{R}_n(A) = \sum_{l=1}^n (\hat{z}_l I - A)(s_l I + A)^{-1}, \quad (24)$$

где \hat{z}_i – новые искомые коэффициенты.

В этом случае вместо (22) для коэффициентов y_i из (24) получаем систему уравнений

$$\hat{C}\hat{z} = \hat{g}, \quad \hat{z} = \{\hat{z}_i\}, \quad (25)$$

в которой элементы матрицы \hat{C} и вектора правой части \hat{g} определяются соотношениями

$$\hat{C} = \{\hat{C}_{i,j} = \sum_{k=1}^{n-1} b_{n-1,k}(\hat{t}_{1,i}, \dots, \hat{t}_{j,i}, \dots, \hat{t}_{n-1,i})\},$$

$$\hat{g} = \{\hat{g}_i = (-1)^i n b_{n,i}(s_1, \dots, s_n) - \sum_{j=1}^n b_{n,j}(\check{t}_{1,i}, \dots, \check{t}_{j,i}, \dots, \check{t}_{n,i})\}.$$

Здесь введены обозначения

$$\hat{t}_{j,i} = \begin{cases} s_i & \text{при } i < j, \\ s_{i+1} & \text{при } i \neq j, \end{cases} \quad \check{t}_{j,i} = \begin{cases} 0 & \text{при } i = j, \\ s_j & \text{при } i \neq j. \end{cases}$$

Рассмотренные в данном параграфе формулы (18)–(25) предполагают, что итерационные параметры s_1, \dots, s_n известны. На вопросах их выбора мы останавливаться не будем, так как они хорошо освещены в литературе. Подчеркнем только делаемое нами предположение, что величины сдвигов s_k и число итераций $n(\varepsilon)$ определяются критерием останова ε и границами спектра матрицы A , $0 < m \leq \lambda(A) \leq M < \infty$, в соответствии с формулами (4), (5), где можно положить $\text{cond}(A) = M/m$.

После нахождения векторов невязок для вычисления последовательных приближений u^n можно воспользоваться соотношениями (14) или

(16). Подчеркнем, что при этом требуется предварительно определить в первом случае векторы r^0, \dots, r^{n-1} , а во втором — g_1, \dots, g_n .

В рассматриваемых методах ADI мы не будем применять итерационные процессы в подпространствах Крылова, так как это не должно принципиально изменить соотношение эффективности распараллеливания мультипликативного алгоритма (2) и соответствующих аддитивных версий, анализ которых является основной целью данной работы.

Заметим также, что для систем уравнений (22) и (25) относительно коэффициентов разложений в простые дроби при малых порядках матрицы C и \hat{C} оказываются невырожденными, если параметры сдвигов s_k различны. Однако в общем случае вопросы разрешимости и устойчивости этих систем требуют дополнительных исследований.

§4. ОЦЕНКИ ЭФФЕКТИВНОСТИ РАСПАРАЛЛЕЛИВАНИЯ ADI

Анализ эффективности распараллеливания различных вариантов реализации метода Писмана–Речфорда мы будем для простоты проводить на примере модельной СЛАУ, получаемой из пятиточечной конечно-разностной аппроксимации двумерной задачи Дирихле для уравнения Пуассона в квадратной расчетной области на квадратной же сетке с шагом h , см. подробнее [1–4]. При этом матрицы A_1 и A_2 являются положительно определенными, блочно-трехдиагональными (для соответствующей упорядоченности неизвестных), а порядок системы равен $N = N_1 N_2 = h^{-2}$, $N_1 = N_2 = h^{-1}$, где N_1 и N_2 — количество узлов сетки по каждому из координатных направлений.

Сравнение производительности разных алгоритмов мы ради простоты будем проводить для абстрактной компьютерной системы с большой общей памятью, достаточной для решения рассматриваемых задач в технологии многопоточковых вычислений без обменов и конфликтов с доступом к данным. Можно предположить, например, что реализация алгоритма осуществляется на одном центральном или графическом вычислительном устройстве (CPU или GPGPU) со значительным количеством ядер.

Время T_m выполнения мультипликативного метода ADI по формулам (2), которое будем мерить в машинных тактах, временная протяженность которых определяется частотой МВС, можно оценить выражением

$$T_m = 2(T_s + T_r) n(\varepsilon). \quad (26)$$

Здесь T_s есть количество тактов для решения СЛАУ с матрицами вида $I + \tau A_k$, а T_r – длительность вычисления правых частей в каждом из уравнений (2).

Величины T_s и T_r можно оценить с помощью формул

$$T_s = C_s/P, \quad T_r = C_r/P, \quad (27)$$

где P – количество организуемых вычислительных потоков (считаем, что число узлов сетки в одном измерении кратно P , т.е. $N^{1/2}/P$ есть целое число), а C_s и C_r – некоторые значения, определяемые конкретным применяемым алгоритмом. Например, трехдиагональные СЛАУ могут решаться с помощью “стандартного” метода прогонки или спектральным алгоритмом. Предполагается, что величины C_s и C_r одинаковы как для мультипликативного, так и для аддитивного ADI, так что с точки зрения их сравнения, конкретные значения C_s и C_r безразличны.

При выводе (27) учитывается, что решение блочно-трехдиагональных СЛАУ фактически сводится к одновременному решению независимых трехдиагональных систем по линиям сетки: сначала по первому координатному направлению, а затем по второму. На каждом из этих итерационных полуэтапов распараллеливание делается естественным (и практически оптимальным) образом, так что общее количество тактов на решение СЛАУ методом (2) равно

$$T_m = 2(C_s + C_r) n(\varepsilon)/P, \quad (28)$$

т.е. при $P \ll N^{1/2}$ ускорение является линейным по P .

Отметим еще следующее: если число потоков P больше количества узлов в одном измерении и, например, кратно $N^{1/2}$, то величины C_s и C_r могут быть уменьшены за счет “внутреннего” распараллеливания арифметических операций для каждой линии сетки (но это, опять же, можно сделать одинаковым образом для мультипликативного и аддитивного ADI, не изменяя соотношения их сравнительной эффективности).

Аддитивный вариант ADI позволяет достичь большей степени распараллеливания (при наличии достаточного количества ядер) за счет синхронного выполнения векторно-матричных умножений при реализации формул (23) и (14). При этом будем иметь в виду, что вычисление коэффициентов z_k разложения рациональной матричной функции

в простые дроби делается до итераций один раз, а объем соответствующих арифметических операций не зависит от h , и им можно пренебречь.

Решение СЛАУ аддитивным методом ADI может быть представлено двумя этапами: вычисление векторов невязок по формуле (23) и нахождение последовательных приближений u^n из (14). В первом случае наиболее ресурсоемкой операцией является двукратное вычисление векторно-матричной суммы вида

$$w = \sum_{l=1}^n z_l (s_l I - A_k) (s_l I + A_k)^{-1} v.$$

Таким образом, время выполнения формулы (23) оценивается величиной

$$T_a^{(1)} \cong 2[T_s + T_1 + C_1 \log_2 n(\varepsilon)], \quad (29)$$

где T_s есть та же величина, что и в (26), T_1 – длительность умножения на $z_l (s_l I - A_k)$, а последнее слагаемое – это затраты на суммирование методом сдвигания на n ядрах. Отметим, что для быстрого определения приближений u^n требуется знание всех векторов r^k , $k = 0, 1, \dots, n$, но они могут быть найдены одновременно за то же время $T_a^{(1)}$, но при этом необходима организация большего количества потоков, т.е. $P \geq N^{1/2} n(\varepsilon)$.

Реализация формулы (14) осуществляется за время

$$T_a^{(2)} \cong T_s + T_1 + C_1 \log_2 n(\varepsilon). \quad (30)$$

А поскольку полное время выполнения аддитивного метода равно $T_a = T_a^{(1)} + T_a^{(2)}$, отсюда получаем, что ускорение за счет применения аддитивного ADI пропорционально отношению $n / \log_2 n$. Разумеется, применение такого алгоритма на одном или малом числе потоков нецелесообразно, так как соответствующий объем необходимых вычислительных ресурсов (и арифметических операций, и памяти) значительно возрастает. Выше в формуле (29) мы использовали первый из рассмотренных вариантов разложения рациональной функции в сумму простых дробей, однако очевидно, что для второго варианта разложения по формулам (24), (25) результат качественно не меняется.

§5. ЗАКЛЮЧЕНИЕ

В данной работе предложен подход к повышению производительности программных реализаций параллельных методов ADI, при возможности увеличения количества используемых вычислительных потоков над общей памятью, за счет разложения матричной рациональной функции в сумму простых дробей. При этом итерационные параметры τ_n (или $\tau_n^{(1)}, \tau_n^{(2)}$), а также скорость сходимости итераций в случае точной арифметики остаются теми же, что и в классическом алгоритме Писмана–Речфорда.

Для сеточных модельных СЛАУ, когда $n(\varepsilon) \sim \log h$, коэффициент ускорения за счет перехода от мультипликативного метода к аддитивному, при оптимальных значениях итерационных параметров, равен $\log h^{-1} / \log(\log h^{-1})$, а при неоптимальных он повышается.

Отметим, что если в (1) векторы u, f имеют компонентными значения сеточных функций в узлах прямоугольной регулярной сетки, т.е.

$$u = \{u_{i,j}\}, \quad f = \{f_{i,j}\}, \quad i = 1, \dots, N_1; \quad j = 1, \dots, N_2,$$

а матрицы A_1, A_2 соответствуют модельной СЛАУ и представимы в блочно-диагональном виде

$$A_1 = \text{blockdiag}\{\bar{A}_1\}, \quad A_2 = \text{blockdiag}\{\bar{A}_2\},$$

где блоки \bar{A}_1 и \bar{A}_2 – трехдиагональные матрицы порядков N_1 и N_2 , то исходная алгебраическая система может быть записана в форме следующего матричного уравнения Сильвестра:

$$\begin{aligned} \bar{A}_1 U + U \bar{A}_2 &= F, \\ \bar{A}_1 \in \mathcal{R}^{N_1, N_1}, \quad \bar{A}_2 \in \mathcal{R}^{N_2, N_2}; \quad U, F &\in \mathcal{R}^{N_1, N_2}. \end{aligned} \quad (31)$$

Как показано в [5] и цитируемых там работах, если матрица правой части (31) представлена в факторизованном виде

$$F = F_1 F_2^T, \quad F_k \in \mathcal{R}^{N_k, m}, \quad m \ll N_k, \quad k = 1, 2,$$

то для решения (31) применим следующий чрезвычайно экономичный “факторизованный” метод fADI (индекс “T” означает транспонирование):

$$\begin{aligned} U^n &= U_1^n D^n (U_2^n)^T, \quad U_1^n = (U_1^1, \dots, U_1^n) \in \mathcal{R}^{N_1, mn}, \\ D &= \text{diag}((s_1^{(1)} + s_1^{(2)})I_m, \dots, (s_n^{(1)} + s_n^{(2)})I_m) \in \mathcal{R}^{mn, mn}, \\ U_2^n &= (U_2^1 \dots U_2^n) \in \mathcal{R}^{N_2, mn}, \end{aligned} \quad (32)$$

где $s_n^{(1)}$ и $s_n^{(2)}$ – предполагаемые заданными итерационные параметры (сдвиги спектра матриц \bar{A}_1 и \bar{A}_2), I_m – единичная матрица порядка m , а последовательные приближения для матричных множителей $U_1^{(k)} \in \mathcal{R}^{N_1, m}$ и $U_2^{(k)} \in \mathcal{R}^{N_2, m}$, $k = 1, \dots, n$, вычисляются из рекуррентных соотношений

$$\begin{aligned} U_1^{(0)} &= (A_1 + s_0^{(1)}I)^{-1}F_1, \quad U_1^{(n)} = \prod_{k=1}^n \hat{R}_k(A_1)U_1^{(0)}, \\ U_2^{(0)} &= (F_2)^T(A_2 + s_0^{(2)}I), \quad U_2^{(n)} = (U_2^{(0)})^T \prod_{k=1}^n \check{R}_k(A_2), \\ \hat{R}_k(z) &= (z - s_{k-1}^{(1)})/(z + s_k^{(2)}), \quad \check{R}_k(z) = (z - s_{k-1}^{(2)})/(z + s_k^{(1)}). \end{aligned} \quad (33)$$

Поскольку в этих формулах произведения матричных рациональных функций $\hat{R}_k(A_1)$ и $\check{R}_k(A_2)$ могут быть аналогично предыдущему представлены в виде суммы простых дробей, то для факторизованного метода (32), (33) также несложно строится параллелизуемая аддитивная версия fADI. Отметим также, что вычисление величин $U_1^{(n)}$ и $U_2^{(n)}$ в формулах (33) может проводиться независимо и одновременно.

ЛИТЕРАТУРА

1. А. А. Самарский, Е. С. Николаев, *Методы решения сеточных уравнений*, Наука, М., 1978.
2. В. П. Ильин, *Методы конечных разностей и конечных объемов для эллиптических уравнений*, Новосибирск, Изд-во ИВМ СО РАН, 2000.
3. В. П. Ильин, *Методы и технологии конечных элементов*, Новосибирск, изд. ИВМиМГ СО РАН, 2007.
4. В. П. Ильин, Д. В. Кныш, *Параллельные алгоритмы решения разделяющихся краевых задач*, Междун. научная конф. “Параллельные вычислительные технологии,” Санкт-Петербург, Изд-во Политехн. ун-та (СПбПУ), 2008, с. 107–118.
5. P. Benner, Ren-Cang Li, Nin Truhar, *On the ADI method for Sylvester equations.* — J. Comput. Appl. Math. **233** (2009), 1035–1045.

Gorbenko N. I., Il'in V. P. The additive Peaceman–Rachford method.

A new version of the parallel alternating direction implicit (ADI) method by Peaceman and Rachford for solving systems of linear algebraic equations with positive-definite coefficient matrices represented as sums of two commuting terms is suggested. The algorithms considered are suited for solving two-dimensional grid boundary-value problems with separable variables, as well as the Sylvester and Lyapunov matrix equations. The approach to parallelization speed up proposed in the paper is based on representing rational functions as sums of partial fractions. An additive version of the factorized ADI method for solving Sylvester's equation is described. Estimates of the speed up obtained by increasing the number of computational units are presented. These estimates demonstrate a potential advantage of using the additive algorithms when implemented on a supercomputer with large number of processors or cores.

Институт вычислительной математики и
математической геофизики СО РАН,
Новосибирский государственный университет
E-mail: ilin@sscc.ru

Поступило 23 ноября 2015 г.