

УДК 519.632.4

А.А. Фомин, Л.Н. Фомина

УСКОРЕНИЕ ПОЛИНЕЙНОГО РЕКУРРЕНТНОГО МЕТОДА В ПОДПРОСТРАНСТВАХ КРЫЛОВА

На примере алгоритма LR1 полинейного рекуррентного метода [1, 2] рассматриваются два механизма его ускорения в подпространствах Крылова. В качестве ускоряющего метода используется алгоритм Bi-CGStab P ван дер Ворста. Показано, что традиционный подход: построение предобуславливателя на базе алгоритма LR1, не приводит к требуемому результату. В то время как прямое сочетание алгоритмов LR1 и Bi-CGStab P позволяет значительно повысить скорость сходимости решения.

Ключевые слова: *разностные эллиптические уравнения, итерационный метод решения, подпространства Крылова, полинейный рекуррентный метод.*

Несмотря на бурный рост вычислительной техники и теоретические достижения в области решения систем линейных алгебраических уравнений [3], возникающих при моделировании многомерных явлений математической физики, проблема создания и развития быстродействующих алгоритмов не теряет своей актуальности. Происходит это потому, что с усложнением поставленных задач возрастают и требования к точности и надежности методов решения.

Современные методы решения задач математической физики зачастую сводятся к разностной аппроксимации многомерных дифференциальных уравнений [4], что, в свою очередь, приводит к построению системы линейных алгебраических уравнений (СЛАУ), матрица которой имеет большую размерность и разреженно-упорядоченную структуру. Для многомерных эллиптических по пространству задач до сих пор не удалось разработать прямой, экономичный, устойчивый к ошибкам округления метод, способный решать СЛАУ с линейной трудоемкостью относительно числа неизвестных, наподобие того, как это было сделано для одномерного случая. В настоящее время наиболее перспективными можно считать те направления разработки новых методов решения СЛАУ, в которых на уровне алгоритма учитывается фундаментальное свойство краевых эллиптических задач обязательной чувствительности решения в каждой точке области определения задачи возмущения в любой иной, включая граничную, точке. Такая чувствительность обеспечивается современными градиентными методами [3, 5], но не напрямую от точки к точке, а опосредованно, через коэффициенты, полученные при минимизации норм соответствующих функционалов. Однако непосредственно градиентные методы не характеризуются высокими скоростями сходимости, и для их ускорения используются предобуславливатели, которые строятся, как правило, на основе известных релаксационных методов [3]. При этом, чем более эффективен исходный релаксационный метод, тем более эффективным получается сочетание соответствующего предобуславливателя с градиентным методом. Например, использование в методе бисопряженных градиентов со стабилизацией предобуславливателя на базе явного метода Булеева позволяет реализовать более высокие скорости сходимости решения по сравнению с предобуславливателем на базе метода Гаусса – Зейделя [2, 6].

Постановка задачи

Пусть имеет место двумерная по пространству краевая задача в единичной области $\Omega = \{(x, y): 0 \leq x \leq 1, 0 \leq y \leq 1\}$. Внутри области Ω поведение искомой функции $\Phi(x, y)$ описывается дифференциальным уравнением

$$U \frac{\partial \Phi}{\partial x} + V \frac{\partial \Phi}{\partial y} = \frac{\partial}{\partial x} \left(v^x \frac{\partial \Phi}{\partial x} \right) + \frac{\partial}{\partial y} \left(v^y \frac{\partial \Phi}{\partial y} \right) + S, \quad (1)$$

где U, V – компоненты скорости, v^x, v^y – коэффициенты переноса, S – источник. На границе области Γ имеют место условия первого рода

$$\Phi|_{\Gamma} = \varphi. \quad (2)$$

Здесь U, V, v^x, v^y – непрерывные дифференцируемые функции внутри и на границе Ω , причем $v^x > 0, v^y > 0$; φ – непрерывная функция, определенная на границе Ω . Выбирая произвольные выражения для функций Φ, U, V, v^x, v^y , можно всегда, путем вычисления источника S из (1) и определения φ из (2) по значениям Φ на границе Ω , сформулировать необходимую для тестирования численного метода задачу. В настоящей работе рассматриваются две такие тестовые задачи.

Задача 1. Решение и коэффициенты задаются в виде следующих зависимостей:

$$\begin{aligned} \Phi(x, y) &= 256[xy(1-x)(1-y)]^2; \quad U=V=0; \\ v^x(x, y) &= 1 + 2[(x-0,5)^2 + (y-0,5)^2], \\ v^y(x, y) &= 1 + 2[0,5 - (x-0,5)^2 - (y-0,5)^2], \end{aligned} \quad (3)$$

а на границе $\Phi|_{\Gamma} = 0$.

Задача 2. Решение и коэффициенты задаются в виде следующих зависимостей:

$$\begin{aligned} \Phi(x, y) &= \exp(-10l^2) \cos(8\pi l^2); \\ v^x(x, y) &= v^y(x, y) = \exp(-5l^2); \\ l^2 &= x^2 + y^2; \quad V(x, y) = y^3/(1+x^2); \end{aligned} \quad (4)$$

на левой границе $U(0, y) = 0$, а внутри области Ω и на оставшихся границах $U(x, y)$ рассчитывается из соотношения $\frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} = 0$. На границе Ω решение Φ определяется из соотношения (4) при соответствующем выборе l : $l(0, y) = y^2$, $l(1, y) = 1 + y^2$, $l(x, 0) = x^2$, $l(x, 1) = 1 + x^2$.

Системы линейных алгебраических уравнений вида $A\Phi = b$ получаются путем разностной аппроксимации задач (3) и (4) методом контрольного объема (вариант экспоненциальной схемы) [7] на равномерной сетке, покрывающей расчетную область Ω . В качестве зависимости коэффициентов схемы от сеточного числа Пекле P_h используется рекомендованная С. Патанкармом функция $f(P_h) = \max(0, (1 - -0,1P_h)^5)$. Как отмечается в [7], получаемая при этом разностная схема

$$a_{P_{ij}} \Phi_{ij} = a_{E_{ij}} \Phi_{i+1,j} + a_{W_{ij}} \Phi_{i-1,j} + a_{N_{ij}} \Phi_{i,j+1} + a_{S_{ij}} \Phi_{i,j-1} + b_{ij}, \quad (5)$$

консервативна и имеет второй порядок аппроксимации. Причем в данном случае $a_P = a_E + a_W + a_S + a_N$, поскольку уравнение (1) стационарно и источник S не за-

висит явным образом от Φ . Здесь $1 < i < n, 1 < j < m$, где n, m – количество узлов сеточного разбиения расчетной области по координатам x, y соответственно. Из вида уравнения (5) следует, что матрица A системы имеет пятидиагональную структуру (см. рис. 1), причем для задачи (3) матричный оператор A будет самосопряжен, а для задачи (4) – нет. Следует отметить, что в общем случае способ разностной аппроксимации исходной дифференциальной задачи не имеет принципиального значения при условии, что получаемая при этом матрица СЛАУ имеет пятидиагональную структуру и положительный тип [1].

Все расчеты проводились на вычислительной системе Intel Core i5 750 2.66GHz, RAM 4Gb, Win32, IVF v.11. Итерационный процесс прекращался при выполнении условия $\|R^k\|_2 / \|R^0\|_2 < \varepsilon$, где R^0 и R^k соответственно начальная и текущая невязки, а ε – заданная точность решения. Во всех расчетах область Ω покрывалась равномерной сеткой 1001×1001 , точность ε принималась равной 10^{-8} , а начальное приближение решения бралось в виде единичного вектора $\Phi_0 = 1$. Оценка числа обусловленности матрицы системы для первой задачи составила $M_A \approx 5,90 \times 10^5$, а для второй – $M_A \approx 9,40 \times 10^6$.

Кроме нормы невязки в итерационном процессе также контролировались отношения норм текущей погрешности к начальной $\|Z^k\|_2 / \|Z^0\|_2$ и средняя скорость сходимости Q^k [4].

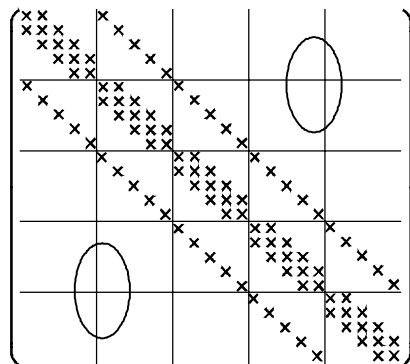


Рис. 1

Градиентный метод решения СЛАУ

Среди многочисленных итерационных градиентных методов решения разностных СЛАУ наиболее эффективным по праву считается метод бисопряженных градиентов со стабилизацией ван дер Ворста Bi-CGStab, основанный на процедуре биортогонализации Ланцоша в подпространствах Крылова [3]. Алгоритм этого метода для случая использования предобуславливателя имеет следующий вид (модификация Bi-CGStab P) [5].

1. Φ_0 – вектор начального приближения решения;
2. $r_0 = b - A \Phi_0$;
3. $\bar{r}_0$ – произвольный вектор, такой, что $(\bar{r}_0, r_0) \neq 0$, например $\bar{r}_0 = r_0$;
4. $\rho_0 = \alpha = \omega_0 = 1$;
5. $v_0 = p_0 = 0$;
6. для $k = 1, 2, 3, \dots$
7. $\rho_k = (\bar{r}_0, r_{k-1})$; $\beta = (\rho_k / \rho_{k-1})(\alpha / \omega_{k-1})$;
8. $p_k = r_{k-1} + \beta(p_{k-1} - \omega_{k-1}v_{k-1})$;
9. Определение вектора u из решения системы $Ku = p_k$;
10. $v_k = Au$;
11. $\alpha = \rho_k / (\bar{r}_0, v_k)$;

12. $s = r_{k-1} - \alpha v_k$;
13. Определение вектора z из решения системы $Kz = s$;
14. $t = Az$;
15. $\omega_k = (t, s) / (t, t)$;
16. $\Phi_k = \Phi_{k-1} + \alpha y + \omega_k z$;
17. Если Φ_k достигло требуемой точности – выход из цикла;
18. $r_k = s - \omega_k t$;
19. Конец цикла по k .

Здесь K – предобуславливатель, причем предполагается разложение $K = K_1 K_2$, где K_1 – нижняя, а K_2 – верхняя треугольные матрицы. При наличии такого разложения строки 9 и 13 алгоритма Bi-CGStab P выполняются с линейной трудоемкостью относительно числа неизвестных.

Предобуславливатель на базе алгоритма LR1

Следуя общей идеологии ускорения градиентного метода с помощью соответствующего предобуславливателя, необходимо произвести неполную факторизацию матрицы A системы уравнений на сомножители K_1 и K_2 таким образом, чтобы в основе этого разложения лежал алгоритм LR1. Из описания полинейного рекуррентного метода следует, что сомножитель K_2 в виде четырехдиагональной почти верхнетреугольной матрицы присутствует в самом алгоритме [8]. Таким образом остается только определиться с сомножителем K_1 . Структура этого сомножителя легко выводится из расчетных формул преобразования правых частей уравнения системы [1], аналогично тому, как это делается в прямом методе Гаусса решения СЛАУ, который, как известно, эквивалентен LU-разложению матрицы системы с последующим последовательным решением двух систем с нижней и верхней треугольной матрицами. На рис. 2 представлены структуры матрицы K_1 (слева) и матрицы K_2 (справа).

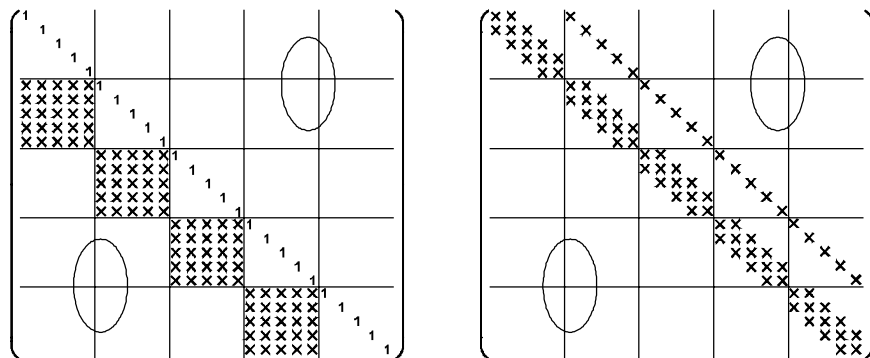


Рис. 2

Нетрудно видеть, что матрица K_1 имеет две особенности: единичную главную диагональ и полностью заполненные клетки, прилегающие к главной диагонали снизу. Вторая особенность является критичной, поскольку в этом случае трудоемкость метода будет не линейной, а пропорциональной $N^{3/2}$, где $N = n \times m$ – количество неизвестных в системе. Для преодоления этого осложнения можно восполь-

зоваться оценкой порядка величин, заполняющих клетки под главной диагональю. Анализ расчетных формул [1] показывает, что на главной диагонали «поддиагональных» клеток стоят величины порядка $1/3$, на прилегающих к ней сверху и снизу диагоналях – $(1/3)^2$, на следующих диагоналях – $(1/3)^3$ и так далее.

Аналогично тому, как это делается, например в [4, 9], в данном случае удобно в «поддиагональных» клетках выделить диагональную ленту шириной $2l_0+1$, элементы которой заметно отличны от нуля, а всеми остальными малыми элементами этих клеток пренебречь. Легко проверить, что уже при $l_0 = 10$ порядок величин на краю ленты будет 10^{-5} , а с другой стороны, даже при обычном сеточном разбиении для двумерных задач порядка 100×100 лента шириной в 21 элемент не будет серьезным обременением для вычислительного процесса.

В дальнейшем для удобства алгоритм метода бисопряженных градиентов с предобуславливателем на базе LR1 называется LR1bCGs. Для более детального анализа влияния полуширины ленты l_0 на скорость сходимости LR1bCGs решались задачи 1 и 2. Результаты решения первой задачи показали, что минимум количества итераций (33 – 35), необходимых для достижения заданной точности, приходится на диапазон $10 \leq l_0 \leq 18$; а для второй задачи минимум количества итераций составил 100 – 110 при $5 \leq l_0 \leq 17$. С дальнейшим ростом l_0 количество итераций вновь начинало возрастать. Объясняется это очевидно тем, что вновь привлекаемые к расчету диагонали уже не влияют на сходимость в силу малости расположенных на них величин, а с другой стороны, дополнительные ошибки округления все более эффективно тормозят процесс сходимости. Примечательно, что увеличение числа диагоналей в области минимально постоянного числа итераций практически не увеличивает время расчета задачи (15 – 16 с для первой задачи и 40 – 47 с для второй), что говорит в пользу приема усечения полностью заполненных «поддиагональных» клеток до относительно не широкой ленты.

Прямое сочетание алгоритмов LR1 и Bi-CGStab P

Решение задач 1 и 2 при тех же условиях напрямую алгоритмом LR1 продемонстрировало сходимость решения в первой задаче за 26 итерация (9 с), а во второй – за 101 итерацию (33 с). С одной стороны, такой результат говорит о высокой эффективности собственно алгоритма LR1, а с другой стороны, что не удалось достичь поставленной цели: ускорить алгоритм LR1 в подпространствах Крылова. И дело не только в том, что алгоритм LR1bCGs работает медленнее (в секундах) – в конце концов, его реализация предполагает двойное решение системы с помощью предобуславливателя на базе LR1 – но и, что важно, количество итераций требуется, вообще говоря, больше.

Выход из создавшегося положения заключается в особенностях алгоритма Bi-CGStab P, в котором используется только правое предобуславливание. При этом в нем не применяется в явном виде разложение K на множители K_1 и K_2 , но в неявном виде это разложение, естественно, предполагается для того, чтобы иметь возможность решить системы $Ky = p_k$ (строка 9 алгоритма Bi-CGStab P) и $Kz = s$ (строка 13 алгоритма Bi-CGStab P) с линейной трудоемкостью относительно числа неизвестных. Действительно, путем последовательного решения двух соответствующих пар систем $K_1\chi = p_k$, $K_2y = \chi$ и $K_1\xi = s$, $K_2z = \xi$ это легко сделать в силу треугольных структур матриц K_1 и K_2 .

По определению предобуславливатель – это легко обратимая матрица K , в некотором смысле близкая к исходной, то есть $K \approx A$. При этом считается, что качество предобуславливателя тем выше, чем ближе он к исходной матрице системы. Вполне естественно эту близость рассматривать как близость решений x и x_A двух систем $Kx = b$ и $Ax_A = b$. С этой точки зрения вектор y , который является точным решением системы $Ky = p_k$ (строка 9 алгоритма Bi-CGStab P), можно трактовать как приближенное решение системы $Ay_A = p_k$. Соответственно вектор z (строка 13 алгоритма Bi-CGStab P) – как приближенное решение системы $Az_A = s$. Получается, что вместо того, чтобы точно решать вспомогательную систему с матрицей предобуславливателя K , можно просто найти приближенное решение той же вспомогательной системы, но уже с основной матрицей A , применяя какой-либо дополнительный метод, который для удобства можно условно обозначить как метод «YZ». В этом случае строки 9 и 13 алгоритма Bi-CGStab P должны быть заменены на следующие:

...
9'. Определение с помощью метода «YZ» вектора приближенного решения y системы $Ay_A = p_k$;

...
13'. Определение с помощью метода «YZ» вектора приближенного решения z системы $Az_A = s$;

...
Иными словами идеологически (но не математически!) нет особой разницы между тем, чтобы точно решить приближенную систему или тем, чтобы приближенно решить точную. На этом основана технология «внедрения» метода «YZ» в алгоритм Bi-CGStab P, которая и составляет суть прямого сочетания «YZ» с Bi-CGStab P, поскольку ни предобуславливатель, ни, тем более, его факторизация не используются. Понятно, что метод «YZ» должен быть итерационным (в противном случае им сразу можно было бы решить исходную систему $AF = b$). При этом вопрос об оптимальном количестве итераций при использовании метода «YZ» остается открытым. Также остается открытым вопрос о начальных приближениях векторов y и z . В настоящей работе в целях максимального сближения результатов предлагаемого подхода и технологии предобуславливания с неполной факторизацией исходной матрицы в качестве начальных приближений y и z использовались нулевые вектора.

Поскольку эффективность прямого сочетания двух методов зависит от точности приближения векторов y и z при минимальных затратах машинного времени для их нахождения, то отсюда следует, что в качестве «YZ» лучше всего использовать такой метод, который за первую итерацию способен резко понизить начальную погрешность решения (соответственно и невязку). В этом смысле применение какого-либо из алгоритмов полинейного рекуррентного метода представляется наиболее оправданным. Дело в том, что данный метод характеризуется «фирменной» особенностью: он резко (на 2 – 4 порядка) понижает первоначальную невязку за первый итерационный проход [1, 2, 8].

В итоге на основании изложенных рассуждений можно определить новый метод, представляющий собой прямое сочетание алгоритмов LR1 и Bi-CGStab P, в дальнейшем для удобства называемый LR1sK. По отношению к предыдущему варианту LR1bCGs метод LR1sK обладает двумя очевидными преимуществами: во-

первых, он опосредованно учитывает все элементы предполагаемой нижней треугольной матрицы K_1 , сохраняя при этом линейную трудоемкость относительно числа неизвестных; во-вторых, в силу точности реализации алгоритма LR1 он наследует его свойство быть прямым методом относительно линейного по координатам решения. При этом понятно, что если вместо LR1 использовать алгоритм LR2 [1,2], то вновь образованный метод будет прямым для квадратичного по координатам решения.

Вычислительный эксперимент

Для оценки рассмотренных методов LR1bCGs и LR1sK был проведен их сравнительный анализ путем решения тестовых задач 1 и 2. Их эффективность сравнивалась с эффективностью метода LR1 и трех алгоритмов метода бисопряженных градиентов со стабилизацией: Bi-CGStab (без предобуславливания), Bi-CGStab P LU (с предобуславливанием на базе LU разложения матрицы A), Bi-CGStab P B (с предобуславливанием на базе явного метода Булеева). Итерационные параметры (в тех алгоритмах, в которых они присутствуют) подбирались таким образом, чтобы реализовать максимальную скорость сходимости решения. Тем самым производилась своеобразная оценка сверху эффективности каждого из рассматриваемых алгоритмов.

Кривые сходимости в виде зависимостей $\|R^k\|_2 / \|R^0\|_2$ от номера итерации представлены на рис. 3, *а* (первая задача) и рис. 3, *б* (вторая задача). Видно, что отличительной особенностью графиков, так или иначе связанных с методом бисопряженных градиентов (кривые 1, 3 – *б*), является их резко немонотонное поведение, что качественно хорошо согласуется с результатами [5, 6]. Еще одной особенностью является отсутствие сходимости решения при использовании алгоритма Bi-CGStab для задачи 2 (несамосопряженный оператор; рис. 3, *б*, кривая 6).

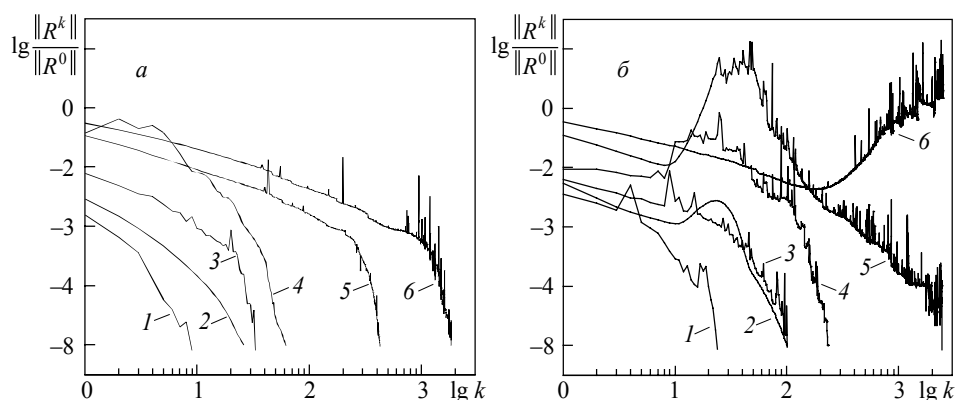


Рис. 3. Кривые сходимости: *а* – первая задача; *б* – вторая задача; кр. 1 – LR1sK; кр. 2 – LR1; кр. 3 – LR1bCGs; кр. 4 – Bi-CGStab P B; кр. 5 – Bi-CGStab P LU; кр. 6 – Bi-CGStab

Итоговые результаты решения обеих задач всеми рассматриваемыми методами представлены в таблице. Кроме количества итераций и времени, необходимых для завершения итерационного процесса, здесь также представлены значения

$\|Z^k\|_2 / \|Z^0\|_2$ и Q^k на момент его завершения. Поскольку алгоритмом Bi-CGStab за 2500 итераций не было достигнуто требуемой точности, то вместо количества итераций указано минимальное значение $\|R^k\|_2 / \|R^0\|_2$, полученное в процессе итераций. Обращает на себя внимание, что отношение норм погрешностей всего на 1 – 3 порядка отличается от заявленной точности решения, что говорит об эффективных разрешающих возможностях всех методов, несмотря на достаточно высокие значения чисел обусловленности задач.

Задача		Метод					
		LR1sK	LR1	LR1bCGs	Bi-CGStab P B	Bi-CGStab P LU	Bi-CGStab
1	Итерации	9	26	33	62	430	1847
	Время, с	5,5	8,9	14,9	4,8	30,3	70,5
	$\ Z^k\ _2 / \ Z^0\ _2$	$4,8 \cdot 10^{-7}$	$2,6 \cdot 10^{-7}$	$1,0 \cdot 10^{-5}$	$9,8 \cdot 10^{-6}$	$3,2 \cdot 10^{-5}$	$1,2 \cdot 10^{-5}$
	Q^k	1,620	0,580	0,350	0,190	0,024	0,006
2	Итерации	24	101	101	235	2431	$1,86 \cdot 10^{-3}$
	Время, с	13,9	33,0	42,7	16,7	170,0	–
	$\ Z^k\ _2 / \ Z^0\ _2$	$1,4 \cdot 10^{-6}$	$1,6 \cdot 10^{-5}$	$5,6 \cdot 10^{-6}$	$2,2 \cdot 10^{-6}$	$4,3 \cdot 10^{-6}$	–
	Q^k	0,560	0,110	0,120	0,055	0,005	–

Из поведения графиков и данных таблицы хорошо видно, что по общему количеству итераций, необходимых для достижения заданной точности решения и итоговым значениям средней скорости сходимости, алгоритмы, в которых присутствует LR1 (кривые 1 – 3), имеют явное преимущество над остальными алгоритмами, представленными на рис. 3 (кривые 4 – 6). Однако по количеству времени, необходимому для завершения итерационного процесса не все так очевидно. Для задачи 1 (самосопряженный оператор) минимальное время реализуется в случае использования алгоритма Bi-CGStab P B, хотя по количеству необходимых итераций он в семь раз превосходит алгоритм LR1sK. В случае более сложной задачи 2 (несамосопряженный оператор) ситуация меняется в пользу алгоритма LR1sK – теперь он явный лидер как по числу итераций, так и по затрачиваемому на сходимость решения времени. Отсюда можно сделать предположение, что для более сложных задач алгоритм LR1sK должен уверенно демонстрировать свои преимущества по отношению к алгоритму Bi-CGStab P B (по отношению к остальным рассмотренным алгоритмам его превосходство сомнений не вызывает).

Для проверки этого предположения были решены алгоритмами LR1sK и Bi-CGStab P B модифицированные задачи 1 и 2 – вместо условий Дирихле на всех границах области были выставлены условия Неймана. Кривые сходимости всех четырех расчетов представлены на рис. 4. Нетрудно видеть, что для достижения заданной точности при использовании алгоритма LR1sK потребовалось 17 итераций (задача 1) и 40 итераций (задача 2), а при использовании алгоритма Bi-CGStab P B – 212 и 998 итераций соответственно. При этом затраты по времени для первой и второй задачи соответственно составили 10,1 с и 22,7 с для алгоритма LR1sK и 15,2 с и 69,8 с для алгоритма Bi-CGStab P B. Иными словами предположение о преимуществе LR1sK перед Bi-CGStab P B при решении более сложных задач в данном случае подтвердилось.

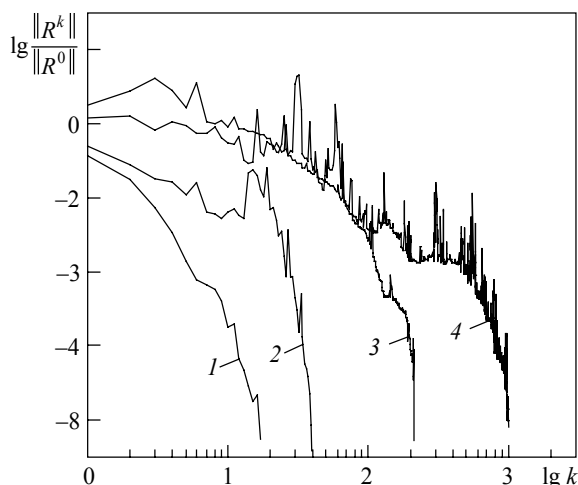


Рис. 4. Задача 1: кр. 1 – LR1sK, кр. 3 – Bi-CGStab P B;
задача 2: кр. 2 – LR1sK, кр. 4 – Bi-CGStab P B

Выводы

На основании полученных результатов можно сделать следующие выводы.

1. Выявлена структура сомножителей неполной факторизации исходной матрицы системы на базе алгоритма LR1 полинейного рекуррентного метода решения эллиптических СЛАУ.

2. Показано, что использование предобуславливателя на базе LR1 в методе сопряженных градиентов со стабилизацией не приводит к ускорению вычислений по отношению к использованию самого алгоритма LR1.

3. Установлено, что в силу особенностей алгоритма Bi-CGStab P с ним напрямую можно сочетать любой итерационный метод решения СЛАУ.

4. Показано, что прямое сочетание алгоритмов LR1 и Bi-CGStab P позволяет значительно повысить скорость сходимости решения по отношению к использованию исходного алгоритма LR1.

ЛИТЕРАТУРА

1. Фомина Л.Н. Использование полинейного рекуррентного метода с переменным параметром компенсации для решения разностных эллиптических уравнений // Вычислительные технологии. ИВТ СО РАН. 2009. Т. 14. № 4. С. 108–120.
2. Фомин А.А., Фомина Л.Н. Сравнение эффективности высокоскоростных методов решения разностных эллиптических СЛАУ // Вестник Томского государственного университета. Математика и механика. 2009. № 2(6). С. 71–77.
3. Yousef Saad. Iterative Methods for Sparse Linear Systems. – N.Y.: PWS Publ., 1996. 460 p.
4. Ильин В.П. Методы конечных разностей и конечных объемов для эллиптических уравнений. Новосибирск: Изд-во Ин-та математики, 2000. 345 с.
5. Van der Vorst H.A. BI-CGSTAB: a fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems // SIAM J. Sci. Stat. Comput. 1992. V. 13. No. 2. P. 631–644.
6. Старченко А.В. Сравнительный анализ некоторых итерационных методов для численного решения пространственной краевой задачи для уравнений эллиптического типа // Вестник ТГУ. Бюллетень оперативной научной информации. Томск: ТГУ, 2003. № 10. С. 70–80.

7. Патанкар С. Численные методы решения задач теплообмена и динамики жидкости. М.: Энергоатомиздат, 1984. 152 с.
8. Фомин А.А., Фомина Л.Н. Об одном варианте полинейного рекуррентного метода решения разностных эллиптических уравнений // Вестник Томского государственного университета. Математика и механика. 2010. № 2(10). С. 20–27.
9. Вишивков В.А., Засыпкина О.А. Итерационный метод решения СЛАУ первого порядка сходимости с регулируемой матрицей перехода // Сибирский журнал индустриальной математики. 2008. Т. 11. № 2. С. 40–49.

Статья поступила 11.09.2010 г.

Fomin A.A., Fomina L.N. ACCELERATION OF THE LINE-BY-LINE RECURRETNT METHOD IN KRYLOV SUBSPACES. Two techniques of acceleration of line-by-line recurrent method in Krylov subspaces are considered by the example of the LR1 algorithm. The van der Vorst Bi-CGStab P algorithm is used as an accelerating method. It is shown that the traditional approach (generation of a preconditioner on the base of LR1 algorithm) doesn't yield the required result. At the same time, the direct combination of LR1 and Bi-CGStab P algorithms allows to raise the convergence speed considerably.

Keywords: difference elliptic equations, iterative method, Krylov subspaces, line-by-line recurrent method.

FOMIN Alexander Arkadyevich

E-mail: fomin_aa@mail.ru

FOMINA Lubov Nikolaevna (Kemerovo State University)

E-mail: lubafomina@mail.ru