

$$\text{если } \rho_1 \neq 1, \text{ и } I = \frac{\sum_{0 \leq l+m \leq 3} \Theta_{l,m} j^l k^m}{\sum_{0 \leq l+m \leq 2} \omega_{l,m} j^l k^m}, \text{ если } \rho_1 = 1, \text{ где } \tau, \kappa, \xi_{l,m}, \delta_{l,m},$$

$\Theta_{l,m}, \omega_{l,m}$ — константы, зависящие от параметров системы, вид которых не приводится из соображений экономии места.

Имея соотношения (12), (13), несложно реализовать на ЭВМ процедуру минимизации функции двух переменных и получить в итоге оптимальные параметры (j^*, k^*) гистерезисной стратегии управления режимом работы данной СМО.

З а м е ч а н и е 1. Имея формулы (3)—(7), можно получить явную зависимость от j, k функционала качества функционирования системы и при более сложном, чем (1), его виде.

З а м е ч а н и е 2. Результат теоремы 1 остается справедливым, если допустить, что с изменением режима работы прибора меняется и интенсивность поступающего потока. Надо только в аргументе преобразований Лапласа — Стильтеса $\beta_v(\lambda(1-z))$ поставить у λ индекс $v, v = 1, 2$.

З а м е ч а н и е 3. Из замечания 2 вытекает, что результаты исследования данной модели можно применять и для решения различных задач динамического управления потоками в сетях связи.

Список литературы

1. Гуд В. // ТИИЭР. 1984. Т. 72. № 11. С. 180.
2. Самойленко С. И. // Методические материалы и документация по пакетам прикладных программ. 1983. Ч. 1. Вып. 24. С. 43.
3. Crabill T., Gross P., Magazine M. // Oper. Res. 1977. V. 25. № 2. P. 219.
4. Горцев А. М., Назаров А. А., Терпугов А. Ф. Управление и адаптация в системах массового обслуживания. Томск, 1978.

Поступила в редакцию 22.03.88.

УДК 519:612

В. А. БАСИК

ОБ ОДНОМ НЕЯВНОМ АЛГОРИТМЕ РЕШЕНИЯ ГРАНИЧНЫХ ЗАДАЧ ДЛЯ РАЗНОСТНЫХ ТРЕХТОЧЕЧНЫХ УРАВНЕНИЙ

При решении задачи Дирихле для дифференциальных уравнений второго порядка эллиптического типа по методу сеток возникают системы линейных алгебраических уравнений вида [1—3]:

$$A_i V_{i-1} - B_i V_i + C_i V_{i+1} = F_i, \quad i = 1, 2, \dots, M-1, \quad (1)$$

$$V_0 = F_0, \quad V_M = F_M, \quad (2)$$

где A_i, C_i — диагональные квадратные матрицы порядка N с положительными элементами на главной диагонали; B_i — квадратные трехдиагональные матрицы того же порядка; F_i — известные векторы. Систему вида (1)—(2) называют [1] также граничной задачей для векторных трехточечных уравнений.

Использование для решения задачи (1)—(2) метода матричной прогонки требует [1] выполнения $\approx CN^3M$ арифметических операций и запоминания $\approx N^2M$ чисел — результатов промежуточных вычислений. Методы полной редукции, быстрого преобразования Фурье требуют значительно меньшего объема вычислений [1], но применимы только к частным видам уравнения (1). Ниже строится итерационный алгоритм решения задачи (1)—(2), в основу которого положена комбинация идей модифицированного метода матричной прогонки [1] и неявных итерационных методов, предложенных В. П. Ильиным в [4] и [5]. При построении алгоритма используется близость матриц A_i, B_i, C_i к матрицам $A_{i+1},$

B_{i+1}, C_{i+1} . Доказывается, что при выполнении определенных условий метод будет сходиться со скоростью, не зависящей от числа узлов сетки.

Предположим существование связи между векторами V_{i-1} и V_i следующего вида:

$$V_{i-1} = P_i(B_{i-1})C_{i-1}V_i + Z_i, \quad (3)$$

где $P_i(B_{i-1})$ — некоторая квадратная матрица. Из (2) и (3) следует

$$P_1(B_0)C_0 = 0, \quad Z_1 = F_0. \quad (4)$$

Получим рекуррентные формулы для вычисления $P_i(B_{i-1})$ и Z_i . С этой целью перепишем уравнение (1) в виде $A_i V_{i-1} - (B_i + A_i P_i(B_{i-1})C_{i-1} - P_i(B_i))V_i + C_i V_{i+1} = F_i - (A_i P_i(B_{i-1})C_{i-1} - P_i(B_i))V_i$. Исключив с помощью (3) V_{i-1} , получим $(B_i - P_i(B_i))V_i = C_i V_{i+1} - F_i + A_i Z_i + (B_i - P_i(B_i) + Q_i)V_i$, где $Q_i = A_i P_i(B_{i-1})C_{i-1} - B_i$. Если

$$\det(B_i - P_i(B_i)) \neq 0, \quad (5)$$

то, сравнив последнее равенство с (3), приходим к рекуррентным формулам:

$$P_{i+1}(B_i) = (B_i - P_i(B_i))^{-1}, \quad (6)$$

$$Z_{i+1} = -P_{i+1}(B_i)[F_i - A_i Z_i - Q_i V_i] + V_i. \quad (7)$$

Если $P_i(B_i)$ определить по правилу $P_i(B_i) = A_i P_i(B_{i-1})C_{i-1}$, то из формул (3) — (4), (6) — (7) получим метод матричной прогонки. В других случаях по аналогии с [4, 5] можем построить итерационный процесс:

$$Z_1^{(n+1)} = F_0,$$

$$Z_{i+1}^{(n+1)} = -P_{i+1}(B_i)[F_i - A_i Z_i^{(n+1)} - Q_i V_i^{(n)}] + V_i^{(n)}, \quad i = 1, 2, \dots, M-1, \quad (8)$$

$$V_M^{(n+1)} = F_M, \quad V_{i-1}^{(n+1)} = P_i(B_{i-1})C_{i-1}V_i^{(n+1)} + Z_i^{(n+1)}, \quad i = M, M-1, \dots, 2. \quad (9)$$

Если итерационный процесс (8) — (9) сходится, то он сходится к решению задачи (1) — (2). Это следует из построения (8) — (9).

Важную роль в описываемом процессе играет выбор матриц $P_k(B_i)$. В [4, 5] в качестве $P_k(B_i)$ берутся ленточные матрицы. Мы исследуем случай

$$P_{i+1}(B_i) = U_i^{-1}(B_i/2)U_{i-1}(B_i/2), \quad (10)$$

где $U_i(x)$ — полиномы Чебышева второго рода i -й степени:

$$U_i(x) = \frac{(x + \sqrt{x^2 - 1})^{i+1} - (x - \sqrt{x^2 - 1})^{i+1}}{2\sqrt{x^2 - 1}}.$$

Представление (10) позволяет находить произведение матрицы $P_k(B)$ на вектор V по следующей схеме. Вначале решаются системы

$$\left(B - 2E \cos \frac{m\pi}{k}\right)W_k^{(m)} = w_{mk}Y, \quad m = 1, 2, \dots, k-1, \quad (11)$$

где $w_{mk} = \frac{2}{k} \sin^2 \frac{m\pi}{k}$, и определяются величины $w_k^{(m)}$. Затем находится искомое произведение

$$P_k(B)Y = \sum_{m=1}^{k-1} W_k^{(m)}. \quad (12)$$

Указанное обстоятельство позволяет провести вычисление $V_i^{(n+1)}$ следующим образом.

1. Задаем $Z_1^{(n+1)} = F_0$.

2. Полагаем $Y = C_{i-1}V_i^{(n)}$ и по (11) — (12) вычисляем вектор $\Omega = P_i(B_{i-1})Y$. Затем задаем $Y = A_i(Z_i^{(n+1)} + \Omega) - B_i V_i^{(n)} - F_i$ и с помощью (11) — (12) находим $Z_{i+1}^{(n+1)} = P_{i+1}(B_i)Y$. Вычисления на данном этапе проводим последовательно при $i = 1, 2, \dots, M-1$. При $i = 1$ можно положить $C_0 = 0, B_0 = B_1$.

3. Задаем $V_M^{(n+1)} = F_M$.

4. Последовательно при $i = M, M-1, \dots, 1$ находим $Y = C_{i-1} V_i^{(n+1)}$
 $V_{i-2}^{(n+1)} = P_i(B_{i-1})Y + Z_i^{(n+1)}$.

В описываемой схеме требуется память для запоминания вспомогательных векторов $Z_i^{(n+1)}$, а основной объем вычислений приходится на решение систем линейных алгебраических уравнений с матрицами, имеющими такую же структуру, как и B_i . На каждой итерации нужно решать $\approx 1,5M^2$ таких систем. Уменьшить число решаемых систем можно, если на 4-м этапе запоминать векторы $P_i(B_{i-1})C_{i-1}V_i^{(n+1)}$.

Для выполнения условий (5) достаточно, чтобы

$$\|B_i^{-1}\| \leq 1/2, \quad i = 1, 2, \dots, M-1. \quad (13)$$

В этом случае

$$\|P_{k+1}(B_i)\| \leq \frac{k}{k+1}, \quad k = 0, 1, \dots, M-1. \quad (14)$$

Исследуем сходимость предлагаемого здесь алгоритма. При этом будем ориентироваться на граничные разностные задачи, возникающие при решении по методу сеток задачи Дирихле для дифференциальных уравнений эллиптического типа, определенных в прямоугольной области и близких к уравнению Пуассона. Будем считать, что справедлива оценка (13), $\|A_i\| \approx 1, \|C_i\| \approx 1$ и

$$C_i B_i = B_i C_i, \quad i = 1, 2, \dots, M-1. \quad (15)$$

Пользуясь представлением (6), легко получить равенство $P_k(B_{i-1}) - P_k(B_i) = P_k(B_{i-1})(B_i - B_{i-1})P_k(B_i) + P_k(B_{i-1})(P_{k-1}(B_{i-1}) - P_{k-1}(B_i))P_k(B_i)$. Отсюда с учетом (14) получаем оценку

$$\|P_k(B_{i-1}) - P_k(B_i)\| \leq \frac{(k-1)(2k-1)}{6k} \|B_i - B_{i-1}\|. \quad (16)$$

Если выполняется (15), то имеем $\|P_{i+1}^{-1}(B_i) + Q_i\| = \|A_i P_i(B_{i-1}) C_{i-1} - P_i(B_i)\| = \|(A_i C_{i-1} - E)P_i(B_{i-1}) + P_i(B_{i-1}) - P_i(B_i)\| \leq \|A_i C_{i-1} - E\| \|P_i(B_{i-1})\| + \|P_i(B_{i-1}) - P_i(B_i)\|$. Принимая во внимание (14), (16), получаем

$$\|P_{i+1}^{-1}(B_i) + Q_i\| \leq \frac{i-1}{i} \omega + \frac{(i-1)(2i-1)}{6i} s, \quad (17)$$

где $\omega = \max_i \|A_i C_{i-1} - E\|$, $s = \max_i \|B_i - B_{i-1}\|$.

Обозначим $\varepsilon_i^{(n)} = V_i - V_i^{(n)}$, $\tau_i^{(n)} = Z_i - Z_i^{(n)}$. Вычитая (8) — (9) из (7) и (3) соответственно, приходим к задаче относительно $\varepsilon_i^{(n+1)}$:

$$\tau_1^{(n+1)} = 0, \quad \tau_{i+1}^{(n+1)} = P_{i+1}(B_i) [A_i \tau_i^{(n+1)} + (P_{i+1}^{-1}(B_i) + Q_i) \varepsilon_i^{(n)}], \quad (18)$$

$$i = 1, 2, \dots, M-1.$$

$$\varepsilon_M^{(n+1)} = 0, \quad \varepsilon_{i-1}^{(n+1)} = P_i(B_{i-1}) C_{i-1} \varepsilon_i^{(n+1)} + \tau_i^{(n+1)}, \quad i = M, M-1, \dots, 2. \quad (19)$$

Отметим, что в случае постоянных относительно i коэффициентов A_i , B_i , C_i и выполнении условий (13) — (15), векторы $\tau_i^{(n+1)}$ будут равняться нулю, если только $A_i C_{i-1} = E$. Значит, в этом случае $\varepsilon_i^{(n+1)} = 0$.

Из (14), (17) — (18) имеем: $\|\tau_1^{(n+1)}\| = 0$, $\|\tau_{i+1}^{(n+1)}\| \leq \frac{\alpha i}{i+1} \|\tau_i^{(n+1)}\| + \left(\frac{i-1}{i+1} \omega + \frac{(i-1)(2i-1)}{6(i+1)} s \right) \varepsilon^{(n)}$, где $\alpha = \max\{1, \|A_i\|, \|C_i\|\}$, $\varepsilon^{(n)} = \max_i \|\varepsilon_i^{(n)}\|$. Последнее неравенство позволяет получить оценку $\|\tau_{i+1}^{(n+1)}\| \leq \left(\frac{i}{2} \omega + \frac{i^2}{9} s \right) \alpha^{i-1} \varepsilon^{(n)}$.

Из (14), (19) имеем $\|\varepsilon_M^{(n+1)}\| = 0$, $\|\varepsilon_{i-1}^{(n+1)}\| \leq \frac{i-1}{i} \alpha \|\varepsilon_i^{(n+1)}\| + \|\tau_i^{(n+1)}\|$,

поэтому для $\varepsilon_i^{(n+1)}$ верна оценка $\|\varepsilon_i^{(n+1)}\| \leq \left[\frac{i(M^2 - i^2)}{18} s + i \frac{M - i}{2} \omega \right] \times \alpha^{2(M-1)-i} \varepsilon^{(n)}$. Так как $i(M^2 - i^2) \leq \frac{2M^3}{3\sqrt{3}}$, $i(M - i) \leq \frac{M^2}{4}$, то окончательно: $\|\varepsilon_i^{(n+1)}\| \leq \left[\frac{M^3}{27\sqrt{3}} s + \frac{M^2}{8} \omega \right] \alpha^{2(M-1)} \varepsilon^{(n)}$.

Полученная оценка показывает, что в случае слабо изменяющихся коэффициентов дифференциального уравнения, из которого получено уравнение (1), метод будет сходиться со скоростью, не зависящей от числа узлов сетки. При этом объем вычислений будет лишь в несколько раз превосходить объем вычислений решения задачи (1) — (2) с постоянными коэффициентами по модифицированному методу матричной прогонки [1].

Численные эксперименты по решению разностных граничных задач с помощью предлагаемого алгоритма проводились на ЕС ЭВМ с использованием алгоритмического языка ПЛ/1. Основные вычисления выполнялись над числами с плавающей точкой стандартной длины (≈ 6 десятичных цифр). Дифференциальные задачи аппроксимировались разностными на квадратной сетке с узлами (x_i, y_j) , $i, j = 0, 1, \dots, N$, $x_i = ih$, $y_j = jh$, $h = 1/N$. Начальные приближения при каждом брались равными нулю во всех внутренних узлах. На каждой итерации полученное приближение сравнивалось с предыдущим приближением и точным решением задачи.

Таблица 1

Погрешность решения задачи (20) на k -й итерации

N	10	20	30	40	50
$\varepsilon^{(1)}$	$5,7 \cdot 10^{-3}$	$7,1 \cdot 10^{-3}$	$7,7 \cdot 10^{-3}$	$8,0 \cdot 10^{-3}$	$8,1 \cdot 10^{-3}$
$\varepsilon^{(2)}$	$2,3 \cdot 10^{-5}$	$4,6 \cdot 10^{-5}$	$6,9 \cdot 10^{-5}$	$9,3 \cdot 10^{-5}$	$8,8 \cdot 10^{-5}$

Таблица 2

Число итераций для решения задачи (21)
с точностью 10^{-4}

c_2/c_1	$N=10$	$N=20$	$N=30$	$N=40$	$N=50$
2	4	4	4	4	4
8	5	5	6	6	6
32	5	6	6	6	7
128	5	6	6	7	7
512	5	6	6	7	7

В табл. 1 приводятся значения $\varepsilon^{(k)} = \max_{i,j} |u_{ij}^{(k)} - u(x_i, y_j)|$, где $u(x, y) = 1$ — решение граничной задачи

$$\frac{\partial^2 u(x, y)}{\partial x^2} + \frac{\partial^2 u(x, y)}{\partial y^2} - xu(x, y) = -x, \quad 0 \leq x, y \leq 1, \quad (20)$$

а u_{ij} — решение разностной задачи в узле (x_i, y_j) , полученной из дифференциальной путем замены производных разностными отношениями с использованием пятиточечного шаблона.

При аппроксимации задачи

$$\frac{\partial}{\partial x} \left(K_1(x, y) \frac{\partial u}{\partial y} \right) + \frac{\partial}{\partial y} \left(K_2(x, y) \frac{\partial u}{\partial x} \right) = f(x, y), \quad 0 \leq x, y \leq 1, \\ u|_{\Gamma} = g(x, y) \quad (21)$$

использовалась разностная схема из [1, с. 254]. Следуя [1], функции $f(x, y)$, $g(x, y)$ взяты так, чтобы $u(x, y) = x(1-x)y(1-y)$ при $K_1(x, y) = 1 + C[(x-0,5)^2 + (y-0,5)^2]$, $K_2(x, y) = 1 + C[0,5 - (x-0,5)^2 - (y-0,5)^2]$. Величина C подбиралась так, чтобы отношение c_2/c_1 , где $c_1 = 1$, $c_2 = 1 + 0,5C$, соответствовало значениям этого отношения, рассматриваемым в [1].

Непосредственное применение предлагаемого алгоритма к решению разностной задачи приводило к медленной сходимости. Скорость сходимости зависела от числа узлов сетки. Полезной оказалась предварительная нормировка системы (1), (2), состоящая в умножении (1) слева на матрицу $(A_i C_i)^{-1/2}$. Отметим, что такая нормировка приводит к приближенному равенству $A_i C_{i-1} \approx E$. Число итераций, выполненных до достижения условия $\max_{i, j} |u_{ij}^{(k)} - u_{ij}^{(k-1)}| \leq 10^{-4}$, приводится в табл. 2.

Список литературы

1. Самарский А. А., Николаев Е. С. Методы решения сеточных уравнений. М., 1978.
2. Самарский А. А. Теория разностных схем. М., 1983.
3. Марчук Г. И. Методы вычислительной математики. М., 1980.
4. Ильин В. П. Численные методы решения задач электрофизики. М., 1985.
5. Ильин В. П. // Сиб. матем. журн. 1985. Т. 26. № 1. С. 83.

Поступила в редакцию 03.07.87.

УДК 519.685

Л. В. ПЕВЗНЕР

О НЕКОТОРЫХ АЛГОРИТМАХ ПРЕОБРАЗОВАНИЯ ГРАММАТИК

В последнее время помимо контекстно-свободных (КС) грамматик для описания языков программирования используются грамматики более широкого класса, охватывающего класс КС грамматик. Для эффективного применения таких грамматик достаточно перспективным представляется иметь несколько анализаторов, разработанных специально для отдельных формализаций, и систему алгоритмов, позволяющих сводить грамматики одного вида к грамматикам другого с наименьшими потерями их специфических свойств.

В настоящей работе рассмотрим грамматики, которые можно разбить на следующие классы: к классу K_1 отнесем графовые, т. е. WEB-грамматики и ГПГ — графово-программные [1], к классу K_2 : РГ — R-грамматики [2]; ЦПГ — цепочечные программные [3]; ИГ — индексные [4]; ГВВ — грамматики ван Вейнгаардена [4].

Установим взаимосвязь между ИГ и РГ. Введем отношение W , означающее анализ цепочки, находящейся в вершине соответствующего магазина, но не выталкивание из него.

Пусть дана ИГ $G^1 = (V_T^1, V_N^1, F, P, \sigma^1)$. Построим для нее РГ $G_2 = (V_T^2, V_N^2, \Gamma, R^2, P^2, r_0, r_\emptyset)$. Здесь V_T^1, V_N^1 — терминальные алфавиты; P — множество правил; σ^1 — аксиома; F — множество индексов; R^2 — множество комплексов; Γ — алфавит G^2 .

Для любой грамматики можно построить эквивалентную ей индексную грамматику в приведенной форме [3]; таким образом, для простоты будем устанавливать связь только между ИГ в приведенной форме и РГ G^2 .

Считаем, что все правила из P^1 имеют вид:

- | | |
|----------------------------|-----------------------|
| а) $A \rightarrow Bf$ | в) $A \rightarrow a$ |
| б) $f = [A \rightarrow B]$ | г) $A \rightarrow BC$ |