МОДЕЛИРОВАНИЕ В ПЛЕНОЧНОЙ ЭЛЕКТРОМЕХАНИКЕ
(Вычислительные системы)

1989 год Выпуск 131

УДК 517.927.2

О МЕТОДЕ ДИФФЕРЕНЦИАЛЬНЫХ ВСТРЕЧНЫХ ПРОГОНОК

С.И.Фадеев

Рассматривается метод встречных прогонок численного решения линейной двухточечной краевой задачи для системы из n обыкновенных дифференциальных уравнений на конечном отрезке. Как и в известном методе ортогональной прогонки С.К. Годунова [1,2], здесь используется процедура, препятствующая "сплющиванию" базисных решений однородной системы, что позволяет найти численное решение краевой задачи, если она достаточно хорошо обусловлена. Эффективность метода иллюстрируется на тестовых примерах.

Автор выражает благодарность С.К.Годунову за полезные замечания.

1. Пусть $A(t)$ - матричная функция скалярного аргумента t размерности $n \times n$, которую для определенности будем считать непрерывной на отрезке $[a, b]$; $f(t)$ - непрерывная на том же отрезке вектор-функция размерности n. Для уравнения

$$\frac{dy}{dt} + A(t)y = f(t), \quad a \leq t \leq b \ , \tag{1}$$

относительно вектор-функции $y(t)$ размерности n рассмотрим краевую задачу, подчинив $y(t)$ условиям:

$$Ly\big|_{t=a} = l, \quad Ry\big|_{t=b} = r, \tag{2}$$

где L - матрица размерности $k \times n$, R - матрица размерности $m \times k$, $m + k = n$;, l и r - векторы размерности k и m соответственно, $\operatorname{rang} L = k$, $\operatorname{rang} R = m$. В дальнейшем будем предполагать, что краевая задача (1), (2) однозначно разрешима. Как известно [3], для существования единственного решения (1), (2), непрерывно дифференцируемого на отрезке $[a, b]$, необходимо и достаточно выполнения условия:

$$\det \begin{bmatrix} LY(a) \\ RY(b) \end{bmatrix} \neq 0, \tag{3}$$

где $Y(t)$ - фундаментальная матрица решений уравнения:

$$\frac{dy}{dt} + A(t)y = 0. \tag{4}$$

В связи с описанием метода встречных прогонок рассмотрим на отрезке $[a, b]$ две задачи Коши относительно матричной функции $u_a(t)$ размерности $k \times n$ и матричной функции $u_b(t)$ размерности $m \times n$:

$$\begin{aligned} \frac{du_a}{dt} &= u_a A(t), \quad u_a\big|_{t=a} = L, \\ \frac{du_b}{dt} &= u_b A(t), \quad u_b\big|_{t=b} = R. \end{aligned} \tag{5}$$

Определив из (5) $u_a(t)$ и $u_b(t)$, составим квадратную $u(t)$ и транспонированную $u^T(t)$ матрицы:

$$u(t) = \begin{bmatrix} u_a(t) \\ u_b(t) \end{bmatrix}, \quad u^T(t) = \left[u_a^T(t), u_b^T(t)\right]. \tag{6}$$

ТЕОРЕМА 1. *Пусть краевая задача (1), (2) однозначно разрешима. Тогда* $u^T(t)$ *является фундаментальной матрицей решений уравнения*

$$\frac{dy}{dt} = A^T(t)y. \tag{7}$$

ДОКАЗАТЕЛЬСТВО. Из (5) следует, что $u_a^T(t)$ и $u_b^T(t)$ являются решениями задач Коши:

$$\frac{du_a^T}{dt} = A^T(t)u_a^T, \quad u_a^T\Big|_{t=a} = L^T,$$
$$\frac{du_b^T}{dt} = A^T(t)u_b^T, \quad u_b^T\Big|_{t=b} = R^T. \tag{8}$$

Таким образом, каждый из вектор-столбцов (6) матричной функции $u^T(t)$ обращает уравнение (7) в тождество. Поэтому для доказательства теоремы достаточно убедиться в том, что $\det u^T(t) \neq 0$, или, что то же самое, $\det u(t) \neq 0$ при всех $t \in [a, b]$.

Пусть $U^T(t)$ - фундаментальная матрица решений (7). Представим решения (8) в виде:

$$u_a^T(t) = U_a^T(t)[U^T(a)]^{-1}L^T,$$

$$u_b^T(t) = U^T(t)[U^T(b)]^{-1}R^T.$$

Отсюда находим, что

$$u_a(t) = LU^{-1}(a)U(t),$$

$$u_b(t) = RU^{-1}(b)U(t),$$

а для составной матрицы $u(t)$ получим выражение:

$$u(t) = \begin{bmatrix} LU^{-1}(a) \\ RU^{-1}(b) \end{bmatrix} U(t).$$

Далее заметим, что $U(t)$ обращает в тождество уравнение

$$\frac{dU}{dt} = UA(t) \ .$$

Следовательно (достаточно продифференцировать тождество $U^{-1}(t)U(t) \equiv I$), $U^{-1}(t)$ является фундаментальной матрицей решений уравнения (4), т.е. $U(t) = Y^{-1}(t)$. С учетом последнего обстоятельства запишем $u(t)$ в виде:

$$u(t) = \begin{bmatrix} LY(a) \\ RY(b) \end{bmatrix} Y^{-1}(t) \ .$$

Поскольку по предположению теоремы условие (3) выполняется, то $\det u(t) \neq 0$. Теорема доказана.

Будем искать решение уравнения (1) в виде:

$$y(t) = u^{-1}(t)v(t) = \tilde{Y}(t)v(t) \ ,$$

где

$$\tilde{Y}(t) = Y(t) \begin{bmatrix} LY(a) \\ RY(b) \end{bmatrix}^{-1} -$$

фундаментальная матрица решений (4). Согласно методу вариаций произвольных постоянных, вектор-функция $v(t)$ должна удовлетворять уравнению

$$\tilde{Y}(t)\frac{dv}{dt} = f(t),$$

или

$$\frac{dv}{dt} = u(t)f(t) . \qquad (9)$$

Рассмотрим теперь две задачи Коши на отрезке $[a, b]$ относительно вектор-функции $v_a(t)$ размерности k и вектор-функции $v_b(t)$ размерности m :

$$\begin{aligned} &\frac{dv_a}{dt} = u_a(t)f(t), \quad v_a\Big|_{t=a} = 1, \\ &\frac{dv_b}{dt} = u_b(t)f(t), \quad v_b\Big|_{t=b} = r. \end{aligned} \qquad (10)$$

Очевидно, составной вектор

$$v(t) = \begin{bmatrix} v_a(t) \\ v_b(t) \end{bmatrix} \qquad (11)$$

обращает в тождество уравнение (9).

ТЕОРЕМА 2. *Пусть* $u(t)$ *- составная матрица (6),* $v(t)$ *- составной вектор (11). Тогда для каждого* $t \in [a,b]$ *решение краевой задачи (1),(2) определяется как решение системы линейных алгебраических уравнений:*

$$u(t)y(t) = v(t). \qquad (12)$$

Действительно, по построению матрицы $u(t)$ и вектора $v(t)$ из решений задач Коши (5) и (10) решение (12) обращает уравнение (1) в тождество. Кроме того, из представления (12) в виде:

$$\begin{aligned} &u_a(t)y(t) = v_a(t), \\ &u_b(t)y(t) = v_b(t), \end{aligned}$$

а также в силу начальных условий задач Коши (5) и (10) сразу следует выполнение краевых условий (2). Теорема доказана.

Вычисление $u_a(t)$ и $v_a(t)$ из решения задач Коши:

$$\frac{du_a^T}{dt} = A^T(t)u_a^T \,, \qquad u_a^T\big|_{t=a} = L^T \,,$$

$$\frac{dv_a^T}{dt} = f^T(t)u_a^T \,, \qquad v_a^T\big|_{t=a} = l^T \,,$$

будем называть прямым правым ходом прогонки. Удобно объединить $u_a(t)$ и $v_a(t)$, представив прогонку в виде одной задачи Коши для однородной системы. Имеем

<u>прямой правый ход прогонки:</u>

$$\frac{d}{dt}\begin{bmatrix} u_a^T \\ v_a^T \end{bmatrix} = \begin{bmatrix} A^T(t) & 0 \\ f^T(t) & 0 \end{bmatrix}\begin{bmatrix} u_a^T \\ v_a^T \end{bmatrix}, \quad \begin{bmatrix} u_a^T \\ v_a^T \end{bmatrix}_{t=a} = \begin{bmatrix} L^T \\ l^T \end{bmatrix}. \qquad (13.1)$$

Матрица системы (13.1) содержит $(n+1)$-й нулевой столбец. Аналогично для вычисления $u_b(t)$ и $v_b(t)$ из решения задачи Коши определяется

<u>прямой левый ход прогонки:</u>

$$\frac{d}{dt}\begin{bmatrix} u_b^T \\ v_b^T \end{bmatrix} = \begin{bmatrix} A^T(t) & 0 \\ f^T(t) & 0 \end{bmatrix}\begin{bmatrix} u_b^T \\ v_b^T \end{bmatrix}, \quad \begin{bmatrix} u_b^T \\ v_b^T \end{bmatrix}_{t=b} = \begin{bmatrix} R^T \\ r^T \end{bmatrix}. \qquad (13.2)$$

По завершении встречных прогонок (13.1), (13.2) остается найти решение краевой задачи $y(t)$, обратившись к системе линейных алгебраических уравнений (12) с матрицей $u(t)$ и правой частью $v(t)$, причем, как следует из теоремы 1, $\det u(t) \neq 0$, если краевая задача (1), (2) однозначно разрешима.

Отметим связь $u(t)$ с матричной функцией Грина $G(t,s)$, $t,s \in [a,b]$, краевой задачи:

$$\frac{dy}{dt} + A(t)y = 0, \quad Ly|_{t=a} = 0, \quad Ry|_{t=b} = 0 .$$

Воспользуемся определением $G(t,s)$ как обобщенным решением краевой задачи с правой частью в виде δ-функции:

$$\frac{dG}{dt} + A(t)G = \delta(t-s)I, \quad LG|_{t=a} = 0, \; RG|_{t=b} = 0. \tag{14}$$

Формальным применением формул встречных прогонок к уравнению (14) получается следующее представление матричной функции Грина:

$$u(t)G(t,s) = \begin{bmatrix} u_a(s) \\ 0 \end{bmatrix}, \quad t > s,$$
$$u(t)G(t,s) = \begin{bmatrix} 0 \\ -u_b(s) \end{bmatrix}, \quad t < s, \tag{15}$$

При этом легко проверяется выполнение условий, определяющих $G(t,s)$:

1) $\dfrac{dG}{dt} + A(t)G = 0, \quad t \neq s;$

2) $G(s+0,s) - G(s-0,s) = I;$

3) $LG|_{t=a} = 0, \quad RS|_{t=b} = 0.$

Согласно (15),

$$\sup_{t,s}\|G(t,s)\| \leq K = \sup_t\|u(t)\|\sup_t\|u^{-1}(t)\| .$$

Напомним, что из однозначной разрешимости краевой задачи (1), (2) следует непрерывная зависимость решения от $A(t)$, $f(t)$, a, b, L, R, l и r в достаточно малой окрестности изменения параметров. Размеры окрестности зависят от константы K, оценивающей сверху норму функции Грина. В этом случае говорят о достаточно хорошей обусловленности краевой задачи.

2. Остановимся на проблеме "сплющивания" решений однородных систем (13.1), (13.2), образующих базисы подпространств, отвечающих краевым условиям при $t = a$ и $t = b$ соответственно. Как известно, игнорирование этого явления при численном решении методом прогонки даже хорошо обусловленной краевой задачи (1), (2) может в ряде случаев привести к неверным результатам или быть причиной авоста. Впервые на проблему "сплющивания" базисных решений однородной системы уравнений было указано в [1]. Там же был предложен эффективный метод решения краевой задачи (1), (2), получивший название ортогональной прогонки С.К.Годунова. Основная идея метода состоит в следующем. При реализации прямого правого хода прогонки отрезок $[a, b]$ разбивается определенным образом на N частей:

$$a=t_1<t_2<\ldots< t_N<t_{N+1}= b, \quad h_i = t_{i+1}- t_i, \tag{16}$$

и на каждом из отрезков $[t_i, t_{i+1}]$ в качестве начальных условий при $t = t_i$ серии задач Коши, описывающих базисные решения однородной системы (4), выбирается система ортонормированных векторов. Следовательно, за счет выбора шагов h_i, $h_i\|A(t_i)\| \approx 1$, базисные решения при всех $t \neq t_i$ остаются "почти ортонормированными".

Преследуя те же цели в рассматриваемом методе встречных прогонок, будем сопровождать численные решения задачи Коши (13.1), (13.2) ортогональной нормировкой в узлах сетки (16) (термин "ортогональная нормировка" предложен С.К.Годуновым). Для определенности остановимся на правом ходе прогонки.

Пусть при $t = t_i$ найдены элементы матрицы $[u_a(t_i), v_a(t_i)]$. На первом шаге процедуры ортогональной нормировки: а) выбирается по всей матрице $u_a(t_i)$ главный элемент, и столбец, его содержащий, меняется местом с первым вектор-столбцом; b) матрица $[u_a(t_i), v_a(t_i)]$ после возможной перестановки столбцов подвергается ортогональному преобразованию отражения, в результате которого все элементы первого столбца, кроме первого элемента, зануляются, а абсолютная величина первого элемента, таким образом, становится равной евклидовой норме первого вектор-столбца; с) элементы первой строки преобразованной матрицы $[u_a(t_i), v_a(t_i)]$ делятся на первый элемент.

Обозначим через $[u_a^{[1]}(t_i), v_a^{[1]}(t_i)]$ результат преобразований первого шага процедуры. На втором шаге пп. "a", "b", "c" применяются к матрице, у которой мысленно вычеркнуты первая строка и первый столбец, причем элементы первой строки участвуют только в перестановке столбцов при выборе главного элемента. На третьем шаге пп. "a", "b", "c" применяются к матрице $[u_a^{[2]}(t_i), v_a^{[2]}(t_i)]$ - итог второго шага - с мысленно вычеркнутыми первыми двумя строками и столбцами и т.д. Очевидно, матрица $[u_a^{[k]}(t_i), v_a^{[k]}(t_i)]$ будет иметь трапецевидную форму с единичными ведущими элементами на главной диагонали. Завершается ортогональная нормировка перестановкой столбцов $u_a^{[k]}(t_i)$, восстанавливающей их последовательность в матрице $u_a(t_i)$. В итоге получим матрицу $[\tilde{u}_a(t_i), \tilde{v}_a(t_i)]$, $\tilde{u}_a(t_i) = u_a^{[k]}(t_i)$, $\tilde{v}_a(t_i) = v_a^{[k]}(t_i)$, $i = 1, 2, \ldots, N$.

Аналогичным преобразованиям подвергается матрица $[u_b(t_i), v_b(t_i)]$ при реализации левого прямого хода прогонки. В результате ортогональной нормировки будем иметь матрицу $[\tilde{u}_b(t_i), \tilde{v}_b(t_i)]$, $i = N+1, N, \ldots, 3, 2$.

Дополним описание метода встречных прогонок, ранее представленного формулами (13.1), (13.2) и (12). При этом задачи Коши решаются последовательно на отрезках $[t_i, t_{i+1}]$ и всякий раз начальные условия предварительно подвергаются ортогональной нормировке.

Прямой правый ход прогонки:

$$\frac{d}{dt}\begin{bmatrix} u_a^T \\ v_a^T \end{bmatrix} = \begin{bmatrix} A^T(t) & 0 \\ f^T(t) & 0 \end{bmatrix}\begin{bmatrix} u_a^T \\ v_a^T \end{bmatrix}, \quad \begin{bmatrix} u_a^T \\ v_a^T \end{bmatrix}_{t=t_i} = \begin{bmatrix} \tilde{u}_a(t_i) \\ \tilde{v}_a(t_i) \end{bmatrix},$$

$$\begin{bmatrix} u_a^T(t_{i+1}) \\ v_a^T(t_{i+1}) \end{bmatrix} = S_{i+1}\begin{bmatrix} u_a^T(t_{i+1}) \\ v_a^T(t_{i+1}) \end{bmatrix}, \; i = 1, 2, \ldots, N, \tag{17.1}$$

где S_{i+1} - операция ортогональной нормировки при $t = t_{i+1}$.

Прямой левый ход прогонки:

$$\frac{d}{dt}\begin{bmatrix} u_b^T \\ v_b^T \end{bmatrix} = \begin{bmatrix} A^T(t) & 0 \\ f^T(t) & 0 \end{bmatrix}\begin{bmatrix} u_b^T \\ v_b^T \end{bmatrix}, \quad \begin{bmatrix} u_b^T \\ v_b^T \end{bmatrix}_{t=t_{i+1}} = \begin{bmatrix} u_b(t_{i+1}) \\ v_b(t_{i+1}) \end{bmatrix},$$

$$\begin{bmatrix} u_b^T(t_{i+1}) \\ v_b^T(t_{i+1}) \end{bmatrix} = S_i\begin{bmatrix} u_b^T(t_i) \\ v_b^T(t_i) \end{bmatrix}, \quad i = N+1, N, \ldots, 2. \tag{17.2}$$

Очевидно, решение краевой задачи (1), (2) в узлах сетки определится из решения системы линейных алгебраических уравнений:

$$
\left.\begin{aligned}
& j = 1, 2, \dots, N+1, \\
& \tilde{u}(t_j) y(t_j) = \tilde{v}(t_j) ,
\end{aligned}\right. \qquad (18)
$$

где

$$
\tilde{u}(t_j) = \begin{bmatrix} \tilde{u}_a(t_j) \\ \tilde{u}_b(t_j) \end{bmatrix}, \quad \tilde{v}(t_j) = \begin{bmatrix} \tilde{v}_a(t_j) \\ \tilde{v}_b(t_j) \end{bmatrix}.
$$

3. Эффективность применения ортогональной нормировки можно наблюдать на следующем примере. Пусть A - постоянная матрица краевой задачи (1),(2), причем ее спектр строго делится мнимой осью, так что $\alpha_1, \alpha_2, \dots, \alpha_k$ - собственные числа A, лежащие в правой полуплоскости, а $\beta_1, \beta_2, \dots, \beta_m$ - собственные числа, лежащие в левой полуплоскости, и $k+m=n$. Напомним, что k - число краевых условий на левом конце отрезка по t, m - число краевых условий на правом конце. Для моделирования ситуации, когда сплющивание базисных решений имеет принципиальное значение при численном решении краевой задачи методом встречных прогонок, мы будем строить матрицу A с заданными собственными числами. Однако несмотря на достаточно большой разброс значений собственных чисел A, краевая задача останется хорошо обусловленной.

Описание примера начнем с рассмотрения двух задач Коши для уравнений высокого порядка относительно функций $x_1(t)$ и $x_2(t)$ на отрезке $[0,1]$ по t:

$$
\left.\begin{aligned}
& P_k\left(\frac{d}{dt}\right)x_1 = F_1(t), \\
& x_1(0) = \frac{dx_1}{dt}(0) = \dots = \frac{d^{k-1}x_1}{dt^{k-1}}(0) = 0,
\end{aligned}\right\} \qquad (19.1)
$$

$$\left.\begin{aligned} &Q_m\left(\frac{d}{dt}\right) x_2 = F_2(t), \\ &x_2(1) = \frac{dx_2}{dt}(1) = \ldots = \frac{d^{m-1}x_2}{dt^{m-1}}(1) = 0 \end{aligned}\right\} \tag{19.2}$$

Здесь $F_1(t)$ и $F_2(t)$ - непрерывные на отрезке $[0,1]$ функции; $P_k(\frac{d}{dt})$, $Q_m(\frac{d}{dt})$ - дифференциальные операторы с постоянными вещественными коэффициентами степени k и m :

$$P_k(\lambda) = \lambda^k + \sum_{i=1}^{k} p_i\lambda^{k-i} = \prod_{i=1}^{k}(\lambda-\alpha_i),$$

$$Q_m(\lambda) = \lambda^m + \sum_{j=1}^{m} q_j\lambda^{m-j} = \prod_{j=1}^{m}(\lambda-\beta_j).$$

Пусть $g_1(t)$ - вектор-функция размерности k, A_1 - $k\times k$-матрица; $g_2(t)$ - вектор-функция размерности m, A_2 - $m\times m$-матрица, определенные в виде:

$$g_1(t) = [0,\ldots,0,F_1(t)]^T,$$

$$g_2(t) = [0,\ldots,0,F_2(t)]^T,$$

$$A_1=\begin{bmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ & & \vdots & & \\ & & & 0 & 1 \\ -p_k & -p_{k-1} & \cdots & & -p_1 \end{bmatrix}, \quad A_2=\begin{bmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ & & \vdots & & \\ & & & 0 & 1 \\ -q_m & -q_{m-1} & \cdots & & -q_1 \end{bmatrix},$$

причем, по построению, $\alpha_1, \alpha_2, \ldots, \alpha_k$ - собственные числа матрицы A_1, $\mathrm{Re}\,\alpha_i \le -\delta < 0$, $i = 1,2,\ldots,k$, а $\beta_1, \beta_2, \ldots, \beta_m$ - собственные числа матрицы A_2, $\mathrm{Re}\,\beta_j \ge \delta > 0$, $j = 1,2,\ldots,m$. Как известно, рассматриваемым задачам Коши можно придать эквивалетные формулировки:

$$\frac{dw_1}{dt} = A_1 w_1 + g_1(t), \quad w_1|_{t=0} = 0, \tag{20.1}$$

$$\frac{dw_2}{dt} = A_2 w_2 + g_2(t), \quad w_2|_{t=1} = 0, \tag{20.2}$$

где

$$w_1 = \left[x_1, \frac{dx_1}{dt}, \ldots, \frac{d^{k-1}x_1}{dt^{k-1}}\right]^T,$$

$$w_2 = \left[x_2, \frac{dx_2}{dt}, \ldots, \frac{d^{m-1}x_2}{dt^{m-1}}\right]^T.$$

Записав решение (20.1) и (20.2) в виде

$$w_1(t) = \int_0^t e^{(t-s)A_1} g_1(s)\,ds, \quad w_2(t) = -\int_t^1 e^{(t-s)A_2} g_2(s)\,ds,$$

образуем из $w_1(t)$ и $w_2(t)$ составной вектор $w(t)$ размерности n:

$$w(t) = \begin{bmatrix} w_1(t) \\ w_2(t) \end{bmatrix} = \int_0^t \begin{bmatrix} e^{(t-s)A_1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} g_1(s) \\ g_2(s) \end{bmatrix} ds +$$

$$+\int_t^1 \begin{bmatrix} 0 & 0 \\ 0 & -e^{(t-s)A_2} \end{bmatrix} \begin{bmatrix} g_1(s) \\ g_2(s) \end{bmatrix} ds. \quad (21)$$

Отсюда следует, что матричная функция $\Gamma(t,s)$,

$$\Gamma(t,s) = \begin{cases} \Gamma_1(t,s), & t > s, \\ \Gamma_2(t,s), & t < s, \end{cases}$$

где

$$\Gamma_1(t,s)=\begin{bmatrix} e^{(t-s)A_1} & 0 \\ 0 & 0 \end{bmatrix}, \quad \Gamma_2(t,s)=\begin{bmatrix} 0 & 0 \\ 0 & -e^{(t-s)A_2} \end{bmatrix},$$

является матричной функцией Грина однородной краевой задачи:

$$\frac{dw}{dt} + \begin{bmatrix} -A_1 & 0 \\ 0 & -A_2 \end{bmatrix} w = 0, \qquad t \in [0,1],$$

$$[I_k, 0]w|_{t=0} = 0, \quad [0, I_m]w|_{t=1} = 0.$$

Здесь I_k и I_m - единичные матрицы размерности $k \times k$ и $m \times m$ соответственно. Отметим, что согласно лемме Гельфанда-Шилова имеют место оценки норм матричных экспонент:

$$\|e^{(t-s)A_1}\| \le c(k) \left(\frac{\|A_1\|}{\delta}\right)^{k-1} e^{-\frac{\delta}{2}(t-s)}, \quad t > s,$$

$$\|e^{(t-s)A_2}\| \le c(m) \left(\frac{\|A_2\|}{\delta}\right)^{m-1} e^{-\frac{\delta}{2}(s-t)}, \quad s > t,$$

где $c(j)$ - постоянная, зависящая только от j. Обратившись далее к (21), (22), приходим к выводу, что норма матричной функции Грина $\Gamma(t,s)$ будет ограничена, каков бы ни был спектр матрицы A, подчиненный указанному условию дихотомии мнимой осью.

Вообще говоря, уже краевая задача

$$\frac{dw}{dt} + \begin{bmatrix} -A_1 & 0 \\ 0 & -A_2 \end{bmatrix} w = \begin{bmatrix} g_1(t) \\ g_2(t) \end{bmatrix}, \quad t \in [0,1], \tag{22}$$

$$[I_k, 0]w|_{t=0} = l, \quad [0, I_m]w|_{t=1} = r,$$

позволяет иллюстрировать явление сплющивания базисных решений. Пусть, например, при $k = n-2$ собственными числами матриц A_1 и A_2 будут: $\alpha_i = -1$, $i = 1,2,\dots,k-1$, $\alpha_k = -\alpha_0$, $\alpha_0 \gg 1$; $\beta_1 \geq 1$, $\beta_2 \geq \beta_1$.

Тогда

$$\left.\begin{aligned} &p_1 = n-3+\alpha_0, \quad p_{n-2} = \alpha_0, \quad n > 3, \\ &p_i = \frac{(n-3)(n-4)\dots(n-1-i)}{(i-1)!}\left(\frac{n-2-i}{i} + \alpha_0\right), \\ &\qquad\qquad i = 2,3,\dots,n-3, \\ &q_1 = -(\beta_1 + \beta_2), \quad q_2 = \beta_1\beta_2, \end{aligned}\right\} \tag{23}$$

что полностью определяет матрицы A_1 размерности $(n-2)\times(n-2)$ и A_2 размерности 2x2 системы уравнений (21). Представляет интерес сопоставление точного и численного решений краевой задачи, когда α_0, β_1 и β_2 много больше 1.

Для построения примера точного решения краевой задачи (22), сохраняющего n в качестве параметра, поступим простейшим образом. Возьмем достаточно гладкие на отрезке $[0,1]$ функции

$\varphi_1(t)$ и $\varphi_2(t)$ и определим правые части уравнений (18.1), (18.2) в виде:

$$F_1(t) = P_{n-2}\left(\frac{d}{dt}\right)\varphi_1(t), \quad F_2(t) = Q_2\left(\frac{d}{dt}\right)\varphi_2(t).$$

При начальных условиях:

$$x_1(0) = \varphi_1(0),$$

$$\frac{dx_1}{dt}(0) = \frac{d\varphi_1}{dt}(0), \ldots, \frac{d^{n-3}x_1}{dt^{n-3}}(0) = \frac{d^{n-3}\varphi_1}{dt^{n-3}}(0),$$

$$x_2(1) = \varphi_2(1), \quad \frac{dx_2}{dt}(1) = \frac{d\varphi_2}{dt}(1),$$

функции $x_1 = \varphi_1(t)$, $x_2 = \varphi_2(t)$, очевидно, будут решениями соответствующих задач Коши. Следовательно, вектор-функция

$$w(t) = \left[\varphi_1(t), \frac{d\varphi_1}{dt}(t), \ldots, \frac{d^{n-3}\varphi_1}{dt^{n-3}}(t), \varphi_2(t), \frac{d\varphi_2}{dt}(t)\right]^T$$

дает решение краевой задачи (22), где

$$l = \left[\varphi_1(0), \frac{d\varphi_1}{dt}(0), \ldots, \frac{d^{n-3}\varphi_1}{dt^{n-3}}(0)\right]^T, \quad r = \left[\varphi_2(1), \frac{d\varphi_2}{dt}(1)\right]^T.$$

Нам осталось придать краевой задаче (22) общий вид, применив к матрице системы преобразование подобия. Здесь удобно использовать ортогональную симметрическую матрицу Q в качестве матрицы перехода, $Q^TQ = I$, $Q^T = Q$. Рассмотрим краевую задачу:

$$\frac{dy}{dt} + Ay = f(t), \qquad 0 \le t \le 1,$$
$$Ly|_{t=0} = l, \quad Ry|_{t=1} = r, \tag{24}$$

где

$$A = Q\begin{bmatrix} -A_1 & 0 \\ 0 & -A_2 \end{bmatrix} Q \,, \qquad f(t) = Q\begin{bmatrix} g_1(t) \\ g_2(t) \end{bmatrix} \,,$$

$$L = [I_k, 0]Q \,, \qquad R = [0, I_m]Q \,.$$

Из предыдущего следует, что

$$G(t,s) = Q\Gamma(t,s)Q$$

является матричной функцией Грина однородной краевой задачи:

$$\frac{dy}{dt} + Ay = 0 \,, \quad 0 \le t \le 1 \,,$$

$$Ly|_{t=} = 0 \,, \quad Ry|_{t=1} = 0 \,,$$

причем норма $G(t,s)$ ограничена. Далее, если $w(t)$ - решение (22), то вектор-функция $y(t) = Qw(t)$, очевидно, будет решением краевой задачи (24).

Как известно, отражения Хаусхолдера задаются симметрической ортогональной матрицей. Пусть, например, матрица Q определяется из условия, что отраженный вектор имеет отличный от нуля только первый элемент:

$$Q\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} -\sigma\|z\| \\ 0 \\ \vdots \\ 0 \end{bmatrix} \,, \qquad \sigma = \begin{cases} 1 \,, & z_1 \ge 0 \,, \\ -1 \,, & z_1 < 0 \,, \end{cases}$$

т.е.

$$Q = \begin{bmatrix} -\gamma_1 & -\gamma_2 & -\gamma_3 & \cdots & -\gamma_n \\ -\gamma_2 & 1-\frac{\gamma_2^2}{1+\gamma_1} & -\frac{\gamma_2\gamma_3}{1+\gamma_1} & \cdots & -\frac{\gamma_2\gamma_n}{1+\gamma_1} \\ -\gamma_3 & -\frac{\gamma_2\gamma_3}{1+\gamma_1} & 1-\frac{\gamma_3^2}{1+\gamma_1} & \cdots & -\frac{\gamma_3\gamma_n}{1+\gamma_1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ -\gamma_n & -\frac{\gamma_2\gamma_n}{1+\gamma_1} & -\frac{\gamma_3\gamma_n}{1+\gamma_1} & \cdots & 1-\frac{\gamma_n^2}{1+\gamma_1} \end{bmatrix},$$

где

$$\gamma_i = \frac{z_i}{\sigma\|z_i\|}, \qquad i = 1,2,\ldots, n.$$

В частности, если $z_2 = z_3 = \ldots = z_{n-1} = 0$, $z_1 > 0$ и $z_n > 0$, то

$$Q = \begin{bmatrix} -\rho_1 & 0 & -\rho_2 \\ 0 & I_{n-2} & 0 \\ -\rho_2 & 0 & \rho_1 \end{bmatrix},$$

где

$$\rho_1 = \frac{z_1}{\sqrt{z_1^2 + z_n^2}}, \qquad \rho_2 = \frac{z_n}{\sqrt{z_1^2 + z_n^2}} = \sqrt{1 - \rho_1^2}.$$

В последнем случае матрицы A, L и R краевой задачи (24), где $k = n-2$, определяются заданием параметров n, α_0, β_1, β_2 и ρ_1: $n \geq 4$, $\alpha_0 \geq 1$, $\beta_1 \geq 1$, $\beta_2 \geq \beta_1$, $0 \leq \rho_1 \leq 1$, причем α_0, $-\beta_1$ и $-\beta_2$ являются собственными числами A (остальные равны 1). Вместе с A, L и R

эти параметры определяют функцию Грина $G(t,s)$ с ограниченной нормой. Имеем:

$$k = n-2,\ \rho_2 = \sqrt{1-\rho_1^2},$$

$$A = \begin{bmatrix} \rho_2^2 q_1 & \rho_1 & 0 & \dots & 0 & -\rho_2 q_2 & -\rho_1\rho_2 q_1 \\ 0 & 0 & -1 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -1 & 0 & 0 \\ -\rho_1 p_k & p_{k-1} & p_{k-2} & \dots & p_1 & 0 & -\rho_2 p_k \\ \rho_2 & 0 & 0 & \dots & 0 & 0 & -\rho_1 \\ -\rho_1\rho_2 q_1 & \rho_2 & 0 & \dots & 0 & \rho_1 q_2 & \rho_1^2 q_1 \end{bmatrix},$$

$$f(t)=[-\rho_2 F_2(t), 0, \dots, 0, F_1(t), 0, \rho_1 F_2(t)-\rho_2 F_1(t)],$$

$$L = \begin{bmatrix} -\rho_1 & 0 & 0 & \dots & 0 & 0 & -\rho_2 \\ 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 & 0 \end{bmatrix},$$

$$R = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 1 & 0 \\ -\rho_2 & 0 & 0 & \dots & 0 & 0 & \rho_1 \end{bmatrix},$$

где p_i, $i = 1,2,\dots,n-2$, q_1 и q_2 вычисляются по формулам (23). При желании можно включить в число параметров краевой задачи бо́льшее число собственных чисел матрицы A, подчинив их различным условиям дихотомии.

4. Программная реализация метода встречных прогонок подтвердила эффективность используемого здесь способа нормировки.

В случае тестового примера легко указать значения параметров α_0, β_1 и β_2 в зависимости от разрядности ЭВМ, когда ожидаемое совпадение точного и численного решений краевой задачи имеет место лишь при использовании нормировки. В противном случае возникает авостная ситуация. Предложенный ранее вариант метода встречных прогонок [4] этим свойством в общем случае не обладает.

5. В заключение отметим частный случай формулировки краевой задачи (1), (2).

Если при $t=a$ либо при $t=b$ задано только одно краевое условие, то при организации соответствующего прямого хода прогонки не возникает проблемы сплющивания базисных решений, а ортогональная нормировка совпадает с обычной нормировкой. Пусть, например, $k=1$ и

$$[u_a(t), v_a(t)] = [u_{11}^{(a)}(t), \ldots, u_{1n}^{(a)}(t), v_1^{(a)}(t)],$$

причем

$$[u_a(t), v_a(t)]|_{t=a} = [L_{11}, L_{12}, \ldots, L_{1n}, l_1].$$

Тогда при $t = t_j, j = 1, 2, \ldots, N$, операция S_{j+1} (17.1) состоит в определении максимального по модулю элемента $u_{1i_0}^{(a)}(t_{j+1})$,

$$\rho = |u_{1i_0}^{(a)}(t_{j+1})| = \max_i |u_{1i}^{(a)}(t_{j+1})|,$$

и вычислении $[\tilde{u}_a(t), \tilde{v}_a(t)]$ по формуле:

$$t = t_{j+1}, \quad [\tilde{u}_a(t), \tilde{v}_a(t)] = \frac{1}{\rho}[u_a(t), v_a(t)].$$

Пусть теперь краевая задача формулируется для двух уравнений первого порядка: $n = 2$, $k = m = 1$. Из предыдущего ясно, что решения задач Коши (17.1), (17.2) сопровождаются процедурой обычной нормировки, включенной в выбранный числен - ный метод решения задач Коши [5].

Литература

1. ГОДУНОВ С.К. О численном решении краевых задач для систем линейных обыкновенных дифференциальных уравнений //Успехи мат.наук. - 1961. -Т.16, №3. -С. 171-175.

2. КУЗНЕЦОВ С.В. Решение краевых задач для обыкновенных дифференциальных уравнений. - Новосибирск: Наука. Сиб.отд-ние (Тр. ин-та математики). - 1985. -Т.6. -С. 85-110.

3. БИБИКОВ Ю.Н. Общий курс обыкновенных дифференциальных уравнений. -Л.: Изд. ЛГУ, 1981. - 232 с.

4. ФАДЕЕВ С.И. Универсальная программа численного решения краевой задачи методом дифференциальной прогонки //Методы сплайн-функций. - Новосибирск. - 1979. -Вып. 81: Вычислитель - ные системы. -С. 87-100.

5. ФАДЕЕВ С.И. О численном решении линейной краевой задачи для обыкновенных дифференциальных уравнений методом диффе - ренциальной прогонки //Методы сплайн-функций. - Новосибирск. - 1978. -Вып. 75: Вычислительные системы. -С. 80-95.

Поступила в ред.-изд.отд.

23 окября 1989 года

ПРИЛОЖЕНИЕ

Приведем краткое описание на алгоритмическом языке BASIC программы BPR-L, предназначенной для решения краевой задачи (1), (2) методом встречных прогонок. Текст программы следует за перечислением обозначений:

N - число разбиений отрезка $[a,b]$;

N0 - размерность системы дифференциальных уравнений (1);

M0 - число краевых условий при $t = a$;

E9 - константа, определяющая условие обращения к процедуре нормировки: если на отрезке $[t_k, t_{k+m}], a \le t_k < t_{k+m} \le b$, окажется, что

$$\int_{t_k}^{t_{k+m}} \|A(t)\| dt > E9,$$

то при $t = t_{k+m}$ применяется операция ортогональной нормировки;

D[I] - узел t_i сетки (16) по t ;

M[J] - номер узла сетки, при котором требуется получить решение краевой задачи (1), (2) из системы (18);

I9 - число таких узлов.

Компоненты $A(t), f(t), L, R, l$ и r , входящие в формулировку краевой задачи, задаются при помощи массивов Q[I], G[I] и H[I]. Их соответствие устанавливается на примере. Если N0=4, M0 = 2, то формулировка краевой задачи (1),(2) принимает вид:

$$\frac{d}{dt}\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} + \begin{bmatrix} Q(1) & Q(2) & Q(3) & Q(4) \\ Q(6) & Q(7) & Q(8) & Q(9) \\ Q(11) & Q(12) & Q(13) & Q(14) \\ Q(16) & Q(17) & Q(18) & Q(19) \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} Q(5) \\ Q(10) \\ Q(15) \\ Q(20) \end{bmatrix},$$

$$\begin{bmatrix} G(1) & G(2) & G(3) & G(4) \\ G(6) & G(7(& G(8) & G(9) \end{bmatrix} \begin{bmatrix} y_1(a) \\ y_2(a) \\ y_3(a) \\ y_4(a) \end{bmatrix} = \begin{bmatrix} G(5) \\ G(10) \end{bmatrix},$$

$$\begin{bmatrix} H(1) & H(2) & H(3) & H(4) \\ H(6) & H(7H & H(8) & H(9) \end{bmatrix} \begin{bmatrix} y_1(b) \\ y_2(b) \\ y_3(b) \\ y_4(b) \end{bmatrix} = \begin{bmatrix} H(5) \\ H(10) \end{bmatrix}.$$

Значения N, NO, MO,E9,I9 и элементы массивов D[N+1], G[MO(NO+1)], H[(NO-MO)(NO+1)], M[I9] задаются в подпрограмме INPUT (строки 3000-3990). Элементы массива Q[NO(NO+1)] при каждом значении D (обозначение t) вычисляются в подпрограмме MATRIC (4000-4990).

В подпрограмме ORTOG (1840-2220) значения компонент $y_k(t_i)$ вектора $y(t_i)$, являющегося решением системы (18), присваиваются элементам массива Y[I,K], размерность Y[I9,NO].Значения t = D[M[J]] и y_k = Y[M[J],K], K = 1,2,...,NO, выдаются на печать в подпрограмме PRINT (5000-5990).

Остальные массивы программы носят вспомогательный характер. Укажем их размерности: A[3,NO(NO+1)], R[3],W[I9,NO(NO+1)], S[NO,NO+1], P[4,NO+1], U[NO(NO+1)], V[NO(NO+1)], K[NO], L[NO], C[4].

Для решения задач Коши (17.1), (17.2) используется "классический" метод Рунге-Кутта точности 4.

Тексты подпрограмм INPOT, MATRIC и PRINT, заполняемые пользователем, в данном случае содержат описанный выше тестовый пример, в котором $\alpha_0 = 20$, $\beta_1 = 10$, $\beta_2 = 30$, $z_1 = 3/5$, $z_2 = 4/5$ в программе соответственно задаются A0 =

= 20, A1 = 10, A2 = 30, P1 = 3/5, P2 = 4/5.) Кроме того, NO = =10,N = 40 с равномерным разбиением отрезка [0,1]. В качестве $\varphi_1(t)$ и $\varphi_2(t)$ взяты t^3 и $1-t$, так что точное решение краевой задачи имеет вид:

$$y(t)=[\rho_2-\rho_1 t^3, 3t^2, 6t, 0, \ldots, 0, 1-t, -\rho_1-\rho_2 t^3]^T.$$

В связи с примером используются вспомогательные массивы B[NO(NO+1)] и O[NO].

Далее приводятся текст программы BPR-L и печать результатов численного решения тестового примера (см.таблицу ниже). В таблице содержатся компоненты y(t) при $t = 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}$ и 1 (I9= = 5), найденные численно. Вторая половина таблицы - точные их значения.

Программа BPR-L

BPR-L

```
100   DIM D[101],A[3,110],Q[110],G[88],H[22]
110   DIM U[110],V[110],W[11,110],Y[11,10],M[11]
120   DIM S[10,11],P[4,11],K[10],L[10],C[4],R[3]
130   GOSUB 3000
140   N1=N0+1
150   N5=N+1
160   FOR M=1 TO N0
170   K[M]=N1*(M-1)
180   NEXT M
190   N2=M0
200   N3=0
210   L=I1=1
220   D=D[1]
230   GOSUB 4000
240   GOSUB 2230
250   FOR J=1 TO N2
260   FOR M=1 TO N1
270   U[K[J]+M]=G[K[J]+M]
280   NEXT M
290   NEXT J
300   IF M[1]>1 THEN 320
310   GOSUB 2580
320   GOSUB 1000
330   FOR I=1 TO N
340   H=D[I+1]-D[I]
350   GOSUB 2380
360   IF M[I1]>I+1 THEN 380
370   GOSUB 2580
380   NEXT I
390   N2=N0-M0
400   N3=N1*M0
410   FOR J=1 TO N2
420   FOR M=1 TO N1
430   U[K[J]+M]=H[K[J]+M]
440   NEXT M
450   NEXT J
460   IF M[I9]<N5 THEN 480
470   GOSUB 2580
480   GOSUB 1000
490   FOR I=N TO 1 STEP -1
500   H=D[I]-D[I+1]
510   GOSUB 2380
520   IF M[I1]<I THEN 540
530   GOSUB 2580
540   NEXT I
```

```
550  FOR I=1 TO 19
560  GOSUB 1040
570  NEXT I
580  GOSUB 5000
590  GOTO 6000
1000  REM               *** SUB-PROGRAM "NORM" ***
1010  R=0
1020  FOR J=1 TO N2
1030  FOR K=1 TO N1
1040  S[J,K]=U[K+K[J]]
1050  NEXT K
1060  NEXT J
1070  FOR K=1 TO N2
1080  C0=0
1090  FOR J=K TO N0
1100  C1=0
1110  FOR M=K TO N2
1120  C1=C1+S[M,J]↑2
1130  NEXT M
1140  IF C1<C0 THEN 1170
1150  C0=C1
1160  J0=L[K]=J
1170  NEXT J
1180  FOR M=1 TO N2
1190  Q[M]=S[M,J0]
1200  S[M,J0]=S[M,K]
1210  S[M,K]=Q[M]
1220  NEXT M
1230  Q0=((Q[K] >= 0)-(Q[K]<0))*SQR(C0)
1240  FOR M=K TO N2
1250  Q[M]=Q[M]/Q0
1260  S[M,K]=0
1270  NEXT M
1280  S[K,K]=1
1290  FOR J=K+1 TO N1
1300  C0=0
1310  FOR M=K TO N2
1320  C0=C0+Q[M]*S[M,J]
1330  NEXT M
1340  C1=(S[K,J]+C0)/(1+Q[K])
1350  S[K,J]=C0/Q0
1360  FOR M=K+1 TO N2
1370  S[M,J]=S[M,J]-C1*Q[M]
1380  NEXT M
1390  NEXT J
1400  NEXT K
```

```
1410  FOR K=N2 TO 1 STEP -1
1420  J0=L[K]
1430  FOR M=1 TO N2
1440  Q[M]=S[M,J0]
1450  S[M,J0]=S[M,K]
1460  S[M,K]=Q[M]
1470  NEXT M
1480  NEXT K
1490  FOR J=1 TO N2
1500  FOR K=1 TO N1
1510  U[K+K[J]]=S[J,K]
1520  NEXT K
1530  NEXT J
1540  RETURN
1550  REM               *** SUB-PROGRAM "RUNGE" ***
1560  C[3]=H
1570  C[1]=C[2]=.5*H
1580  C[4]=.166667*H
1590  FOR J=1 TO N2
1600  K0=K[J]
1610  FOR K=1 TO N1
1620  V[K+K0]=U[K+K0]
1630  NEXT K
1640  FOR L=1 TO 4
1650  I0=L-(L>2)
1660  FOR K=1 TO N1
1670  C0=0
1680  FOR M=1 TO N0
1690  C0=C0+A[I0,K[M]+K]*V[M+K0]
1700  NEXT M
1710  P[L,K]=C0
1720  NEXT K
1730  IF L=4 THEN 1770
1740  FOR K=1 TO N1
1750  V[K+K0]=U[K+K0]+C[L]*P[L,K]
1760  NEXT K
1770  NEXT L
1780  FOR K=1 TO N1
1790  M=K+K0
1800  U[M]=U[M]+C[4]*(P[1,K]+2*(P[2,K]+P[3,K])+P[4,K]
1810  NEXT K
1820  NEXT J
1830  RETURN
```

```
1840  REM                 *** SUB-PROGRAM "ORTOG" ***
1850  FOR J=1 TO N0
1860  FOR K=1 TO N1
1870  S[J,K]=W[I,K[J]+K]
1880  NEXT K
1890  NEXT J
1900  FOR K=1 TO N0
1910  C0=0
1920  FOR M=K TO N0
1930  C0=C0+S[M,K]↑2
1940  NEXT M
1950  Q0=((S[K,K] >= 0)-(S[K,K]<0))*SQR(C0)
1960  FOR M=K TO N0
1970  Q[M]=S[M,K]/Q0
1980  NEXT M
1990  S[K,K]=1
2000  FOR J=K+1 TO N1
2010  C0=0
2020  FOR M=K TO N0
2030  C0=C0+Q[M]*S[M,J]
2040  NEXT M
2050  C1=(S[K,J]+C0)/(1+Q[K])
2060  S[K,J]=C0/Q0
2070  FOR M=K+1 TO N0
2080  S[M,J]=S[M,J]-C1*Q[M]
2090  NEXT M
2100  NEXT J
2110  NEXT K
2120  FOR K=N0-1 TO 1 STEP -1
2130  C0=S[K,N1]
2140  FOR J=N0 TO K+1 STEP -1
2150  C0=C0-S[J,N1]*S[K,J]
2160  NEXT J
2170  S[K,N1]=C0
2180  NEXT K
2190  FOR J=1 TO N0
2200  Y[I,J]=S[J,N1]
2210  NEXT J
2220  RETURN
2230  REM                 *** SUB-PROGRAM "MASSIV" ***
2240  FOR M=1 TO N0
2250  FOR J=1 TO N1
2260  A[L,K[M]+J]=Q[K[M]+J]
2270  NEXT J
2280  NEXT M
```

```
2290  IF L=2 THEN 2370
2300  C0=0
2310  FOR M=1 TO N0
2320  FOR J=1 TO N0
2330  C0=C0+Q[K[M]+J]↑2
2340  NEXT J
2350  NEXT M
2360  R[L]=SQR(C0)
2370  RETURN
2380  REM                  *** SUB-PROGRAM "STEP" ***
2390  L=3
2400  D=D[I+(H>0)]
2410  GOSUB 4000
2420  GOSUB 2230
2430  L=2
2440  D=.5*(D[I]+D[I+1])
2450  GOSUB 4000
2460  GOSUB 2230
2470  GOSUB 1550
2480  R=R+.5*(R[1]+R[3])*ABS(H)
2490  R[1]=R[3]
2500  FOR M=1 TO N0
2510  FOR J=1 TO N1
2520  A[1,K[M]+J]=A[3,K[M]+J]
2530  NEXT J
2540  NEXT M
2550  IF R<E9 THEN 2570
2560  GOSUB 1000
2570  RETURN
2580  REM                  *** SUB-PROGRAM "LOCAL" ***
2590  FOR J=1 TO N2
2600  FOR M=1 TO N1
2610  W[I1,K[J]+M+N3]=U[K[J]+M]
2620  NEXT M
2630  NEXT J
2640  I1=I1+(I1<I9)*(N3=0)-(I1>1)*(N3>0)
2650  RETURN
```

```
3000  REM              *** SUB-PROGRAM "INPUT" ***
3010  READ N,N0,M0,E9,I9
3020  DATA 40,10,8,0,5
3030  READ M[1],M[2],M[3],M[4],M[5]
3040  DATA 1,11,21,31,41
3050  FOR I=1 TO N+1
3060  D[I]=(I-1)/N
3070  NEXT I
3080  DIM B[110],O[10]
3090  READ A0,A1,A2,P1,P2
3100  DATA 20,10,30,.6,.8
3110  C0=SQR(P1↑2+P2↑2)
3120  P1=P1/C0
3130  P2=P2/C0
3140  R1=-(A1+A2)
3150  R2=A1*A2
3160  N1=N0+1
3170  FOR M=1 TO N0
3180  FOR K=1 TO N1
3190  B[K+N1*(M-1)]=0
3200  NEXT K
3210  NEXT M
3220  Q=1
3230  A=O[1]=N0-3+A0
3240  FOR M=2 TO N0-3
3250  B[M+1+N1*(M-1)]=-1
3260  Q=Q*(N0-1-M)/M
3270  A=A+A0-1
3280  O[M]=Q*A
3290  NEXT M
3300  O[N0-2]=A0
3310  B[1]=P2↑2*R1
3320  B[2]=P1
3330  B[N0-1]=-P2*R2
3340  B[N0]=B[N1*(N0-1)+1]=-P1*P2*R1
3350  B[N1*(N0-3)+1]=-P1*O[N0-2]
3360  FOR M=2 TO N0-2
3370  B[N1*(N0-3)+M]=O[N0-1-M]
3380  NEXT M
3390  B[N1*(N0-2)-1]=-P2*O[N0-2]
3400  B[N1*(N0-2)+1]=B[N1*(N0-1)+2]=P2
3410  B[N1*(N0-1)-1]=-P1
3420  B[N1*N0-2]=P1*R2
3430  B[N1*N0-1]=P1↑2*R1
```

```
3440   FOR M=1  TO M0
3450   FOR K=1  TO N1
3460   G[K+N1*(M-1)]=0
3470   NEXT K
3480   G[M+N1*(M-1)]=1
3490   NEXT M
3500   G[1]=-P1
3510   G[N0]=-P2
3520   FOR M=1  TO 2
3530   FOR K=1  TO N1
3540   H[K+N1*(M-1)]=0
3550   NEXT K
3560   NEXT M
3570   H[N0-1]=1
3580   H[N0+2]=-P2
3590   H[2*N1-1]=P1
3600   IF M0<4 THEN 3620
3610   G[4*N1]=6
3620   H[2*N1]=-1
3990   RETURN
4000   REM                 *** SUB-PROGRAM "MATRIC" ***
4010   IF N0=4 THEN 4050
4020   IF N0=5 THEN 4070
4030   F1=6*O[N0-5]+D*(6*O[N0-4]+D*(3*O[N0-3]+D*O[N0-2]))
4040   GOTO 4080
4050   F1=D*(6+D*(3*O[1]+D*O[2]))
4060   GOTO 4080
4070   F1=6+D*(6*O[1]+D*(3*O[2]+D*O[3]))
4080   F2=R2*(1-D)-R1
4090   FOR M=1  TO N0
4100   Q[N1*M]=0
4110   FOR J=1  TO N0
4120   Q[J+K[M]]=B[J+K[M]]
4130   NEXT J
4140   NEXT M
4150   Q[N1]=-P2*F2
4160   Q[N1*(N0-2)]=F1
4170   Q[N1*N0]=P1*F2
4990   RETURN
5000   REM                 *** SUB-PROGRAM "PRINT" ***
5010   PRINT "                 N0=";N0;"N=";N;" E9="; E9
5020   PRINT "                 A0=";A0;"A1=";A1;"A2=";A2
5030   PRINT "                 P1=";P1;"P2=";P2
5040   PRINT
```

```
5050   PRINT "+----------+--------+---------+----------+
5060   PRINT "      T=0       T=.25       T=.5       T=.75
5070   PRINT "+----------+--------+---------+----------+
5080   FOR M=1 TO N0
5090   PRINT  USING "2X,5(SD.4D,4X)";Y[1,M],Y[2,M],Y[3,M]
5100   NEXT M
5110   FOR I=1 TO I9
5120   FOR M=1 TO N0
5130   Y[I,M]=0
5140   NEXT M
5150   D=D[M[I]]
5160   Y[I,1]=-P1*D↑3+P2
5170   Y[I,2]=3*D↑2
5180   Y[I,N0-1]=1-D
5190   Y[I,N0]=-P2*D↑3-P1
5200   IF N0=4 THEN 5240
5210   Y[I,3]=6*D
5220   IF N0=5 THEN 5240
5230   Y[I,4]=6
5240   NEXT I
5250   PRINT "+----------+--------+---------+----------+
5260   FOR M=1 TO N0
5270   PRINT  USING "2X,5(SD.4D,4X)";Y[1,M],Y[2,M],Y[3,M]
5280   NEXT M
5290   PRINT "+----------+--------+---------+----------+
5990   RETURN
6000   END
```

Таблица

Результаты численного решения тестового примера

```
RUN
BPR-L

            N0= 10    N= 40    E9= 0
            A0= 20    A1= 10    A2= 30
            P1= .6          P2= .8
```

T=0	T=.25	T=.5	T=.75	T=1
+0.8001	+0.7906	+0.7250	+0.5469	+0.2000
-0.0000	+0.1875	+0.7500	+1.6875	+3.0000
-0.0000	+1.5000	+3.0000	+4.5000	+6.0000
+6.0000	+6.0000	+6.0000	+6.0000	+6.0000
+0.0000	+0.0000	+0.0001	-0.0000	+0.0000
+0.0000	-0.0000	-0.0001	-0.0001	-0.0001
+0.0000	-0.0002	-0.0002	+0.0001	+0.0000
-0.0000	+0.0003	-0.0002	+0.0017	-0.0007
+1.0000	+0.7500	+0.5000	+0.2500	+0.0000
-0.6001	-0.6125	-0.7000	-0.9375	-1.4000
+0.8000	+0.7906	+0.7250	+0.5469	+0.2000
+0.0000	+0.1875	+0.7500	+1.6875	+3.0000
+0.0000	+1.5000	+3.0000	+4.5000	+6.0000
+6.0000	+6.0000	+6.0000	+6.0000	+6.0000
+0.0000	+0.0000	+0.0000	+0.0000	+0.0000
+0.0000	+0.0000	+0.0000	+0.0000	+0.0000
+0.0000	+0.0000	+0.0000	+0.0000	+0.0000
+0.0000	+0.0000	+0.0000	+0.0000	+0.0000
+1.0000	+0.7500	+0.5000	+0.2500	+0.0000
-0.6000	-0.6125	-0.7000	-0.9375	-1.4000

DONE