МЕТОДЫ СПЛАЙН-ФУНКЦИЙ
(Вычислительные системы)
1979 год Выпуск 81

УДК 517.927:519.62

УНИВЕРСАЛЬНАЯ ПРОГРАММА ЧИСЛЕННОГО РЕШЕНИЯ КРАЕВОЙ ЗАДАЧИ МЕТОДОМ ДИФФЕРЕНЦИАЛЬНОЙ ПРОГОНКИ

С.И. Фадеев

В работе дается изложение алгоритма численного решения многоточечной линейной краевой задачи на конечном интервале $[a, b]$ для системы из M_0, $M_0 \geq 1$, обыкновенных дифференциальных уравнений первого порядка [1]. Алгоритм представляет вариант метода дифференциальной прогонки, обладающий свойством универсальности: корректность использования этого варианта обусловлена только корректностью формулировки краевой задачи. Алгоритм используется для численного решения более общей проблемы - многоточечной линейной краевой задачи для системы из M_2, $M_2 \geq 1$, обыкновенных дифференциальных уравнений порядка M_3, $M_3 \geq 1$, которую, очевидно, можно представить в виде задачи для системы $M_2 M_3$ уравнений первого порядка. Основные особенности алгоритма были рассмотрены в [2] применительно к двухточечной краевой задаче для системы дифференциальных уравнений второго порядка и многоточечной краевой задаче для уравнения порядка более высокого, чем первый.

Алгоритм реализован в виде программы BPRKR, составленной на языке BASIC. В работе приводится текст программы с пояснениями для пользователя. Для удобства чтения программы при описании алгоритма используются обозначения, близкие к тексту программы.

I. Рассмотрим краевую задачу для системы уравнений:

$$\frac{dy_i}{dx}(x) + \sum_{j=1}^{M_0} Q_t(x) y_j(x) = Q_z(x), \quad x \in [a, b], \tag{I}$$

$$i = 1, 2, \ldots, M_0, \quad t = (i-1)M_1 + j, \quad z = iM_1, \quad M_1 = M_0 + 1,$$

с граничными условиями в точках $s_k \in [a, b]$,

$$\sum_{j=1}^{M_0} H_{k,j}\, y_j(s_k) = H_{k,M_1}, \quad k = 1, 2, \ldots, M_0 . \tag{2}$$

Здесь $Q_t(x), Q_2(x), H_{k,j}$ и H_{k,M_1} – заданные функции и постоянные. Для определенности будем предполагать, что коэффициенты системы (I) принадлежат классу кусочно-непрерывных на $[a,b]$ функций с конечным числом разрывов первого рода, налагая на коэффициенты при необходимости другие ограничения. Некоторые из точек могут быть кратными. В дальнейшем будем подразумевать корректность краевой задачи (I)-(2), понимая под этим существование единственного решения с непрерывной зависимостью от исходных данных.

Согласно методу дифференциальной прогонки, численное решение краевой задачи сводится к последовательности устойчиво решаемых задач Коши с начальными условиями, которые определяются граничными условиями краевой задачи [3-5]. В рассматриваемом варианте независимо строятся на $[a,b]$ решения M_0 задач Коши для одной и той же линейной однородной системы из M_0+1 обыкновенных дифференциальных уравнений первого порядка в соответствии с последовательностью перебора начальных условий в точках s_k:

$$\frac{du_j}{dx}(x) = \sum_{i=1}^{M_0} Q_t(x)\, u_i(x), \quad u_j(s_k) = H_{k,j}, \tag{3}$$
$$j = 1, 2, \ldots, M_1 .$$

ТЕОРЕМА. Пусть в каждой точке $x \in [a,b]$ определены функции $u_{k,j}(x)$ как решения M_0 задач Коши (3):

$$u_{k,j}(x) = u_j(x), \quad u_j(s_k) = H_{k,j}, \quad k = 1, \ldots, M_0, \; j = 1, \ldots, M_1 . \tag{4}$$

Тогда в каждой точке $x \in [a,b]$ существует единственное решение линейной системы из M_0 алгебраических уравнений:

$$\sum_{i=1}^{M_0} u_{k,i}(x)\, y_j(x) = u_{k,M_1}(x), \tag{5}$$

совпадающее с решением краевой задачи (1)-(2).

Действительно, в силу сделанных предположений относительно коэффициентов системы (1) существует единственное решение задачи Коши (3). Кроме того, в точках непрерывности коэффициентов производные $du_{\varepsilon,j}(x)/dx$ ограничены. Поэтому утверждение теоремы можно проверить непосредственно, используя формулы Остроградского-Лиувилля-Якоби [6].

Благодаря линейности и однородности системы (3), имеется возможность дополнить численный метод решения задачи Коши процедурой нормировки с целью повышения устойчивости вычислительного процесса [2,7]. Легко видеть, что равенства (3),(5) допускают определение функций $u_{\varepsilon,j}(x)$ с точностью до пропорционального множителя. Покажем, как можно использовать это свойства, взяв в качестве примера формулы численного интегрирования по методу Рунге-Кутта точности R на отрезке $[x, x+h]$, h - шаг интегрирования [8]. Для этого представим систему (3) в виде:

$$\frac{dU}{dx}(x) = \Phi(x)\,U(x). \qquad (6)$$

Здесь $U(x)$ - вектор размерности M_1 с элементами $u_j(x)$; $\Phi(x)$ - матрица размерности $M_1 \times M_1$ с элементами $\varphi_{j,\ell}(x)$, причем $\varphi_{j,M_1} = 0$. Выпишем формулы метода Рунге-Кутта применительно к системе (6):

$$U(x+h) = V_{R_2+1} + O(h^{R+1}), \quad V_1 = U(x),$$

$$V_s = U(x) + h\sum_{m=1}^{s-1} c_{s-1,m} K_m, \quad s = 2,3,\ldots, R_2+1, \qquad (7)$$

$$K_m = \Phi(\xi_m)V_m, \quad \xi_m = x + hd_m, \quad d_1 = 0,$$

где R_2, d_m и $c_{s-1,m}$ - заданные параметры.

Процедура нормировки связана с определением следующих нормирующих множителей: $j = 1,2,\ldots, M_1$,

$$\alpha(x) = 1/\max_j |u_j(x)|,$$

$$\beta(\xi_m) = 1/\max\Big(1, \sum_{i=1}^{M_0} |V_m^i|\Big), \qquad (8)$$

$$\gamma(\xi_m) = 1/\max\Big\{1, \max_j\Big[\sum_{\ell=1}^{M_0} |\varphi_{j,\ell}(\xi_m)|\Big]\Big\},$$

где V_m^i - элементы вектора V_m. При этом ход вычисления $u(x+h)$ становится иным. На первом шагу вектор $u(x)$ заменяется нормированным вектором $\alpha(x)u(x)$. Далее, k_m вычисляется по преобразованной формуле:

$$k_m = [\gamma(\xi_m)\Phi(\xi_m)][\beta(\xi_m)V_m],$$

а векторы $k_\nu, \nu=1,2,\dots,m-1$, и $u(x)$ заменяются нормированными векторами $\lambda_m k_\nu$ и $\lambda_m u(x)$, где $\lambda_m=\beta(\xi_m)\gamma(\xi_m)$, $\beta(\xi_1)=1$. После вычисления $u(x+h)$ определяется $\alpha(x+h)$ и так далее. Смысл этих преобразований очевиден. Нормировка элементов матрицы $\Phi(\xi_m)$ и вектора V_m исключает участие в операции умножения чисел, по абсолютной величине превышающих единицу. Нормировка вектора $u(x)$ препятствует возможности "зануления" всех его элементов при численном интегрировании системы (6). Заметим, что процедура нормировки (вся или частично) может быть опущена в зависимости от устойчивости выбранного метода решения задачи Коши.

2. Покажем, как к задаче (I), (2) преобразуется многоточечная линейная краевая задача для системы из M_2, $M_2\geq 1$ обыкновенных дифференциальных уравнений порядка M_3, $M_3\geq 1$:

$$\frac{d^{M_3}y_m}{dx^{M_3}}(x)+\sum_{p=1}^{M_3}\sum_{j=1}^{M_2}Q_\tau(x)\frac{d^{p-1}y_j}{dx^{p-1}}(x)=Q_\rho(x), \tag{9}$$

$$x\in[a,b],\quad m=1,2,\dots,M_2,$$

$$\tau=(m-1)M_1+\nu,\quad \nu=(p-1)M_2+j,$$

$$M_0=M_2M_3,\quad M_1=M_0+1,\quad \rho=mM_1,$$

с граничными условиями в точках $s_k\in[a,b]$, $k=1,2,\dots,M_0$:

$$\sum_{p=1}^{M_3}\sum_{j=1}^{M_2}H_{k,\nu}\frac{d^{p-1}y_j}{dx^{p-1}}(s_k)=H_{k,M_1}. \tag{10}$$

Для этого, используя обозначения:

$$y_{i+M_2}(x)=\frac{dy_i}{dx}(x),\quad i=1,2,\dots,M_0-M_2, \tag{11}$$

запишем системы (9), (10) в виде:

$$\frac{dy_i}{dx}(x)+\sum_{j=1}^{M_0} Q_\lambda(x)\, y_j(x)=Q_\rho(x), \tag{12}$$

$$i=M_0-M_2+m,\quad \lambda=(m-1)M_1+j,\quad \rho=mM_1,\quad m=1,\dots,M_2,$$

$$\sum_{j=1}^{M_0} H_{k,j}\, y_j(s_k)=H_{k,M_1} \tag{13}$$

Осталось заметить, что системы (II), (I2) можно представить следующим образом:

$$\frac{dy_i}{dx}(x)+\sum_{j=1}^{M_0} P_t(x)\, y_j(x)=P_z(x), \tag{14}$$

$$i=1,2,\dots,M_0,\quad t=(i-1)M_1+j,\quad z=iM_1.$$

Здесь $P_t(x)$ с учетом (I2) выражается либо через $Q_\lambda(x)$, либо через символ Кронекера $\delta_{j,\ell}$, $\ell=i+M_2$, в зависимости от i :

$$P_t=-\delta_{j,\,i+M_2},\quad P_z=0,\quad \text{если}\quad i\le i\le M_0-M_2,$$

$$P_t=Q_\lambda,\quad P_z=Q_\rho,\quad \text{если}\quad M_0-M_2<i\le M_0.$$

3. Остановимся на некоторых свойствах алгоритма.

1) Точность решения краевой задачи совпадает с точностью решений задач Коши (3).

2) Определение $M_0 M_1$ коэффициентов системы (5) "расщеплено" на независимые решения M_0 задач Коши (3) размерности M_1 .

3) "Цена" за универсальность алгоритма с учетом простоты его реализации на ЭВМ выглядит умеренной: требуется хранить массивы значений коэффициентов и правых частей системы (5) в тех точках $x\in[a,b]$, в которых необходимо знать решение краевой задачи.

4) Если некоторые из коэффициентов системы (I) отличаются сложностью определения, то можно избежать M_0 -кратного повторения трудоемких вычислений этих коэффициентов, связанного с решением M_0 задач Коши, путем организации массивов их значений в точках обращения к правым частям системы (3) при численном интегрировании.

5) Обратим внимание на единообразие в подходе к численному решению широкого класса линейных краевых задач (в частности, достаточно общую формулировку имеют граничные условия), что также является признаком универсальности алгоритма.

4. Программа BPRKR (Boundary - Problem - Runge - Kutt - R) предназначена для численного решения задачи (9),(10). При этом интегрирование системы (3) связано с формулами метода Рунге-Кутта точности R. Приведем краткое описание программы, в котором мы не будем останавливаться на определении обозначений, тождественных ранее употреблявшимся (например, MO и M_0, $C(S,M)$ и $c_{s,m}$). Прежде всего, перечислим нестандартные блоки программы, которые требуется заполнить для использования программы. Текст программы и тестовый пример даны в приложении.

Параметры формул метода Рунге-Кутта точности R содержатся в подпрограмме 510-770 (см. (7)). В ней $R1$ - размерность массива $E(S)$, где $E(S)$ - значения ξ_m без учета их кратности; $G(L)$ - массив размерности $R2$, задающий последовательность обращений к точкам $E(S)$ при вычислении правых частей системы (3) $K(L, J)$ (размерности массива $R2 \times MO$). В программе приведены значения параметров, полученных Батчером [9], $R=6$.

Разбиение отрезка $[a, b]$ (массив $X(1)$ размерности N, $a=X(1)$, $b=X(N)$, где $N-1$ - число разбиений) определяется заданием параметра LO и массивов $L(K)$, $N(K)$ размерностей $LO+1$, $X(1)=L(1)$, $N(1)=1$, $N=N(LO+1)$, $X(N)=L(LO+1)$, $L(1)<L(2)<\ldots<L(LO+1)$. Каждый из отрезков $[L(K), L(K+1)]$ имеет равномерное разбиение с шагом $[L(K+1)-L(K)]/[N(K+1)-N(K)]$.

Граничные условия задаются массивами $S(K)$ размерности MO и $H(P,J)$ размерности $MO \times M1$ (см. 13)).

Определение коэффициентов системы (9) как функций параметра X содержится в подпрограмме 360-500. При заданном X значения функций $Q_\lambda(X)$ присваиваются элементам массива $Q(K)$, $K=1,2,\ldots, M1 \cdot M2$, в том порядке, который указан в (12).

Значения $MO, M1, M2, M3$ и LO содержатся в строке 200. В строки 210-240 заносятся значения $S(J), H(J,1), H(J,2),\ldots, H(J, M1)$, $I = 1,2,\ldots MO$, в указанной последовательности. В строке 250 - значения $L(1), N(1),\ldots, L(LO+1), N(LO+1)$.

Выдача результатов решения краевой задачи формируется в подпрограмме 2520-2630. Значения искомых функций присваиваются элементам массива $U(J,P)$ размерности $N \times MO$ (см. (11), (12)).

Кроме массивов $X(J)$ и $Q(K)$, в строке 10 описаны также массивы $U(I,J)$ размерности $N \times MO \cdot M1$ и $P(S,J)$ размерности $R1 \times MO \times M1$. Необходимость в описании других массивов может возникнуть, начиная с $M1>10$.

Программа содержит тестовый пример, в качестве которого выбрана краевая задача для дифференциального уравнения четвертого порядка [1] (см. формулировку задачи в строках 50-70). В строках 310-340 определены функции, дающие точное решение задачи $T(x)$ и ее производные до третьего порядка включительно. Результаты приведены на рис. 1, где сравниваются точное и численное решения краевой задачи. На рис. 2 даны результаты решения той же задачи с использованием "классических" формул метода Рунге-Кутта, $R=4$ (см. [8]). Там же приведена подпрограмма, содержащая параметры формул.

Стандартная часть программы содержится в строках 780-2360. Перечислим основные входящие в нее блоки.

780-1200 - подпрограмма схемы Гаусса для решения системы линейных алгебраических уравнений с выбором главного элемента.

1220-1310 - определение индекса $T(K)$ точки $S(K)$ как элемента массива $X(I)$, $K=1,2,\dots,MO$.

1380-2270 - процедура численного решения MO задач Коши для системы (3) по формулам метода Рунге-Кутта, дополненная нормировкой (исключая нормировку коэффициентов системы (3)). Значению $Q=1$ (строка 1490) соответствуют решения задач Коши в сторону возрастания $X(I)$ от точки $X(T(P))$, $P=1,2,\dots,MO$, а значению $Q=2$ - решения задач Коши в противоположном направлении. Результаты вычислений присваиваются элементам массива $U(I,J)$ размерности $N\times MO\cdot M1$.

2280-2360 - вычисление искомых функций краевой задачи по схеме Гаусса с выбором главного элемента.

2370-2500 - формирование порядка выдачи результатов в режиме диалога. При этом в строку 2380 заносятся границы интервала, на котором требуется узнать результаты вычислений искомых функций. В строке 2460 - шаг, с которым эти результаты будут выдаваться на печать.

Имеющийся опыт работы с программой BPRKB подтвердил эффективность предлагаемого алгоритма численного решения рассматриваемого класса линейных краевых задач.

Литература

1. БЕЛЛМАН Р., КАЛАБА Р. Квазилинеаризация и нелинейные краевые задачи. - М.: Мир, 1968.

2. ФАДЕЕВ С.И. О численном решении линейной краевой задачи для обыкновенного дифференциального уравнения методом дифференциальной прогонки. - В кн.: Методы сплайн-функций (Вычислительные системы, вып. 75.) Новосибирск, 1978, с. 80-95.

3. ГЕЛЬФАНД И.М., ЛОКУЦИЕВСКИЙ О.В.Метод прогонки для решения разностных уравнений, препринт, см. также С.К.Годунов,В.С.Рябенький. Введение в теорию разностных схем. - М.: Физматгиз,1962.

4. МАРЧУК Г.И. Численные методы расчета ядерных реакторов.М.: Атомиздат, 1958.

5. БАБУШКА И., ВИТАСЕК Э., ПРАГЕР М. Численные процессы решения дифференциальных уравнений.- М.: Мир, 1969.

6. МАТВЕЕВ Н.М. Методы интегрирования обыкновенных дифференциальных уравнений.- М.: Высшая школа, 1967.

7. БОГОЛЮБОВ А.Н., ТЕЛЕГИН В.И. Об одном численном методе решения линейных систем уравнений с трехдиагональной матрицей.-Журн. вычислит.математики и математ.физики. 1974, т. 14, № 3,с.539-541.

8. БЕРЕЗИН И.С., ЖИДКОВ Н.П. Методы вычислений. Т.2.- М.:Физматгиз, 1962.

9. BATCHER I.C. On Runge-Kutta processes of high order. - J. Austral.Math.Soc.,1964,v.4, N 1-2,p.179-194.

Поступила в ред.-изд.отд.
15 ноября 1979 года

BPRKR

```
10  COM X[101],U[101,20],P[4,20],Q[20]
20  REM
30  REM                        "BPRKR"
40  REM
50  REM   Y''''(X)-9*Y'''(X)-79*Y''(X)-159*Y'(X)-90*Y(X)=0
60  REM                     X=0, Y=0, Y''=2
70  REM                     X=1, Y'''=.453234
80  REM
90  REM
100  READ M0,M1,M2,M3,L0
110  FOR I=1 TO M0
120  READ S[I]
130  FOR J=1 TO M1
140  READ H[I,J]
150  NEXT J
160  NEXT I
170  FOR I=1 TO L0+1
180  READ L[I],N[I]
190  NEXT I
200  DATA 4,5,1,4,1
210  DATA 0,1,0,0,0,0
220  DATA 0,0,1,0,0,0
230  DATA 0,0,0,1,0,2
240  DATA 1,0,0,0,1,.453234
250  DATA 0,1,1,11
260  M4=M0-M2
270  M5=M4+1
280  M6=M1*M2
290  M7=M1*M4
300  N=N[L0+1]
310  DEF FNT(X)=EXP(-X)-2*EXP(-2*X)+EXP(-3*X)
320  DEF FNS(X)=-EXP(-X)+4*EXP(-2*X)-3*EXP(-3*X)
330  DEF FNP(X)=EXP(-X)-8*EXP(-2*X)+9*EXP(-3*X)
340  DEF FNQ(X)=-EXP(-X)+16*EXP(-2*X)-27*EXP(-3*X)
350  GOTO 1210
360  Q[1]=-90
370  Q[2]=-159
380  Q[3]=-79
390  Q[4]=-9
400  Q[5]=0
470  FOR K=1 TO M6
480  P[S,M7+K]=Q[K]
490  NEXT K
500  RETURN
510  RESTORE 530
520  READ R1,R2,E[1],E[4],G[1],G[2],G[3],G[4],G[5],G[6],G[7]
530  DATA 4,7,0,1,1,2,3,2,3,2,4
540  Q0=SQR(5)
550  E[2]=(5-Q0)/10
```

```
BPRK9

560  E[3]=(5+Q0)/10
570  C[1,1]=(5-Q0)/10
580  C[2,1]=-Q0/10
590  C[2,2]=(5+2*Q0)/10
600  C[3,1]=(-15+7*Q0)/20
610  C[3,2]=(-1+Q0)/4
620  C[3,3]=(15-7*Q0)/10
630  C[4,1]=(5-Q0)/60
640  C[4,2]=C[5,2]=C[6,2]=C[7,2]=C[7,3]=C[7,4]=0
650  C[4,3]=C[5,4]=C[6,1]=1/6
660  C[4,4]=(15+7*Q0)/60
670  C[5,1]=(5+Q0)/60
680  C[5,3]=(9-5*Q0)/12
690  C[5,5]=(-5+3*Q0)/10
700  C[6,3]=(-55+25*Q0)/12
710  C[6,4]=(-25-7*Q0)/12
720  C[6,5]=5-2*Q0
730  C[6,6]=(5+Q0)/2
740  C[7,1]=C[7,7]=1/12
750  C[7,5]=C[7,6]=5/12
760  PRINT "R=6"
770  RETURN
780  FOR M=1 TO M0
790  C0=0
800  FOR P=1 TO M0
810  FOR Q=1 TO M0
820  IF M=1 THEN 860
830  FOR T=1 TO M-1
840  IF (I[T]=P)+(J[T]=Q)>0 THEN 900
850  NEXT T
860  IF ABS(Z[P,Q])<C0 THEN 900
870  C0=ABS(Z[P,Q])
880  I[M]=P
890  J[M]=Q
900  NEXT Q
910  NEXT P
920  FOR Q=1 TO M1
930  FOR T=1 TO M
940  IF J[T]=Q THEN 1040
950  NEXT T
960  IF ABS(Z[I[M],J[M]]) <= 1.E-37 THEN 1170
970  Z[I[M],Q]=Z[I[M],Q]/Z[I[M],J[M]]
980  FOR P=1 TO M0
990  FOR T=1 TO M
1000  IF I[T]=P THEN 1030
1010  NEXT T
1020  Z[P,Q]=Z[P,Q]-Z[P,J[M]]*Z[I[M],Q]
1030  NEXT P
1040  NEXT Q
1050  NEXT M
1060  Y[J[M0]]=Z[I[M0],M1]
1070  FOR M=1 TO M0-1
1080  Y[J[M0-M]]=Z[I[M0-M],M1]
1090  FOR P=1 TO M
1100  Y[J[M0-M]]=Y[J[M0-M]]-Z[I[M0-M],J[M1-P]]*Y[J[M1-P]]
```

BPRKR

```
1110  NEXT P
1120  NEXT M
1130  FOR P=1 TO M0
1140  U[I,P]=Y[P]
1150  NEXT P
1160  RETURN
1170  FOR P=1 TO M0
1180  U[I,P]=7777
1190  NEXT P
1200  RETURN
1210  GOSUB 510
1220  FOR K=1 TO L0
1230  N1=N[K]
1240  N2=N[K+1]
1250  X[N1]=L[K]
1260  X[N2]=L[K+1]
1270  H=(X[N2]-X[N1])/(N2-N1)
1280  FOR I=N1+1 TO N2-1
1290  X[I]=L[K]+H*(I-N1)
1300  NEXT I
1310  NEXT K
1320  FOR K=1 TO M0
1330  FOR I=1 TO N
1340  IF ABS(X[I]-S[K]) <= .00001 THEN 1360
1350  NEXT I
1360  T[K]=I
1370  NEXT K
1380  FOR P=1 TO M0
1390  C0=0
1400  FOR J=1 TO M1
1410  IF ABS(H[P,J])<C0 THEN 1430
1420  C0=ABS(H[P,J])
1430  NEXT J
1440  P1=M1*(P-1)
1450  FOR J=1 TO M1
1460  U[T[P],P1+J]=H[P,J]/C0
1470  NEXT J
1480  NEXT P
1490  FOR Q=1 TO 2
1500  Q0=(Q=1)-(Q=2)
1510  Q1=(Q=1)*N+(Q=2)
1520  Q2=Q1-Q0
1530  Q3=2*Q0
1540  FOR P=1 TO M0
1550  IF T[P]=Q1 THEN 2260
1560  H=X[T[P]+Q0]-X[T[P]]
1570  FOR S=1 TO R1
1580  X=X[T[P]]+H*E[S]
1590  GOSUB 360
1600  NEXT S
1610  P1=M1*(P-1)
1620  FOR I=T[P] TO Q2 STEP Q0
1630  H=X[I+Q0]-X[I]
1640  FOR J=1 TO M0
1650  V[J]=U[I,P1+J]
```

BOOKR

```
1660  NEXT J
1670  FOR L=1 TO R2
1680  S=J[L]
1690  FOR J=1 TO M1
1700  C0=0
1710  FOR M=M5 TO M0
1720  C0=C0+P[S,M1*(M-1)+J]*V[M]
1730  NEXT M
1740  K[L,J]=C0
1750  NEXT J
1760  IF M3=1 THEN 1800
1770  FOR J=1 TO M4
1780  K[L,M2+J]=K[L,M2+J]-V[J]
1790  NEXT J
1800  FOR J=1 TO M1
1810  K[L,J]=H*K[L,J]
1820  NEXT J
1830  FOR J=1 TO M0+(L=R2)
1840  V[J]=U[I,P1+J]
1850  FOR M=1 TO L
1860  V[J]=V[J]+C[L,M]*K[M,J]
1870  NEXT M
1880  NEXT J
1890  IF L=R2 THEN 2070
1900  C0=0
1910  FOR J=1 TO M0
1920  C0=C0+ABS(V[J])
1930  NEXT J
1940  C0=(1 MAX C0)
1950  IF C0=1 THEN 2060
1960  C0=1/C0
1970  FOR J=1 TO M0
1980  V[J]=C0*V[J]
1990  NEXT J
2000  FOR J=1 TO M1
2010  U[I,P1+J]=C0*U[I,P1+J]
2020  FOR M=1 TO L
2030  K[M,J]=C0*K[M,J]
2040  NEXT M
2050  NEXT J
2060  NEXT L
2070  C0=0
2080  FOR J=1 TO M1
2090  IF ABS(V[J])<C0 THEN 2110
2100  C0=ABS(V[J])
2110  NEXT J
2120  FOR J=1 TO M1
2130  U[I+Q0,P1+J]=V[J]/C0
2140  FOR M=M5 TO M0
2150  P2=M1*(M-1)+J
2160  P[1,P2]=P[R1,P2]
2170  NEXT M
2180  NEXT J
2190  IF I=Q2 THEN 2260
2200  H=X[I+Q3]-X[I+Q0]
```

```
BPPKR

2210  FOR S=2 TO R1
2220  X=Y[I+Q0]+H*E[S]
2230  GOSUB 360
2240  NEXT S
2250  NEXT I
2260  NEXT P
2270  NEXT Q
2280  FOR I=1 TO N
2290  FOR P=1 TO M0
2300  P1=M1*(P-1)
2310  FOR Q=1 TO M1
2320  Z[P,Q]=U[I,P1+Q]
2330  NEXT Q
2340  NEXT P
2350  GOSUB 780
2360  NEXT I
2370  PRINT "X1?,X2";
2380  INPUT O[1],O[2]
2390  FOR K=1 TO 2
2400  FOR I=1 TO N
2410  IF O[K] <= X[I] THEN 2430
2420  NEXT I
2430  N[K]=I
2440  NEXT K
2450  PRINT "H(PRINT)";
2460  INPUT H
2470  GOSUB 2520
2480  PRINT "DONE?,P";
2490  INPUT P
2500  IF P=0 THEN 2640
2510  GOTO 2370
2520  PRINT "   X            Y(X)            Y'(X)         T(X)          T'(X)"
2530  FOR I=N[1] TO N[2] STEP H
2540  PRINT  USING "#,2D.5D2X";X[I]
2550  PRINT  USING "2X4D.6D";U[I,1],U[I,2],FNT(X[I]),FNS(X[I])
2560  NEXT I
2570  PRINT
2580  PRINT "   X            Y''(X)          Y'''(X)       T''(X)        T'''(X)"
2590  FOR I=N[1] TO N[2] STEP H
2600  PRINT  USING "#,2D.5D2X";X[I]
2610  PRINT  USING "2X4D.6D";U[I,3],U[I,4],FNP(X[I]),FNQ(X[I])
2620  NEXT I
2630  RETURN
2640  END
```

Тестовый пример

```
BPRKR

50  REM    Y''''(X)-9*Y'''(X)-79*Y''(X)-159*Y'(X)-90*Y(X)=0
60  REM                         X=0, Y=0, Y''=2
70  REM                         X=1, Y'''=.453234
RUN
BPRKR

R=6
X1?,X2?0,1
H(PRINT)?2
  X            Y(X)          Y'(X)        T(X)          T'(X)
 0.00000       0.000000     0.000000     0.000000      0.000000
 0.20000       0.026902     0.216114     0.026902      0.216115
 0.40000       0.072856     0.223413     0.072856      0.223414
 0.60000       0.111722     0.160068     0.111722      0.160069
 0.80000       0.136254     0.086103     0.136254      0.086103
 1.00000       0.146996     0.024100     0.146996      0.024101

  X            Y''(X)        Y'''(X)      T''(X)        T'''(X)
 0.00000       2.000000    -12.000002     2.000000    -12.000000
 0.20000       0.395476     -4.911527     0.395473     -4.911520
 0.40000      -0.213563     -1.613304    -0.213565     -1.613299
 0.60000      -0.373051     -0.192776    -0.373052     -0.192775
 0.80000      -0.349381      0.331629    -0.349382      0.331632
 1.00000      -0.266719      0.453234    -0.266719      0.453234
DONE?,R?0

DONE
```

Рис.I

```
BPRKR

510  RESTORE 530
520  READ R1,R2,E[1],E[2],E[3],G[1],G[2],G[3],G[4]
530  DATA 3,4,0,.5,1,1,2,2,3
540  READ C[1,1],C[2,1],C[2,2],C[3,1],C[3,2],C[3,3]
550  DATA .5,0,.5,0,0,1
560  C[4,1]=C[4,4]=1/6
570  C[4,2]=C[4,3]=1/3
760  PRINT "R=4"
770  RETURN
RUN
BPRKR

R=4
X1?,X2?0,1
H(PRINT)?2
  X            Y(X)          Y'(X)        Y''(X)        Y'''(X)
 0.00000       0.000000     0.000000     2.000000    -11.999998
 0.20000       0.026914     0.215075     0.395607     -4.911944
 0.40000       0.072867     0.223372    -0.213425     -1.613749
 0.60000       0.111730     0.160038    -0.372944     -0.193131
 0.80000       0.136258     0.086083    -0.349307      0.331389
 1.00000       0.146998     0.024089    -0.266661      0.453234
DONE?,R?0
DONE
```

Рис.2