ОБЪЕДИНЕННЫЙ ИНСТИТУТ ЯДЕРНЫХ ИССЛЕДОВАНИЙ

P11-89-728

Т.П. Пузынина

SLIPS2 – ПРОГРАММА ЧИСЛЕННОГО РЕШЕНИЯ ЗАДАЧИ ШТУРМА–ЛИУВИЛЛЯ ДЛЯ СИСТЕМЫ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

1989

1. Введение

В работе дано описание комплекса программ для решения системы дифференциальных уравнений

$$\varphi^{(1)}(z) \equiv \left[\frac{d^2}{dx^2}\hat{I} - 2\hat{Q}(x)\frac{d}{dx} - \hat{K}(x) + \lambda\hat{\Gamma}(x)\right]\vec{\chi}(x) = 0, \tag{1.1}$$

$$z = \{\lambda, \vec{\chi}(x)\}, \qquad a < x < b,$$

с граничными условиями

$$\varphi^{(1+k)}(z) \equiv \hat{G}^{(1+k)}(\lambda)\vec{\chi}(x)|_{x=x^{(k)}} = 0, \qquad (k = 1, 2)\ , \tag{1.2}$$

$$x^{(1)} = a, \qquad x^{(2)} = b,$$

и условием нормировки

$$\varphi^{(4)}(z) \equiv \int\limits_a^b (\vec{\chi}(x), \hat{\Gamma}(x)\vec{\chi}(x))dx - 1 = 0\ . \tag{1.3}$$

Данный комплекс является развитием программ решения задачи Штурма–Лиувилля /1,2/. В нем, как и в указанных программах, реализованы ньютоновские итерационные схемы вычисления частичных решений задачи (1.1)–(1.3) с выбором итерационного параметра. Среди них – схема с фиксированным сдвигом по собственному значению λ и ортогонализацией вектора собственных функций $\vec{\chi}(x)$, которая обеспечивает сходимость итераций к решению с собственным значением, ближайшим к заданному начальному приближению, от достаточно произвольных начальных приближений для вектора собственных функций.

Представляемая версия SLIPS2 реализует решение задачи (1.1)–(1.3) для системы двух уравнений. Таким образом, в задаче

$$\hat{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},\ \hat{Q}(x) = \{Q_{ij}(x)\},\ \hat{K}(x) = \{K_{ij}(x)\},\ \hat{\Gamma}(x) = \{\gamma_{ij}(x)\},$$

$$\hat{G}^{(1+k)}(\lambda) = \begin{pmatrix} d_{1k}(\lambda)\frac{d}{dx} + f_{1k}(\lambda) & 0 \\ 0 & d_{2k}(\lambda)\frac{d}{dx} + f_{2k}(\lambda) \end{pmatrix}, \tag{1.4}$$

$$d_{ik}^2(\lambda) + f_{ik}^2(\lambda)\ > 0, \qquad (\ i,\ j,\ k = 1, 2),$$

и предполагается, что матрицы (1.4) коэффициентов задачи (1.1)–(1.3) заданы и обеспечивают существование нетривиальных решений

$$z_n^* = \{\lambda_n^*,\ \chi_{1n}^*(x),\ \chi_{2n}^*(x)\}, \qquad n = 0, 1, \ldots\ .$$

Используется разностная аппроксимация уравнений на равномерной сетке

$$\omega_h = \{x_i = a + (i-1)h, \qquad i = \overline{1, N}, \qquad x_N = b, \qquad h = (b-a)/(N-1)\} \qquad (1.5)$$

с точностью $0(h^2)$.

Текст программы написан на языке FORTRAN . Программа использовалась в расчетах уровней энергии и волновых функций мезомолекул и задачи рассеяния в двухуровневом приближении адиабатического представления задачи трех тел с кулоновским взаимодействием [/3,4/], в обратной задаче рассеяния в рамках баргмановского формализма [/5/] и ряде других задач.

Программа может быть эффективно использована как самостоятельно, так и в качестве генератора начальных приближений при численном анализе сходимости различных многоуровневых приближений для квантовомеханических систем. Кроме того, все алгоритмы и программные решения, заложенные в комплекс SLIPS2 , были использованы при создании программ для решения систем произвольного числа уравнений [/6/].

Ниже приводятся описания алгоритмов, структуры комплекса и его программ, а также пример, иллюстрирувщий его применение.

2. Алгоритмы

2.1. Описание итерационного процесса

В работах [/1,2/] подробно изложены идеи, формулы и вычислительные схемы, реализующие непрерывный аналог метода Ньютона (НАМН) решения частичной задачи Штурма–Лиувилля для одного дифференциального уравнения второго порядка.

В данной работе представлены окончательные рабочие схемы и выражения для решения задачи (1.1)–(1.4).

Введем обозначения:

$$\chi'_{1t} = v(x,t) = v_1(x,t) + \mu v_2(x,t), \tag{2.1.1}$$

$$\chi'_{2t} = w(x,t) = w_1(x,t) + \mu w_2(x,t),$$

где μ определяется из условия нормировки (1.3).

Функции $v_i,\ w_i\ (i = 1,2)$ являются решениями двух краевых задач для уравнений

$$v''_1 - 2Q_{11}v'_1 - [K_{11} - \lambda\gamma_{11}]v_1 - [K_{12} - \lambda\gamma_{12}]w_1 - 2Q_{12}w'_1 = -\varphi^{(1)}(z), \tag{2.1.2}$$

$$w''_1 - 2Q_{22}w'_1 - [K_{22} - \lambda\gamma_{22}]w_1 - [K_{21} - \lambda\gamma_{21}]v_1 - 2Q_{21}v'_1 = -\varphi^{(2)}(z)$$

и

$$v''_2 - 2Q_{11}v'_2 - [K_{11} - \lambda\gamma_{11}]v_2 - [K_{12} - \lambda\gamma_{12}]w_2 - 2Q_{12}w'_2 = -\gamma_{11}\chi_1 - \gamma_{12}\chi_2, \tag{2.1.3}$$

$$w''_2 - 2Q_{22}w'_2 - [K_{22} - \lambda\gamma_{22}]w_2 - [K_{21} - \lambda\gamma_{21}]v_2 - 2Q_{21}v'_2 = -\gamma_{22}\chi_2 - \gamma_{21}\chi_1$$

с краевыми условиями для $v_1,\ w_1$

$$d_{11}(\lambda)v'_1 + f_{11}(\lambda)v_1 = -d_{11}(\lambda)\chi'_1 - f_{11}(\lambda)\chi_1, \tag{2.1.2a}$$

$$d_{21}(\lambda)w'_1 + f_{21}(\lambda)w_1 = -d_{21}(\lambda)\chi'_2 - f_{21}(\lambda)\chi_2$$

при $x = a$ и

$$d_{12}(\lambda)v'_1 + f_{12}(\lambda)v_1 = -d_{12}(\lambda)\chi'_1 - f_{12}(\lambda)\chi_1, \tag{2.1.2b}$$

$$d_{22}(\lambda)w'_1 + f_{22}(\lambda)w_1 = -d_{22}(\lambda)\chi'_2 - f_{22}(\lambda)\chi_2$$

при $x = b$ и для $v_2,\ w_2$

$$d_{11}(\lambda)v'_2 + f_{11}(\lambda)v_2 = -d'_{11\lambda}(\lambda)\chi'_1 - f'_{11\lambda}(\lambda)\chi_1, \tag{2.1.3a}$$

$$d_{21}(\lambda)w'_2 + f_{21}(\lambda)w_2 = -d'_{21\lambda}(\lambda)\chi'_2 - f'_{21\lambda}(\lambda)\chi_2$$

при $x = a$ и

$$d_{12}(\lambda)v_2' + f_{12}(\lambda)v_2 = -d_{12\lambda}'(\lambda)\chi_1' - f_{12\lambda}'(\lambda)\chi_1, \qquad (2.1.3b)$$

$$d_{22}(\lambda)w_2' + f_{22}(\lambda)w_2 = -d_{22\lambda}'(\lambda)\chi_2' - f_{22\lambda}'(\lambda)\chi_2$$

при $x = b$.

Из (2.1.2) – (2.1.2b) следует, что

$$v_1 = -\chi_1, \qquad w_1 = -\chi_2. \qquad (2.1.4)$$

Выражение для вычисления μ имеет вид

$$\mu = \frac{1 + \int\limits_a^b \gamma_{11}\chi_1^2 dx + \int\limits_a^b (\gamma_{12}+\gamma_{21})\chi_1\chi_2 dx + \int\limits_a^b \gamma_{22}\chi_2^2 dx}{2\int\limits_a^b \gamma_{11}\chi_1 v_2 dx + \int\limits_a^b (\gamma_{12}+\gamma_{21})(v_2\chi_2 + w_2\chi_1)dx + 2\int\limits_a^b \gamma_{22} w_2\chi_2 dx}\,. \qquad (2.1.5)$$

Итерационный процесс определяется по формулам (k – номер итерации)

$$\chi_1^{k+1} = \chi_1^k + \tau^k v^k = \chi_1^k + \tau^k(v_1^k + \mu^k v_2^k), \qquad (2.1.6)$$

$$\chi_2^{k+1} = \chi_2^k + \tau^k w^k = \chi_2^k + \tau^k(w_1^k + \mu^k w_2^k),$$

$$\lambda^{k+1} = \lambda^k + \tau^k \mu^k.$$

Здесь τ^k – параметр, вычисляемый в зависимости от заданного значения параметра – ключа L_τ по одной из формул:

$$\text{для} \qquad L_\tau = 1 \qquad \tau^k \equiv \tau^0, \qquad (2.1.7.1)$$

т.е. имеет постоянное заданное значение;

$$\text{для} \qquad L_\tau = 2 \qquad \tau^k = \begin{cases} \min(1, 2\tau^{k-1}), & \text{если} \quad \delta^k < \delta^{k-1}, \\ \max(\tau^0, \frac{1}{2}\tau^{k-1}), & \text{если} \quad \delta^k \ge \delta^{k-1}, \end{cases} \qquad (2.1.7.2)$$

$$\text{для} \qquad L_\tau = 3 \qquad \tau^k = \begin{cases} \min(1, \dfrac{\tau^{k-1}\delta^{k-1}}{\delta^k}), & \text{если} \quad \delta^k < \delta^{k-1}, \\ \max(\tau^0, \dfrac{\tau^{k-1}\delta^{k-1}}{\delta^k}), & \text{если} \quad \delta^k \ge \delta^{k-1}, \end{cases} \qquad (2.1.7.3)$$

$$\text{для} \qquad L_\tau = 4 \qquad \tau^k = \frac{\delta^{k-1}}{\delta^{k-1} + \delta^k(1)}\,, \qquad (2.1.7.4)$$

где $\delta^k(1)$ – невязка на k-той итерации для $\tau^k = 1$.

Для $L_\tau = 5$ значение τ^k определяется из минимума последовательности невязок δ_i, вычисленных для значений τ_i на равномерной сетке ω_τ отрезка $[\ 0,\ 1\ ]$ с шагом $\triangle\tau$. Например, для $\tau_i \in \omega_\tau \equiv 0;\ 0.1;\ \ldots 0.9;\ 1$. Это более общий подход, чем для $L_\tau = 4$, но он требует большего объема вычислений, а следовательно, и времени, особенно при больших N .

Все пять алгоритмов прошли проверку при решении широкого круга задач.

Невязки δ^k вычисляются по формуле (наибольшее отклонение)

$$\delta^k = \max_{x\in[a,b]} \left|\varphi^{(1)}(z^k)\right| \tag{2.1.8.1}$$

для (2.1.7.1) – (2.1.7.3) и $L_\tau = 5$, либо по формуле

$$\delta^k = \left\|\varphi_1^{(1)}\ (z^k)\right\|^2 = \int_a^b \left|\varphi^{(1)}\ (z^k)\right|^2 dx \tag{2.1.8.2}$$

для (2.1.7.4) (квадратичная невязка).

Итерационный процесс (2.1.6) прекращается, если достигается заданная точность по невязке, т.е. при выполнении условия

$$\delta^k\ \leq\ \varepsilon\ , \tag{2.1.9}$$

где ε – малое число.

Выдаче окончательных результатов будет предшествовать фраза

SOLUTION OF SYSTEM WITH GIVEN PRECISION EPS= ε.

Итерационный процесс прерывается при условии исчерпания максимального допустимого количества итераций K , если точность (2.1.9) к этому времени не достигнута. В этом случае печатается фраза

RETURN FOR NUMBER OF ITERATIONS K=

и далее печатаются все результаты после K итераций.

2.2. Модифицированный процесс. $\hat{\Gamma}(x) \equiv \hat{I}$

Суть модификации итерационного процесса легко понять, рассматривая ее на уровне эволюционного уравнения

$$\varphi^{(j)'}(z)\frac{dz}{dt} = -\varphi^{(j)}(z); \quad j = 1,2,3,4; \; z = z(t); \quad 0 \le t < \infty, \tag{2.2.1}$$

из которого следуют все формулы пункта 2.1.

Будем рассматривать решение $\bar{z} = \{\bar{\lambda}, \chi_1(x), \chi_2(x)\}$, где $\bar{\lambda}$ – некоторое фиксированное собственное значение. Эволюционное уравнение (2.2.1) будет иметь вид

$$\varphi^{(j)'}(\bar{z})\frac{dz}{dt} = -\varphi^{(j)}(z); \quad j = 1,2,3,4, \tag{2.2.2}$$

что приведет к следующим изменениям в формулах пункта 2.1:

а) в левых частях уравнений (2.1.2) и (2.1.3) вместо λ будет $\bar{\lambda}$,

б) в левых частях краевых условий (2.1.2a) – (2.1.3b) все d_{ij}, f_{ij} будут функциями от $\bar{\lambda}$,

в) в формулах (2.1.3a) и (2.1.3в) $d'_{ij\lambda}$, $f'_{ij\lambda}$ также будут зависеть от $\bar{\lambda}$.

Здесь на каждой итерации применена ортогонализация решения ко всем предыдущим вычисленным решениям

$$\hat{\chi}^k_{1n} = \frac{u^k_{1n}}{\sqrt{A}}\,, \qquad \hat{\chi}^k_{2n} = \frac{u^k_{2n}}{\sqrt{A}}\,, \tag{2.2.3}$$

где

$$u^k_{1n} = \chi^k_{1n} - \sum_{m=0}^{n-1}\{\int_a^b (\chi^k_{1n}\chi^*_{1m} + \chi^k_{2n}\chi^*_{2m})dx\}\chi^*_{1m}\,, \tag{2.2.4}$$

$$u^k_{2n} = \chi^k_{2n} - \sum_{m=0}^{n-1}\{\int_a^b (\chi^k_{1n}\chi^*_{1m} + \chi^k_{2n}\chi^*_{2m})dx\}\chi^*_{2m}\,,$$

$$A = \int_a^b (u^k_{1n})^2 dx + \int_a^b (u^k_{2n})^2 dx\,, \tag{2.2.5}$$

$\vec{\chi}^*_0$, $\vec{\chi}^*_1$, $\ldots$, $\vec{\chi}^*_{n-1}$ – вычисленные с заданной точностью собственные решения.

2.3. Дискретное представление и точность вычислительных схем

Дискретное представление задачи (1.1)– (1.3) осуществляется методом конечных разностей на равномерной сетке ω_h (1.5) с использованием разностных формул порядка $0(h^2)$.

Интегрирование в формулах (2.1.5), (2.1.8.2), (2.2.4) и (2.2.5) выполняется с помощью квадратурной формулы Симпсона.

Полученные дискретные краевые задачи решаются методом прогонки.

Пусть $z_n = \{\lambda_n, \vec{\chi}_n\}$ – решение задачи (1.1) – 1.3), существование которого предполагается; $z_{nh}^k = \{\lambda_{nh}^k, \vec{\chi}_{nh}^k\}$ – решение, полученное по изложенной в пункте 2.1 вычислительной схеме за конечное число итераций k с точностью ε , т.е. при выполнении условия $\delta^k < \varepsilon$, где невязка δ^k определяется по формуле (2.1.8.1) или (2.1.8.2).

Если за z_{nh} обозначить точное решение разностных задач, то в сеточной С – норме можно написать следующую оценку

$$\left\| z_n - z_{nh}^k \right\| \ \le \ \| z_n - z_{nh} \| + \left\| z_{nh} - z_{nh}^k \right\| . \tag{2.3.1}$$

Для первого слагаемого оценка порядка точности дискретного представления задачи (1.1)– (1.3), т.е. $0(h^2)$. Второе слагаемое оценивается как [7]

$$\left\| z_{nh}^k - z_{nh} \right\| \ \sim B_n \delta^k,$$

B_n – константа.

Если $B_n \delta^k << Ch^2$, что можно сделать при задании достаточно малого ε , то точность полученного разностного решения близка к теоретической оценке

$$\left\| z_n - z_{nh}^k \right\| \ \approx C_n h^2.$$

В приведенном тестовом примере это подтверждается путем сравнения полученных результатов на последовательности вдвое сгущающихся узлов разностной сетки ω_h.

3. Описание комплекса программ

Комплекс SLIPS2 состоит из главной программы и набора подпрограмм, которые условно можно разделить на две группы.

К первой группе относятся подпрограммы, которые должны быть составлены пользователем и к которым есть обращение только из главной программы. Сюда относятся, в частности, подпрограммы вычисления сетки ω_h , коэффициентов системы (1.4), начального приближения к решению. В предложенном тестовом пакете это подпрограммы COEFS , SINNZ .

Вторая группа подпрограмм реализует итерационный процесс, описанный в пунктах 2.1 и 2.2. Две подпрограммы из этой группы, D1F1 и D2F2 , должны быть написаны пользователем.

Структура комплекса представлена на рисунке.

```
MAIN
 ├──→ CALL COEFS
 ├──→ CALL SINNZ
 └──→ CALL SLIPS2
        ├──→ CALL DELTA2
        ├──→ CALL PROGS2
        │      ├──→ CALL D1F1
        │      ├──→ CALL D2F2
        │      └──→ CALL LINEQ1
        ├──→ CALL SIMPS
        ├──→ CALL TAUKA2
        │      └──→ CALL DELTA2
        ├──→ CALL NORMS2
        │      └──→ CALL SIMPS2
        └──→ CALL ORTS2
               ├──→ CALL SIMPS2
               └──→ CALL NORMS2
```

3.1. Главная программа и подпрограммы пользователя

В главной программе необходимо указать максимальную размерность NMAX для всех массивов, в которые будут помещаться значения узлов сетки ω_h , коэффициентов системы, собственных функций и пр. Здесь также надо задать значения для a, b, количество узлов сетки NR ($NR \leq NMAX$) и организовать вычисление коэффициентов системы (1.4), сетки ω_h , начальных приближений к решению $z^0 = \{E^0, \vec{\chi}^0\}$, а также присвоить значения фактическим параметрам в обращении к подпрограмме SLIPS2 для получения решения с заданной точностью.

Необходимо ввести три COMMON блока.

COMMON /AMATR/AK11(NMAX), AK12(NMAX), AK21(NMAX), AK22(NMAX),

G11(NMAX), G12(NMAX), G21(NMAX), G22(NMAX) ,

Q11(NMAX), Q12(NMAX), Q21(NMAX), Q22(NMAX) ,

где AKIJ, GIJ, QIJ - массивы для коэффициентов K_{ij}, Γ_{ij}, Q_{ij} в (1.4).

COMMON/BOUND/D1(2), DD1(2), F1(2), DF1(2), D2(2), DD2(2), F2(2), DF2(2) для помещения коэффициентов d_{ij}, f_{ij} из граничных условий $\hat{G}$ в (1.4), а также их производных $d'_{ij\lambda}, f'_{ij\lambda}$.

COMMON /MOD/ MOD, NSOL , где

MOD является параметром, определяющим при MOD = 0 не модифицированный итерационный процесс (пункт 2.1) и модифицированный, описанный в пункте 2.2, при MOD=2.

NSOL – (количество найденных решений) – можно использовать для организации продолжения счета с накоплением найденных решений на диске в режиме MOD=2 , когда необходима ортогонализация функций.

В начале счета NSOL=O .

Подпрограмма COEFS пишется пользователем и предназначается для вычисления коэффициентов (1.4) системы (1.1) в узлах x_i сетки ω_h (1.5), которая также вычисляется здесь.

Обращение

CALL COEFS(R,NR,A,B) ,

где

A, B – левая и правая границы отрезка $[\ a, b\ ]$ изменения переменной x;

NR – количество узлов равномерной сетки ω_h;

R – массив размерности NR , куда помещаются элементы сетки ω_h .

В подпрограмме должен быть использован блок COMMON /AMATR/.

В тестовом примере получение начального приближения к функциям осуществляется обращением к подпрограмме SINNZ с таким набором значений фактических параметров, чтобы получить функции, обращающиеся в нуль только при $x = a$ и $x = b$.

Обращение к подпрограмме SINNZ:

CALL SINNZ(NR,X,Y,NZ,LYA,LYB).

Здесь

NR – размерность массивов X,Y;

X – массив узлов сетки ω_h;

Y – массив значений построенной функции;

NZ – заданное число нулей функции;

LYA – параметр, задающий при значении LYA = 0 ($\neq 0$) нулевое значение функции (ее производной) в точке $x = a$;

LYB – параметр, задающий при значении LYB = 0 ($\neq 0$) нулевое значение функции (ее производной) в точке $x = b$.

3.2. Описание подпрограмм SLIPS2, D1F1, D2F2, DELTA2, PROGS2, TAUKA2, BOUND2, NORMS2, ORTS2 , SIMPS, SIMPS2

Подпрограмма SLIPS2 использует все указанные в заглавии п.3.2 подпрограммы, а также библиотечную подпрограмму LINEQ1 , предназначенную для решения системы линейных алгебраических уравнений (F401, F402-1)[8] .

Обращение:

CALL SLIPS2(A,B,HR,NR,R,AM,EV0,EV,T0,EPS,ITER,NHIT,LST,LR,

HI10,HI20,HI11,HI21,V1,V2,W1,W2,HI12,HI22) ,

где

A, B – интервал определения переменной x;

HR – шаг дискретной сетки ω_h;

NR – число узлов этой сетки;

R – массив узлов сетки ω_h;

AM – физический параметр ($\lambda = 2ME$, при $AM = 0.5$ значение $\lambda = E$);

EV0 – начальное собственное значение;

EV – уточненное собственное значение;

T0 – начальное значение параметра τ^k в процессе (2.1.6);

EPS – заданная точность по невязке в (2.1.9);

ITER – максимальное допустимое число итераций в процессе (2.1.6);

NHIT – шаг по итерациям для печати;

LST – параметр – ключ L_τ выбора τ^k в (2.1.7);

LR – шаг по узлам x_i, для печати;

HI10, HI20 – массивы размерности NR , где находится начальное приближение к решениям χ_1^0, χ_2^0. Сюда же засылаются уточненные решения. Начальные приближения сохраняются в массивах HI11, HI21 размерности NR.

V1, V2, W1, W2, HI12, HI22 – рабочие массивы размерности NR .

В подпрограмме используется блок COMMON/MOD/MOD,NSOL .

Подпрограммы D1F1 и D2F2 используются для вычисления коэффициентов d_{ij}, f_{ij} и их производных по λ при $x = a$ и $x = b$, соответственно.

Обращения:

CALL D1F1(EV,A),

CALL D2F2(EV,B).

Все параметры уже описаны выше. Используется блок

COMMON/BOUND/D1(2), DD1(2), F1(2), DF1(2), D2(2), DD2(2), F2(2), DF2(2).

Подпрограмма DELTA2 предназначена для вычисления невязки δ^k по формулам (2.1.8.1) и (2.1.8.2).

Обращение:

CALL DELTA2 (EV,AM,NR,HR,R,HI11,HI21,DEL,RDEL,NEQ,DEL2).

Первые семь параметров уже описаны.

DEL – величина невязки (отклонения) δ^k в формуле (2.1.8.1);

RDEL – значение узла , в котором достигается максимум отклонения;

NEQ – номер уравнения, в котором достигается максимум отклонения;

DEL2 – величина среднеквадратичной невязки δ^k по формуле (2.1.8.2).

В подпрограмме используется COMMON/AMATR/.

Подпрограмма PROGS2 осуществляет метод прогонки при решении систем (2.1.2) и (2.1.3) с краевыми условиями (2.1.2а), (2.1.2в) и (2.1.3а),(2.1.3в) соответственно.

Обращение:

CALL PROGS2(A,B,NR,V1,V2,W1,W2,HI10,HI20,AM,EV,EVF,HR,C).

Параметры A, B, NR, HI10, HI20, AM, EV, HR описаны выше при обращении к подпрограмме SLIPS2.

V1, V2, W1,W2 – массивы размерности NR для решений систем (2.1.2) и (2.1.3).

EVF – фиксированное собственное значение при использовании модифицированного процесса (MOD=2).

C – параметр, характеризующий корректное (C = 0) решение системы линейных алгебраических уравнений 5-го порядка, появляющейся на правом конце отрезка $[\ a, b\ ]$, или невозможность ее решения (C = 1) при нулевом значении ее определителя.

В подпрограмме используются COMMON блоки /AMATR/ и /BOUND/.

Есть обращения к подпрограммам D1F1 и D2F2 и к библиотечной подпрограмме LINEQ1 /8/.

Подпрограмма TAUKA2 производит вычисление итерационного параметра τ^k в процессе (2.1.6).

Обращение:

CALL TAUKA2(LST,DELKM1,DELK,T0,TKM1,TK,NR,EV,AMJU,Y1,Y2, V1,V2,W1,W2,Y11,Y21,AM,HR,R) ,

где

LST – параметр $L_\tau = 1, 2, \ldots, 5$, определяющий выбор формулы из (2.1.7.1) – (2.1.7.4);

DELKM1, DELK – δ^{k-1} и δ^k соответственно;

T0, TKM1, TK – τ^0, τ^{k-1} и τ^k соответственно;

AMJU – параметр μ^k в (2.1.6), вычисляемый в SLIPS2 по формуле (2.1.5);

Y1 ,Y2, V1, V2, W1, W2, Y11, Y21 – массивы размерности NR .

Остальные параметры описаны выше.

В подпрограмме имеется обращение к подпрограмме DELTA2.

Подпрограмма BOUND2 присваивает нулевые значения первым и последним элементам массивов Y1 и Y2 , если граничные условия заданы именно в таком виде ($d_{ij} = 0$) .

Обращение:

CALL BOUND2(NR,Y1,Y2) , где

Y1, Y2 – массивы размерности NR , содержащие таблицы решений.

В подпрограмме используется COMMON /BOUND/.

Подпрограмма NORMS2 осуществляет нормировку решений по формулам (2.2.3) – (2.2.5).

Обращение:

CALL NORMS2(HR,NR,Y1,Y2).

В ней есть обращение к подпрограмме SIMPS , осуществляющей интегрирование функции по формуле Симпсона.

Подпрограмма ORTS2 осуществляет ортогонализацию функции ко всему набору предыдущих найденных решений и нормировку по формулам (2.2.3) – (2.2.5).

Обращение:

CALL ORTS2(K,NR,HR,Y1,Y2,V1,V2,W1,W2,NT1,NT2,LW) ,

где

K – количество найденных точных решений;

NT1 и NT2 – номера лент (файлов) для хранения собственных функций $\vec{\chi}_i^*$;

LW – параметр – ключ. При LW $\neq 0$ ($= 0$) происходит (не происходит) запись на ленту (диск) нормированных функций из массивов Y1 и Y2;

Y1, Y2, V1, V2, W1, W2 – массивы размерности NR;

В Y1 и Y2 помещаются собственные функции $\chi_1,\ \chi_2$;

V1, V2, W1, W2 – рабочие массивы.

Имеется обращение к SIMPS и NORMS2.

Подпрограмма SIMPS осуществляет вычисление интеграла по формуле Симпсона от произведения двух функций Y и V .

Обращение:

CALL SIMPS(HR,NR,Y,V,S),

где S – значение интеграла.

4. Тестовый пример

В качестве тестового примера взята система вида

$$\chi_1'' - [e^{-x} + 1 - \frac{2}{x}]\chi_1 + \lambda\chi_2 = 0 \ , \qquad (4.1)$$

$$\chi_2'' - [e^{x} + 4 - \frac{4}{x}]\chi_2 + \lambda\chi_1 = 0 \ ,$$

имеющая аналитическое решение

$$\lambda = 1, \qquad \chi_1 = xe^{-x}, \qquad \chi_2 = xe^{-2x} \ . \qquad (4.2)$$

Здесь $Q_{ij} \equiv 0$, $K_{11} = e^{-x} + 1 - \frac{2}{x}$, $K_{12} \equiv K_{21} \equiv 0$, $K_{22} = e^{x} + 4 - \frac{4}{x}$, $\Gamma_{11} \equiv \Gamma_{22} \equiv 0$, $\Gamma_{12} \equiv \Gamma_{21} \equiv 1$.

Краевые условия на любом отрезке [0 , b]

$$\chi_1(0) = 0, \qquad \chi_2(0) = 0, \tag{4.3}$$

$$\chi_1(b) = be^{-b}, \qquad \chi_2(b) = be^{-2b}$$

в виде (1.2), (1.4), требуемом для использования в комплексе и реализуемом в подпрограммах D1F1 , D2F2 , имеют вид ($d_{ij}\chi_i^{'} + f_{ij}\chi_i = 0$) :

$$d_{11} = d_{12} = 0, \qquad f_{11} = f_{12} = 1, \tag{4.4}$$

$$d_{21} = b, \qquad f_{21} = b - 1, \qquad d_{22} = b, \qquad f_{22} = 2b - 1.$$

Скорость вычисления решения: при $a = 0$, $b = 10$, NR = 401 (HR = 0.025), $\tau_0 = 0.1$, $L_\tau = 4$ от начального приближения $\lambda^0 = 0.5$, $\chi_i^0 = \sin\frac{\pi x}{b-a}$
($\delta^0 = 0.8{\cdot}10^3$) потребовалось 4 итерации для получения решения с точностью $\varepsilon = 10^{-5}$ ($\delta^4 \sim 0.6 \cdot 10^{-6}$) и t=3.5 секунд времени центрального процессора CDC-6500. Объем требуемой памяти для счета составляет 53300 слов.

Замечания:

1. В программе $\lambda = 2ME$. При $M = 0.5 \quad \lambda = E$.
2. Для работы программы не требуются значения коэффициентов системы (1.1) на концах отрезка [0 , b] , поэтому в первой таблице стоит символ $-I$, т.е. неопределенность.
3. В приложенной программе с тестом имеется способ уточнения решения с помощью экстраполяции по Ричардсону, используя полученные решения с шагами $h, h/2, h/4$.

RUN LIST OUTPUT

NMAX = 401 A = 0.0 B = .100E+02 NR= 401 HR = .250E-01 NH=40 AM = .5
ITER = 50 NHIT = 5 EPS= .100E-04 LST=4 MOD = 0 NSOL =0
TAU0 = .100E+00 E0= .500E+00

J	R	K11	K12	K21	K22
1	0.	-I	-I	-I	-I
41	.100000E+01	-.632121E+00	0.	0.	.271828E+01
81	.200000E+01	.135335E+00	0.	0.	.938906E+01
121	.300000E+01	.383120E+00	0.	0.	.227522E+02
161	.400000E+01	.518316E+00	0.	0.	.575982E+02
201	.500000E+01	.606738E+00	0.	0.	.151613E+03
241	.600000E+01	.669145E+00	0.	0.	.406762E+03
281	.700000E+01	.715198E+00	0.	0.	.110006E+04
321	.800000E+01	.750335E+00	0.	0.	.298446E+04
361	.900000E+01	.777901E+00	0.	0.	.810664E+04
401	.100000E+02	-I	-I	-I	-I

DELTA0 = .79E+03 AT R = .902500E+01 FOR EQUATION NEQ=2

DEL2= .98E+07 MOD = 0 LST=4

ITER = 0 DELTA = .98E+07 AT R = .903E+01 FOR NEQUATION =2

TAUKA= .100E+00

AMJU= -I EV= .500E+00 MOD=0 LST =4

J	R	HI1	HI2	J	R	HI1	HI2
1	0.0	0.	0.	41	1.0	.977198E-01	.977198E-01
81	2.0	.185874E+00	.185874E+00	121	3.0	.255834E+00	.255834E+00
161	4.0	.300750E+00	.300750E+00	201	5.0	.316228E+00	.316228E+00
241	6.0	.300758E+00	.300750E+00	281	7.0	.255834E+00	.255834E+00
321	8.0	.185874E+00	.185874E+00	361	9.0	.977198E-01	.977198E-01
401	10.0	.113519E-08	.113519E-08				

SOLUTION OF SYSTEM WITH GIVEN PRECISION EPS= .10E-04

ITER = 4 DELTA = .57E-06 AT R = .700E+00 FOR NEQUATION =2

TAUKA= .993E+00

AMJU= .203E-01 EV= .100E+01 MOD=0 LST =4

J	R	HI1	HI2	J	R	HI1	HI2
1	0.0	0.	0.	41	1.0	.693712E+00	.255146E+00
81	2.0	.510424E+00	.691055E-01	121	3.0	.281656E+00	.140284E-01
161	4.0	.138149E+00	.253021E-02	201	5.0	.635263E-01	.427706E-03
241	6.0	.280436E-01	.693694E-04	281	7.0	.120360E-01	.109283E-04
321	8.0	.506037E-02	.168460E-05	361	9.0	.209430E-02	.255547E-06
401	10.0	.856003E-03	.383711E-07				

J	R	Y1ANALYT	Y1	Y2ANALYT	Y2
1	0.00000	0.	0.	0.	0.
41	1.00000	.693680E+00	.693712E+00	.255191E+00	.255146E+00
81	2.00000	.510381E+00	.510424E+00	.690726E-01	.691055E-01
121	3.00000	.281638E+00	.281656E+00	.140219E-01	.140284E-01
161	4.00000	.138145E+00	.138149E+00	.253022E-02	.253021E-02
201	5.00000	.635260E-01	.635263E-01	.428035E-03	.427706E-03
241	6.00000	.280439E-01	.280436E-01	.695138E-04	.693694E-04
281	7.00000	.120362E-01	.120360E-01	.109756E-04	.109283E-04
321	8.00000	.506044E-02	.506037E-02	.169759E-05	.168460E-05
361	9.00000	.209433E-02	.209430E-02	.258461E-06	.255547E-06
401	10.0000	.856069E-03	.856003E-03	.388655E-07	.383711E-07

MFA JINR - DUBNA CDC - 6500

053600 MAXIMUM EXECUTION FL.

3.663 CP SECONDS EXECUTION TIME.

Литература

1. *Пузынин И.В., Пузынина Т.П.* В сб.: Алгоритмы и программы для решения некоторых задач физики. Изд-во KFKI - 74-34, Будапешт, 1974, с.93-111.

2. *Пузынин И.В., Пузынина Т.П., Стриж Т.А.* ОИЯИ, P11-87-332, Дубна, 1987.

3. *Пономарев Л.И., Пузынин И.В., Пузынина Т.П.* ЖЭТФ, 1973, 65, 1(7), с.28-34.

4. *Ponomarev L.I., Puzynin I.V., Puzynina T.P., Somov L.N.* Annals of Phys., 1978, 110, 2, p. 274-286.

5. *Amirkhanov I.V., Puzynin I.V., Puzynina T.P., Zakhariev B.N.* JINR Commun., E4-89-312, Dubna, 1989.

6. *Melezhik V.S., Puzynin I.V., Puzynina T.P., Somov L.N.* J. Comput. Phys., 1984, 54, 2, p. 221-236.

7. *Гавурин М.К.* Изв. ВУЗов, Матем., 1958, 5(6), с.18.

8. *Библиотека программ на ФОРТРАНе и автокоде МАДЛЕН для БЭСМ-6,* т. 2, Б2-11-98-77, ОИЯИ, Дубна, 1977.

Рукопись поступила в издательский отдел 24 октября 1989 года.