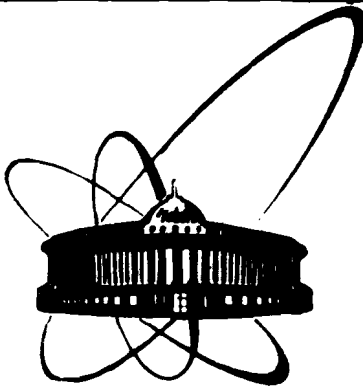


89-728



**ОБЪЕДИНЕННЫЙ  
ИНСТИТУТ  
ЯДЕРНЫХ  
ИССЛЕДОВАНИЙ  
ДУБНА**

П 882

P11-89-728

Т.П.Пузынина

**SLIPS2 - ПРОГРАММА ЧИСЛЕННОГО РЕШЕНИЯ  
ЗАДАЧИ ШТУРМА-ЛИУВИЛЛЯ  
ДЛЯ СИСТЕМЫ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ**

Направлено в журнал "Computer Physics  
Communications"

**1989**

## I. Введение

В работе дано описание комплекса программ для решения системы дифференциальных уравнений

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

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

с граничными условиями

$$\varphi^{(1+k)}(z) \equiv \hat{G}^{(1+k)}(\lambda) \vec{\chi}(x) \Big|_{x=x^{(k)}} = 0; \quad k=1,2; \quad (\text{I.2})$$

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

и условием нормировки

$$\varphi^{(4)}(z) \equiv \int_a^b (\vec{\chi}(x), \hat{\Gamma}(x) \vec{\chi}(x)) dx - 1 = 0. \quad (\text{I.3})$$

Данный комплекс является развитием программ решения задачи Штурма-Лиувилля [1,2]. В нем, как и в указанных программах, реализованы ньютоновские итерационные схемы вычисления частных решений задачи (I.1)-(I.3) с выбором итерационного параметра. Среди них - схема с фиксированным сдвигом по собственному значению  $\lambda$  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 программы SLIPSYS реализует решение задачи (I.1)-(I.3) для системы двух уравнений. Таким образом, в задаче

$$\hat{I} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \hat{Q}(x) = \{q_{ij}(x)\}, \quad \hat{K}(x) = \{K_{ij}(x)\}, \quad \hat{r}(x) = \{r_{ij}(x)\},$$

$$(i, j = 1, 2) \quad (I.4)$$

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

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

и предполагается, что матрицы (I.4) коэффициентов задачи (I.I)-(I.3) заданы и обеспечивают существование нетривиальных решений

$Z_n^* = \{\lambda_n^*, X_{1n}^*(x), X_{2n}^*(x)\}; n = 0, 1, \dots$ . Используется разностная аппроксимация уравнений на равномерной сетке

$$\omega_h = \{x_i = a + (i-1)h, i = \overline{1, N}, x_N = b, h = (b-a)/(N-1)\} \quad (I.5)$$

точности  $O(h^2)$ .

Текст программы написан на языке FORTRAN, пример использования этой версии дан применительно к операционной системе ЭМ

сbc-6500. Программа использовалась в расчетах уровней энергии и волновых функций мезомолекул и задачи рассеяния в двухуровневом приближении адиабатического представления задачи трех тел с кулоновским взаимодействием<sup>3,4/</sup>, в обратной задаче рассеяния в рамках баргмановского формализма<sup>5/</sup> и ряде других задач.

Комплекс программ адаптирован на PC/AT с использованием системы RM/FORTRAN.

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 систем произвольного числа уравнений<sup>6/</sup>.

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

## 2. АЛГОРИТМЫ

### 2.1. Описание итерационного процесса

В работах<sup>1,2/</sup> 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 (I.I)-(I.4).

Введем обозначения

$$X_{1t}' = v(x, t) = v_1(x, t) + \mu v_2(x, t), \quad (2.I.I)$$

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

где  $\mu$  определяется из условия нормировки (I.3).

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

$$v_1'' - 2Q_{11} v_1' - [K_{11} - \lambda \delta_{11}] v_1 - [K_{12} - \lambda \delta_{12}] w_1 - 2Q_{12} w_1' = -\varphi^{(1)}(z), \quad (2.I.2)$$

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

и

$$v_2'' - 2Q_{11} v_2' - [K_{11} - \lambda \delta_{11}] v_2 - [K_{12} - \lambda \delta_{12}] w_2 - 2Q_{12} w_2' =$$

$$- \delta_{11} X_1 - \delta_{12} X_2, \quad (2.I.3)$$

$$w_2'' - 2Q_{22} w_2' - [K_{22} - \lambda \delta_{22}] w_2 - [K_{21} - \lambda \delta_{21}] v_2 - 2Q_{21} v_2' =$$

$$- \delta_{22} X_2 - \delta_{21} X_1$$

с краевыми условиями для  $v_1, w_1$

$$d_{11}(\lambda) v_1' + f_{11}(\lambda) v_1 = -d_{11}(\lambda) X_1' - f_{11}(\lambda) X_1, \quad (2.I.2a)$$

$$d_{21}(\lambda) w_1' + f_{21}(\lambda) w_1 = -d_{21}(\lambda) X_2' - f_{21}(\lambda) X_2$$

при  $x = a$  и

$$d_{12}(\lambda) v_1' + f_{12}(\lambda) v_1 = -d_{12}(\lambda) X_1' - f_{12}(\lambda) X_1, \quad (2.I.2b)$$

$$d_{22}(\lambda) w_1' + f_{22}(\lambda) w_1 = -d_{22}(\lambda) X_2' - f_{22}(\lambda) X_2$$

при  $x = b$

и для  $v_2, w_2$

$$d_{11}(\lambda)v_2' + f_{11}(\lambda)v_2 = -d_{11\lambda}'(\lambda)x_1' - f_{11\lambda}'(\lambda)x_1, \quad (2.1.3a)$$

$$d_{21}(\lambda)w_2' + f_{21}(\lambda)w_2 = -d_{21\lambda}'(\lambda)x_2' - f_{21\lambda}'(\lambda)x_2$$

при  $x=a$  и

$$d_{12}(\lambda)v_2' + f_{12}(\lambda)v_2 = -d_{12\lambda}'(\lambda)x_1' - f_{12\lambda}'(\lambda)x_1, \quad (2.1.3b)$$

$$d_{22}(\lambda)w_2' + f_{22}(\lambda)w_2 = -d_{22\lambda}'(\lambda)x_2' - f_{22\lambda}'(\lambda)x_2$$

при  $x=b$ .

Из (2.1.2)-(2.1.2в) следует, что

$$v_1 = -x_1, \quad w_1 = -x_2. \quad (2.1.4)$$

Выражение для вычисления  $\mu$  имеет вид

$$\mu = \frac{1 + \int_a^b \delta_{11} x_1^2 dx + \int_a^b (\delta_{12} + \delta_{21}) x_1 x_2 dx + \int_a^b \delta_{22} x_2^2 dx}{2 \int_a^b \delta_{11} x_1 v_2 dx + \int_a^b (\delta_{12} + \delta_{21}) (v_2 x_2 + w_2 x_1) dx + 2 \int_a^b \delta_{22} w_2 x_2 dx} \quad (2.1.5)$$

Итерационный процесс определяется по формулам ( $k$ -номер итерации)

$$x_1^{k+1} = x_1^k + \tau^k v_1^k = x_1^k + \tau^k (v_1^k + \mu^k v_2^k), \quad (2.1.6)$$

$$x_2^{k+1} = x_2^k + \tau^k w_1^k = x_2^k + \tau^k (w_1^k + \mu^k w_2^k),$$

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

Здесь  $\tau^k$  - параметр, вычисляемый в зависимости от заданного значения параметра - ключа  $L_\tau$  по одной из формул:

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

т.е. имеет постоянное заданное значение

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

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

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

где  $\delta^k(1)$  - невязка

на  $k$ -той итерации для  $\tau^k = 1$ ; для  $L_\tau = 5$  значение  $\tau^k$  определяется из минимума последовательности невязок  $\delta_i$ , вычисленных для значений  $\tau_i$  на равномерной сетке  $\omega_\tau$  отрезка  $[0,1]$  с шагом  $\Delta\tau$ . Например, для  $\tau_i \in \omega_\tau \equiv \{0; 0,1; \dots; 0,9; 1\}$ . 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  $\delta^k$  вычисляются по формуле

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

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

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

для (2.1.7.4) (квадратичная невязка).

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

$$\delta^k \leq \varepsilon, \quad (2.1.9)$$

где  $\varepsilon$  - малое число.

Выдаче окончательных результатов будет предшествовать фраза  
SOLUTION OF SYSTEM WITH GIVEN ACCURACY EPS=  $\varepsilon$

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 $\hat{\Gamma}(z) \equiv \hat{I}$

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

$$\varphi^{(j)'}(z) \frac{dz}{dt} = -\varphi^{(j)}(z), \quad j=1-4, \quad (2.2.1)$$

$$z = z(t), \quad 0 \leq t < \infty,$$

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

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

что приведет к следующим изменениям в формулах пункта 2.1:

- а) в левых частях уравнений (2.1.2) и (2.1.3) вместо  $\lambda$  будет  $\bar{\lambda}$ ,
- б) в левых частях краевых условий (2.1.2а)-(2.1.3в) все  $d_{ij}$ ,  $f_{ij}$  будут функциями от  $\bar{\lambda}$ ,
- в) в формулах (2.1.3а) и (2.1.3в)  $d'_{ija}$ ,  $f'_{ija}$  также будут зависеть от  $\bar{\lambda}$ .

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

$$\hat{\chi}_{1n}^k = \frac{u_{1n}^k}{\sqrt{A}}, \quad \hat{\chi}_{2n}^k = \frac{u_{2n}^k}{\sqrt{A}}, \quad (2.2.3)$$

где

$$u_{1n}^k = \chi_{1n}^k - \sum_{m=0}^{n-1} \left\{ \int_a^b (\chi_{1n}^k \chi_{1m}^* + \chi_{2n}^k \chi_{2m}^*) dx \right\} \chi_{1m}^*, \quad (2.2.4)$$

$$u_{2n}^k = \chi_{2n}^k - \sum_{m=0}^{n-1} \left\{ \int_a^b (\chi_{1n}^k \chi_{1m}^* + \chi_{2n}^k \chi_{2m}^*) dx \right\} \chi_{2m}^*.$$

$$A = \int_a^b (u_{1n}^k)^2 dx + \int_a^b (u_{2n}^k)^2 dx, \quad (2.2.5)$$

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

### 2.3. Дискретное представление и точность вычислительных

#### схем

Дискретное представление задачи (I.1)-(I.3) осуществляется методом конечных разностей на равномерной сетке  $\omega_h$  (I.5) с использованием разностных формул порядка  $O(h^2)$ .

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 (I.1)-(I.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  $\varepsilon$ , т.е. при выполнении условия  $\delta^k < \varepsilon$ , где невязка  $\delta^k$  определяется по формуле (2.1.8.1) или (2.1.8.2).

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

$$\| z_n - z_{nh}^k \| \leq \| z_n - z_{nh} \| + \| z_{nh} - z_{nh}^k \|. \quad (2.3.1)$$

Для первого слагаемого оценка порядка точности дискретного представления задачи (I.1)-(I.3), т.е.  $O(h^2)$ . Второе слагаемое оценивается как

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

$B_n$  - константа.

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

$$\|Z_n - Z_{nh}^k\| \approx C_n h^2.$$

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

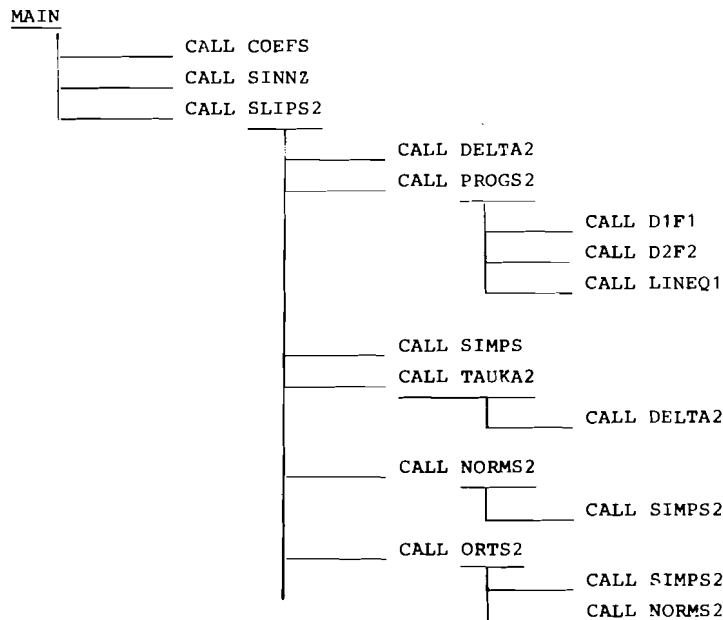
### 3. Описание комплекса программ

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  $\omega_h$ , коэффициентов системы (I.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.



### 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  $\omega_h$ , коэффициентов системы, собственных функций и пр. Здесь также надо задать значения для  $a, b$ , количество узлов сетки NR (NR ≤ NMAX) и организовать вычисление коэффициентов системы (I.4), сетки  $\omega_h$ , начальных приближений к решению  $Z^0 = \{E^0, \bar{Y}^0\}$ , 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  $K_{ij}, \gamma_{ij}, Q_{ij}$  в (I.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 (I.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=0.

Подпрограмма COEFS пишется пользователем и предназначена для вычисления коэффициентов (I.4) системы (I.1) в узлах  $x_i$  сетки  $\omega_h$  (I.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  $\omega_h$

R - массив размерности NR, куда помещаются элементы сетки  $\omega_h$ .

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,LVB).

Здесь

- NR - размерность массивов X,Y
- X - массив узлов сетки  $\omega_h$
- Y - массив значений построенной функции
- NZ - заданное число нулей функции
- LYA - параметр, задающий при значении LYA=0(≠0) нулевое значение функции (ее производной) в точке  $x = a$
- LVB - параметр, задающий при значении LVB=0(≠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 заголовке п.3.2 подпрограммы, а также библиотечную подпрограмму LINEQ1, предназначенную для решения системы линейных алгебраических уравнений (F401,402-1).

Обращение:

CALL SLIPS2(A,B,HR,NR,R,AM,EVO,EV,TO,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  $\omega_h$
- NR - число узлов этой сетки
- R - массив узлов сетки  $\omega_h$
- AM - физический параметр ( $\lambda = 2ME$ , при  $AM=0.5 \lambda = E$ ).
- EVO - начальное собственное значение
- EV - уточненное собственное значение
- TO - начальное значение параметра  $\tau^k$  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  $L_\tau$  выбора  $\tau^k$  в (2.1.7)
- LR - шаг по узлам  $x_i$  для печати
- HI10,HI20 - массивы размерности NR, где находится начальное приближение к решениям  $X_1^0, X_2^0$ .

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  $\lambda$  при  $x = a$  и  $x = b$  соответственно.

Обращение:

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  $\delta^k$  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)  $\delta^k$  в формуле (2.1.8.1)
- RDEL } - значение узла и номера уравнения, в которых достигается
- NEQ } максимум отклонения;
- DEL2 - величина среднеквадратичной невязки  $\delta^k$  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 (2.1.2a), (2.1.2b) и (2.1.3a), (2.1.3b) 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 под-  
 программе LINEQ1<sup>18/</sup>.

Подпрограмма TAUKA2 производит вычисление итерационного  
 параметра  $\tau^k$  в процессе (2.1.6).

Обращение:

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

где

LST - параметр  $L_\tau = 1, 2, \dots, 5$ , определяющий выбор формулы из  
 (2.1.7.1)-(2.1.7.4)

DELKM1, DELK - есть  $\delta^{k-1}$  и  $\delta^k$  соответственно

то TKM1, TK -  $\tau^{\sigma \tau^{k-1}}$  и  $\tau^k$  соответственно

AMJU - параметр  $\mu^k$  в (2.1.6), вычисляемый в SLIP2 по форму-  
 ле (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 ( $a_{ij}=0$ ).

Обращение:

CALL BOUND2(NR, Y1, Y2) ,

где Y1, Y2 - массивы размерности NR, содержащие таблицы решений.  
 В подпрограмме используется COMMON блок /BOUND/ .

Подпрограмма NORMS2 осуществляет нормировку решений по форму-  
 лам (2.2.3)-(2.2.5).

Обращение:

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

В ней есть обращение к подпрограмме SIMPS, осуществляющей интегри-  
 рование 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 (место на диске) для хранения собствен-  
 ных функций  $\tilde{X}_i^*$

LW - параметр-ключ. При LW#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  $X_1, X_2$  .

V1, V2, W1, W2 - рабочие массивы.

Имеется обращение к SIMPS2 и NORMS2 .

Подпрограмма SIMPS осуществляет вычисление интеграла по  
 формуле Симпсона от произведения двух функций  $u$  и  $v$  .

Обращение:

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

S - значение интеграла.

#### 4. Тестовый пример

В качестве тестового примера взята система вида

$$X_1'' - [e^{-x} + 1 - \frac{2}{x}] X_1 + \lambda X_2 = 0, \quad (4.1)$$

$$X_2'' - [e^x + 4 - \frac{4}{x}] X_2 + \lambda X_1 = 0,$$

имеющая аналитическое решение

$$\lambda = 1, \quad X_1 = x e^{-x}, \quad X_2 = x e^{-2x} \quad (4.2)$$

Краевые условия на любом отрезке  $[0, b]$

$$X_1(0) = 0, \quad X_2(0) = 0, \quad (4.3)$$

$$X_1(b) = b e^{-b}; \quad X_2(b) = b e^{-2b}$$

в виде (1.2), (1.4), требуемом для использования в комплексе и реали-  
 зуемом в подпрограммах D1F1, D2F2, имеют вид ( $dX' + fX=0$ ):

$$d_{11} = d_{12} = 0; \quad f_{11} = f_{12} = 1 \quad (4.4)$$

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

Скорость вычисления решения: при  $a=0$ ,  $b=10$ ,  $NR=401$  ( $HR=0.025$ )  
 $\tau_0=0,1$ ,  $L_\tau=4$  от начального приближения  $\lambda^0=0,5$   $X_i^0 = \sin \frac{\pi x}{b-a}$  ( $\delta^0=0,8 \cdot 10^3$ )  
 потребовалось 4 итерации для получения решения с точностью  $\epsilon = 10^{-5}$   
 ( $\delta^4 \sim 0,6 \cdot 10^{-6}$ ) и  $t=3,5$  секунд времени центрального процессора  
 СДС-6500. Объем требуемой памяти для счета составляет 53300 слов.



PRINT LIST OUTPUT

Литература

NMAX= 401 A=0.0 B= .100E+00 NR= 401 HR= .250E-01 NH=40 AM= .5  
 ITER= 50 N=11 E= 5 EPS= .100E-04 LST=4 MOD=0 NSOL= 0  
 TAU0= .100E+00 E0= .500E+00

J	R	K11	K12	K21	K22
1	0.				
41	1.00000E+00	-1.632121E+00	0.	0.	.271828E+01
81	2.00000E+00	-1.353335E+00	0.	0.	.938906E+01
121	3.00000E+00	-1.363120E+00	0.	0.	.227522E+02
161	4.00000E+00	-1.518315E+00	0.	0.	.575982E+02
201	5.00000E+00	-1.606738E+00	0.	0.	.151613E+03
241	6.00000E+00	-1.689145E+00	0.	0.	.406762E+03
281	7.00000E+00	-1.715198E+00	0.	0.	.110006E+04
321	8.00000E+00	-1.750335E+00	0.	0.	.298446E+04
361	9.00000E+00	-1.777901E+00	0.	0.	.810664E+04
401	1.00000E+00				

DELTA= .98E+07 ON R= .902500E+01 FOR EQUATION NED=2  
 DEL2= .98E+07 MOD=0 LST=4

ITER= 0 DELTA= .98E+07 ON R= .903E+01 ON EQUATION=2 TAU0= .100E+00  
 AMJU= -1 EV= .500E+00 MOD=0 LST=4

J	R	H11	H12	J	R	H11	H12
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	.300750E+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 ACCURACY EPS= .10E-04

ITER= 4 DELTA= .57E-06 ON R= .700E+00 ON EQUATION=2 TAU0= .993E+00  
 AMJU= .203E-01 EV= .100E+01 MOD=0 LST=4

J	R	H11	H12	J	R	H11	H12
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	Y1ANALIT	Y1	Y2ANALIT	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	10.00000	.856069E-03	.856003E-03	.388655E-07	.383711E-07

MFA JINR-DUBNA CDC-6500

053600 MAXIMUM EXECUTION FL.

3.663 CF SECONDS EXECUTION TIME.

1. Пузынин И.В., Пузынина Т.П. В сб.: Алгоритмы и программы для решения некоторых задач физики. Изд-во КФКИ - 74-34, Будапешт, 1974, с.93-III.
2. Пузынин И.В., Пузынина Т.П., Стриж Т.А. ОИЯИ, ПИ-87-332, Дубна, 1987.
3. Пономарев Л.И., Пузынин И.В., Пузынина Т.П. ЖЭТФ, 1973, 65, I(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, БЭ-II-98-77, ОИЯИ, Дубна, 1977.

Рукопись поступила в издательский отдел  
 24 октября 1989 года.