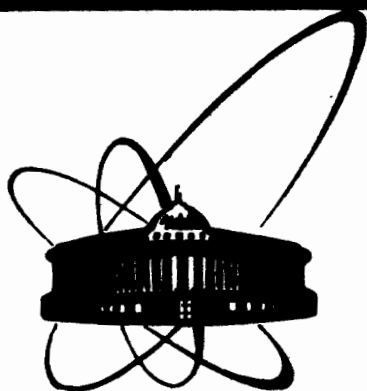


87-332.



**сообщения
объединенного
института
ядерных
исследований
дубна**

P11-87-332

И.В.Пузынин, Т.П.Пузынина, Т.А.Стриж

**SLIPN4 – ПРОГРАММА
ДЛЯ ЧИСЛЕННОГО РЕШЕНИЯ
ЗАДАЧИ ШТУРМА-ЛИУВИЛЛЯ**

1987

I. Введение

SLIPN4 - комплекс программ, являющийся развитием программы SLIP1^{/I/}, предназначенный для численного решения задачи Штурма-Лиувилля в виде

$$\varphi^{(1)}(\lambda, y) \equiv \left[\frac{d^2}{dx^2} + 2p(x) \frac{d}{dx} + q(x) - \lambda r(x) \right] y(x) = 0, \quad (1a)$$

$$a < x < b,$$

$$\varphi^{(j)}(\lambda, y) \equiv \left[d_{j-1}(\lambda, x) \frac{d}{dx} + f_{j-1}(\lambda, x) \right] y(x) \Big|_{x=l_j} = 0, \quad (1b, c)$$

$$j = 2, 3; \quad l_2 = a, \quad l_3 = b.$$

Здесь $p(x), q(x), r(x), d_{j-1}(\lambda, x), f_{j-1}(\lambda, x)$ - заданные функции, обеспечивающие существование нетривиальных собственных решений $y = y_n(x)$ граничной задачи (I), которым соответствуют собственные значения параметра $\lambda = \lambda_n$. Функции d_{j-1}, f_{j-1} непрерывно дифференцируемы по λ и $d_{j-1}^2 + f_{j-1}^2 > 0$.

Программа SLIPN4 по заданному начальному приближению $\{ \lambda_n^0, y_n^0(x) \}$ вычисляет частичное решение $\{ \lambda_n^*, y_n^*(x) \}$ задачи (I), где n - фиксировано, $x \in \omega_h$, ω_h - равномерная разностная сетка на отрезке $a \leq x \leq b$ с шагом h . Точность вычисления является величиной $O(h^4)$ и совпадает с точностью используемой разностной аппроксимации задачи (I)^{/2/}.

Основные особенности комплекса SLIPN4 по сравнению с программой SLIP1 заключаются в следующем.

1. В комплекс включена программа вычисления начальных приближений $\{ \lambda_n^0, y_n^0(x) \}$ с помощью модифицированного алгоритма^{/3/}, основанного на методе Ньютона для нахождения корней полинома с удалением вычисленных корней^{/4/}. Это позволяет последовательно вычислять собственные значения λ_n^* и соответствующие им собственные функции $y_n^*(x)$ в достаточно широкой части спектра задачи (I).

2. Используется разностная аппроксимация на сетке ω_h для граничной задачи (I) с точностью $O(h^4)$. Она построена таким образом, что почти во всех внутренних узлах сетки является трехточечной. Это дает возможность решать задачу (I) с точностью $O(h^4)$ без существенного изменения структуры программы SLIP1.

3. В комплекс включена программа, реализующая модифицированную ньютоновскую схему с дополнительной ортогонализацией найденного приближенного собственного решения. Эта программа позволяет одновременно решить задачи нахождения начального приближения и его уточнения в рамках единой вычислительной схемы.

Работа комплекса SLIPN4 проверена на достаточно широком наборе тестовых задач^{/5/}. Кроме того, комплекс успешно использовался при решении разнообразных физических задач^{/6/}.

Ниже дается краткое описание алгоритмов, используемых в программах комплекса SLIPN4, и обсуждаются вопросы анализа точности результатов. Приведено описание программ и их параметров, рассмотрены варианты использования комплекса.

2. Алгоритмы

2.1. Алгоритм вычисления начального приближения

Алгоритм вычисления начального приближения основан на решении двух задач Коши. Для первой задачи Коши ставятся начальные условия в точке $x=a$, где условие (Iв) дополняется условием на производную. Аналогичное условие для второй задачи ставится в точке $x=b$.

Для приближенного решения этих задач применяется трехточечная разностная схема второго порядка. Рекуррентные соотношения, полученные с помощью разностных схем, позволяют последовательно находить значения функции y в узлах сетки ω_n при заданном значении λ .

Если граничные условия (Iв,с) полиномиально зависят от λ , что выполняется для достаточно широкого круга задач, то упомянутые выше рекуррентные соотношения также определяют полиномы от λ , степени которых зависят от числа узлов сетки. Задача об отыскании собственного значения сводится тогда к отысканию корней полинома, появляющегося из условия равенства логарифмических производных для решений двух задач Коши в некоторой внутренней точке x_M сетки ω_n .

$$\frac{y'_n(x_M)_{лев}}{y_n(x_M)_{лев}} = \frac{y'_n(x_M)_{прав}}{y_n(x_M)_{прав}}, \quad (2.1.1)$$

$$T(\lambda) \equiv y'_n(x_M)_{лев} \cdot y_n(x_M)_{прав} - y'_n(x_M)_{прав} \cdot y_n(x_M)_{лев} = 0. \quad (2.1.2)$$

Корни полинома $T(\lambda)$ находятся по методу Ньютона с исключением уже найденных корней (k - номер итерации)

$$\lambda_n^{k+1} = \lambda_n^k - \frac{T(\lambda_n^k)}{T'_\lambda(\lambda_n^k) - T(\lambda_n^k) \sum_{j=1}^{k-1} \frac{1}{\lambda_n^k - \lambda_j^k}}. \quad (2.1.3)$$

Итерационный процесс (2.1.3) прекращается при одновременном выполнении двух условий

$$\left| \lambda_n^{k+1} - \lambda_n^k \right| \leq \varepsilon_T \quad \text{и} \quad \left| T(\lambda_n^{k+1}) \right| \leq \varepsilon_T, \quad (2.1.4)$$

где $\varepsilon_T > 0$ заданное малое число. После определения λ_n с требуемой точностью мы одновременно получаем и приближенное значение соответствующей собственной функции $y_n(x_1)$.

2.2. Алгоритм уточнения начального приближения

Заданное или вычисленное с помощью алгоритма п.2.1 начальное приближение λ^0 , $y^0(x)$ уточняется с помощью итерационной ньютоновской процедуры для задачи Штурма-Лиувилля^{/1/}.

При этом задача (I) дополняется условием нормировки

$$\varphi^{(4)}(\lambda, y) \equiv \int_a^b y^2(x) dx + c^2 \int_b^\infty \Psi^2(\lambda, x) dx - 1 = 0, \quad (2.2.1)$$

которое дает возможность учитывать особенности тех задач, в которых решение ищется на полубесконечном интервале $[a, \infty)$. Здесь

$\Psi(\lambda, x)$ - асимптотическое выражение для искомой собственной функции $y(x)$ при $x \rightarrow \infty$, c - константа шивки решения в некоторой достаточно удаленной точке $x=b$,

$$y(b) = c \Psi(\lambda, b). \quad (2.2.2)$$

Если задача рассматривается на конечном отрезке $[a, b]$, то следует положить $c=0$. Один шаг с номером k итерационного процесса заключается в следующем:

I. Решить краевую задачу для функции ψ_k :

$$\varphi^{(1)}(\lambda_k, \psi_k) = s_{2k}(x), \quad (2.2.3)$$

$$\varphi^{(2)}(\lambda_k, \psi_k) = a_{2k},$$

$$\varphi^{(3)}(\lambda_k, \psi_k) = b_{2k},$$

где $s_{2k}(x) = r(x)y_k(x)$,

$$a_{2k} = -[d'_{1\lambda} y'_k(a) + f'_{1\lambda} y_k(a)], \quad (2.2.4)$$

$$b_{2k} = -[d'_{2\lambda} y'_k(b) + f'_{2\lambda} y_k(b)].$$

при известных с предыдущего шага ($k-1$) значениях $\lambda_k, y_k(x)$.

2. Вычислить поправку μ_k для собственного значения λ_k с помощью выражения

$$\mu_k = \frac{\int_a^b y_k^2(x) dx - c^2 \int_a^b \psi^2(\lambda_k, x) dx + 1}{2 \left(\int_a^b y_k(x) v_k(x) dx + c^2 \int_a^b \psi(\lambda_k, x) \psi'_\lambda(\lambda_k, x) dx \right)} \quad (2.2.5)$$

3. Получить следующее приближение :

$$\lambda_{k+1} = \lambda_k + \tau_k \mu_k, \quad (2.2.6)$$

$$y_{k+1}(x) = (1 - \tau_k) y_k(x) + \tau_k y_k^*(x)$$

Итерационный параметр τ_k вычисляется с помощью алгоритмов, приведенных ниже. Итерации прекращаются при выполнении условия

$$\delta_k < \varepsilon, \quad (2.2.7)$$

где невязка δ_k может определяться как

$$\delta_k = \max_j \max_{x \in [a, b]} |\varphi^{(j)}(\lambda_k, y_k(x))|, \quad j=1, 2, 3, \quad (2.2.8)$$

или

$$\delta_k = \left[\int_a^b \varphi^{(1)2}(\lambda_k, y_k(x)) dx \right]^{1/2}, \quad (2.2.9)$$

$\varepsilon > 0$ заданное малое число.

2.3. Модифицированный алгоритм

Алгоритм, изложенный в п.2.2., может быть изменен таким образом, что будет одновременно и находить, и уточнять приближенное решение задачи (I), (2.2.I). За основу взята модифицированная ньютоновская схема, в которой на каждом шаге фиксируется сдвиг $\bar{\lambda}$ по собственному значению.

В этом случае на каждом шаге (с номером k) итерационного процесса для нахождения решения $\{ \lambda_n^*, y_n^*(x) \}$ требуется выполнить следующее.

I. Решить две краевые задачи, отличающиеся правыми частями, относительно функций $v_k^{(1)}(x)$ и $v_k^{(2)}(x)$,

$$\varphi^{(1)}(\bar{\lambda}, v_k^{(1)}) = s_k^{(i)}(x),$$

$$\varphi^{(2)}(\bar{\lambda}, v_k^{(1)}) = a_{ik},$$

$$\varphi^{(3)}(\bar{\lambda}, v_k^{(1)}) = b_{ik}, \quad i=1, 2, \quad \text{где} \quad (2.3.1)$$

$$s_k^{(1)} = -\varphi^{(1)}(\lambda_k, y_k), \quad s_k^{(2)} = x(x) y_k(x),$$

$$a_{1k} = -\varphi^{(2)}(\lambda_k, y_k(a)), \quad a_{2k} = -[d_{1\lambda}'(\bar{\lambda}) y_k'(a) + f_{1\lambda}'(\bar{\lambda}) y_k(a)],$$

$$b_{1k} = -\varphi^{(3)}(\lambda_k, y_k(b)), \quad b_{2k} = -[d_{2\lambda}'(\bar{\lambda}) y_k'(b) + f_{2\lambda}'(\bar{\lambda}) y_k(b)] \quad (2.3.2)$$

при известных с предыдущего шага (с номером $k-1$) значениях

$\lambda_k, y_k(x)$.

2. Вычислить поправку μ_k к собственному значению по формуле

$$\mu_k = \frac{1 - \int_a^b y_k^2(x) dx - c^2 \int_a^b \psi^2(\lambda_k, x) dx - 2 \int_a^b y_k(x) v_k^{(1)}(x) dx}{2 \left(\int_a^b y_k(x) v_k^{(2)}(x) dx + c^2 \int_a^b \psi(\bar{\lambda}, x) \psi'_\lambda(\bar{\lambda}, x) dx \right)} \quad (2.3.3)$$

3. Найти следующие приближения

$$\lambda_{k+1} = \lambda_k + \tau_k \mu_k, \quad (2.3.4)$$

$$y_{k+1}(x) = y_k(x) + \tau_k (v_k^{(1)}(x) + \mu_k v_k^{(2)}(x)).$$

4. Выполнить ортогонализацию приближения $y_{k+1}(x)$ к собственной функции $y_n^*(x)$ по отношению ко всем найденным собственным функциям $y_m^*(x)$, $m=0, 1, \dots, n-1$, по формуле

$$u_{k+1}(x) = y_{k+1}(x) - \sum_{m=0}^{n-1} \left(\int_a^b y_{k+1}(x) y_m^*(x) dx \right) y_m^*(x). \quad (2.3.5)$$

5. Нормировать приближенную собственную функцию

$$y_{k+1}(x) = u_{k+1}(x) \left[\int_a^b u_{k+1}^2(x) dx \right]^{-1/2}. \quad (2.3.6)$$

Изложенная процедура является обобщением метода обратных итераций со сдвигом $\bar{\lambda}$. Она обеспечивает при довольно грубых приближениях к собственной функции сходимость к решению $\{ \lambda_n^*, y_n^*(x) \}$, где

λ_n^* - ближайшее к сдвигу $\bar{\lambda}$ собственное значение. Операции (2.3.5), (2.3.6) позволяют дополнительно подавлять вычислительные погрешности.

Изменение сдвига $\bar{\lambda}$ дает возможность сходиться к решению с нужным номером n . В качестве управления изменением сдвига можно принять следующую процедуру.

Для нахождения решения $\{\lambda_0^*, y_0^*(x)\}$ в качестве $\bar{\lambda}$ можно задать соответствующую границу спектра задачи (I). Далее в качестве сдвига задавать величину

$$\bar{\lambda} = \lambda_m^* + \varepsilon, \quad m=0, 1, \dots, \quad (2.3.7)$$

где λ_m^* - вычисленное собственное значение, $\varepsilon > 0$ - параметр регуляризации.

Отметим, что в данной процедуре большую роль играет задание итерационного параметра τ_k .

2.4. Дискретное представление

Решение краевых задач (2.2.3), (2.3.1) и (2.3.2) осуществляется с помощью метода конечных разностей на равномерной сетке ω_h . Эта задача аппроксимируется с помощью разностных формул с точностью аппроксимации $O(h^4)$. Как и в методе Нумерова^{8/}, при выводе разностных формул используется прием повышения порядка точности аппроксимации на решении уравнения.

Это дало возможность построить трехточечные разностные формулы с точностью аппроксимации $O(h^4)$ в узлах сетки с номерами $i=3, \dots, n-2$.

В приграничных ($i=2, n-1$) и граничных ($i=1, n$) узлах используются пятиточечные схемы того же порядка точности.

Формулы строятся так, чтобы коэффициенты уравнения (I) использовались только во внутренних узлах сетки ω_h .

Это важно в тех задачах, где коэффициенты уравнения в граничных точках имеют особенности.

В случае $p(x) \equiv 0$ выведенная разностная схема совпадает со схемой Нумерова.

Интегрирование в формулах (2.2.5), (2.2.9) и (2.3.3) выполняется с помощью квадратурной формулы Симпсона точности порядка h^4 .

Полученные дискретные краевые задачи решаются методом прогонки.

2.5. Алгоритмы вычисления τ_k

В итерационных процедурах, описанных в пунктах 2.2 и 2.3, важную роль играет итерационный параметр τ_k . Выбор этого параметра позволяет регулировать ход итерационного процесса. При этом изменение параметра τ_k связано с изменением невязки δ_k в ходе итераций. Ниже приводятся следующие алгоритмы выбора τ_k , в которых τ_0 - некоторое заданное значение, $0 \leq \tau_0 \leq 1$.

$$1. \quad \tau_k \equiv \tau_0. \quad (2.5.1)$$

Этот алгоритм при достаточно малом τ_0 обычно применяется при плохих начальных приближениях, с целью проверить возможность сходимости от этих приближений. Сходимость при этом очень медленная. При $\tau_0 \equiv 1$ получается классическая схема Ньютона.

$$2. \quad \tau_k = \begin{cases} \min(1, 2\tau_{k-1}), & \text{если } \delta_k < \delta_{k-1}, \\ \max(\tau_0, \tau_{k-1}/2), & \text{если } \delta_k \geq \delta_{k-1}, \end{cases} \quad (2.5.2)$$

где δ_k определяется по формуле (2.2.8). Этот алгоритм аналогичен широко распространенному способу выбора шага интегрирования в стандартных программах решения задачи Коши, вычисления интегралов и т.п.^{9/}

Алгоритм рекомендуется применять при хороших начальных приближениях. Он обеспечивает быструю сходимость, однако не всегда устойчив.

$$3. \quad \tau_k = \begin{cases} \min(1, \tau_{k-1} \cdot \frac{\delta_{k-1}}{\delta_k}), & \text{если } \delta_k < \delta_{k-1}, \\ \max(\tau_0, \tau_{k-1} \cdot \frac{\delta_{k-1}}{\delta_k}), & \text{если } \delta_k \geq \delta_{k-1}. \end{cases} \quad (2.5.3)$$

Здесь δ_k тоже вычисляется по формуле (2.2.8). Этот алгоритм более устойчив и обеспечивает сходимость в достаточно широкой области начальных приближений.

$$4. \quad \tau_k = \frac{\delta_{k-1}}{\delta_{k-1} + \delta_k(1)}, \quad (2.5.4)$$

где $\delta_k(1)$ - невязка на k -той итерации для τ_{k-1} . Величина δ_k вычисляется по формуле (2.2.9). Это алгоритм оптимального выбора τ_k , предложенный в работе^{10/}. Он основан на квадратичной аппроксимации зависимости δ от τ . При итерациях он должен обеспечить минимум невязки на каждом шаге.

5. На равномерной сетке ω_τ отрезка $[0, 1]$ с шагом $\Delta\tau$ вычисляется последовательность невязок δ^i по формуле (2.2.8) и выбирается

такое значение \bar{U}_k , которому соответствует минимальная невязка. Этот алгоритм более общий, чем в п.4, хотя и требует большего объема вычислений. Точность нахождения оптимального шага \bar{U}_k , обеспечивающего минимум невязки на каждом шаге, зависит от выбора сетки ω_{τ} . Эту сетку можно выбрать таким образом, чтобы точность нахождения τ и быстродействие алгоритма оптимально сочетались. Алгоритмы 1,2,3 были реализованы в SLIP1 и прошли проверку на широком круге задач. Алгоритмы 4,5 использовались при решении задач /6/.

2.6. Точность вычислительной схемы

Пусть $z_n^* = \{ \lambda_n^*, y_n^*(x) \}$ - решение задачи (I), существование которого предполагается.

В результате вычислений по изложенной схеме за конечное число итераций k , т.е. при выполнении условия $\delta_k < \varepsilon$ (δ_k - невязка (2.2.8) в сеточной C - норме), получается приближенное решение $z_{nh}^k = \{ \lambda_{nh}^k, y_{nh}^k(x) \}$.

Пусть z_{nh} - точное разностное решение. Тогда в сеточной C -норме можно написать следующую оценку:

$$\| z_n^* - z_{nh}^k \| \leq \| z_n^* - z_{nh} \| + \| z_{nh} - z_{nh}^k \| . \quad (2.6.1)$$

Для первого слагаемого в правой части (2.6.1) имеется следующая оценка /11/:

$$\| z_n^* - z_{nh} \| \leq C_n h^4 . \quad (2.6.2)$$

Второе слагаемое оценивается как /12/

$$\| z_{nh} - z_{nh}^k \| \leq B_n \delta_k . \quad (2.6.3)$$

Здесь $B_n > 0$, $C_n > 0$ - константы.

Когда $B_n \delta_k \ll C_n h^4$, что выполняется при задании малого ε , точность полученного разностного решения близка к теоретической оценке

$$\| z_n^* - z_{nh}^k \| \approx C_n h^4 . \quad (2.6.4)$$

Ее можно детально исследовать путем расчетов на последовательности сгущающихся сеток. При этом возможно уточнение разностного решения /11/.

В случае, если исходная задача является сингулярной, в оценку (2.6.1) добавляется слагаемое, характеризующее ошибку аппроксимации сингулярных граничных условий условиями (Iв,с) и (2.2.1).

Как правило, оценить эту погрешность можно путем проведения последовательных расчетов на расширяющихся интервалах $[a, b]$.

Все эти факторы следует иметь в виду при задании параметров вычислительной схемы и при оценке полученных результатов.

3. Описание программ комплекса

Комплекс SLIP4 состоит из набора подпрограмм, которые можно объединить в три группы.

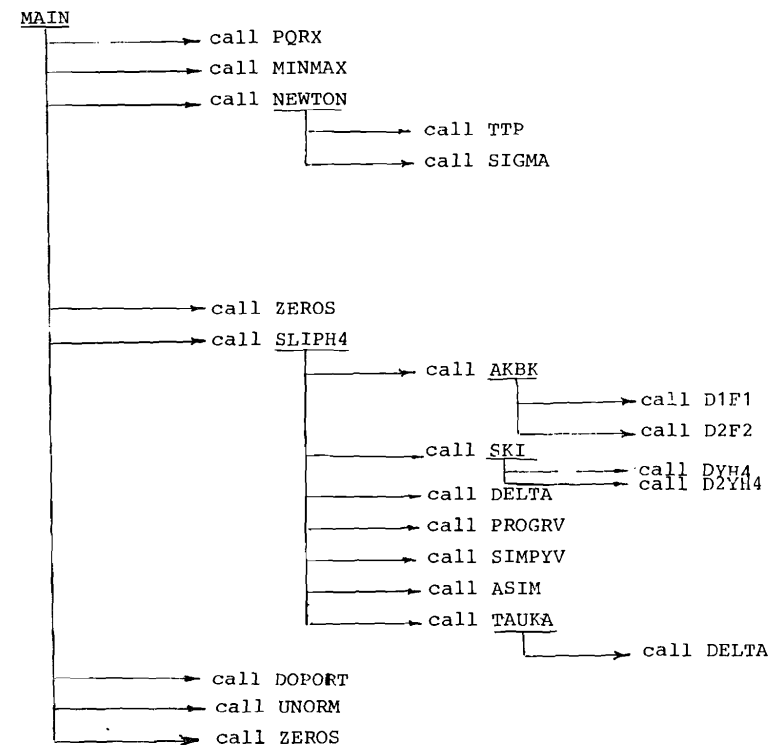
К первой группе относятся программы общего назначения. В них определяются величины, необходимые для работы остальных подпрограмм. Некоторые из них составляются пользователем комплекса.

Вторая группа программ обеспечивает нахождение начального приближения по алгоритму, изложенному в п.2.1.

В третью группу входят подпрограммы, осуществляющие уточнение начального приближения (см. п.2.2 и п.2.3).

Каждая из этих групп подпрограмм может быть использована самостоятельно.

На рисунке приведена схема комплекса программ SLIP4.



Главная программа MAIN и подпрограммы PQRX, D1F1, D2F2 и ASIM составляются пользователем комплекса с учетом особенностей конкретной задачи. Часть подпрограмм, таких как TTR, АКВК, SIGMA, DELTA, PROGRV, SIMPYV, SKI и ТАУКА, являются внутренними программами комплекса и пользователю недоступны. Ниже дается их краткое описание. Подробно описаны обращения и параметры подпрограмм, доступных пользователю.

3.1. Подпрограммы общего назначения PQRX, MINMAX, ZEROS, UNORM

Подпрограмма PQRX предназначена для заполнения массивов P, Q, R таблицами значений коэффициентов $p(x)$, $q(x)$, $r(x)$ уравнения (I) в узлах x_i сетки ω_h . Значения узлов x_i помещаются в массив X.

Обращение:

CALL PQRX(A, B, N, P, Q, R, X, XMIN, XMAX, XW, M),

где

A, B – левая и правая границы отрезка изменения x ,

N – количество узлов равномерной сетки ω_h ,

P, Q, R, X – массивы размерности N, которые должны быть описаны в вызывающей программе,

XMIN, XMAX – левая и правая границы интервала, на котором ищутся начальные приближения собственных значений,

XW и M – точка сшивки (2.1.1) и ее номер в массиве X(N).

В подпрограмме используется ключ LQ из общего блока

COMMON/SWITCH/LINFUN, LOUT, LQ, LEVZ.

Если пользователь вычисляет в подпрограмме значения XMIN, XMAX, XW и M сам, то следует присвоить LQ значение \emptyset ($LQ = \emptyset$).

Подпрограмма MINMAX по ключу $LQ \neq \emptyset$ из общего блока

| SWITCH | вычисляет значения $XMIN = \min Q(x)$ и $XMAX = \max Q(x)$

для $x \in \omega_h$, определяет точку сшивки XW как $Q(XW) = XMAX$

и M – ее номер в массиве X. M должно удовлетворять условию

$$4 \leq M \leq N-3. \quad (3.1)$$

Обращение:

CALL MINMAX(N, X, Q, XMIN, XMAX, XW, M).

Если $LQ = \emptyset$, то проверяется условие (3.1). Выполнение его ведет к возврату в вызывающую программу. При невыполнении условия (3.1) величине M присваивается ближайшее из двух значений 4 или $N-3$, определяется новое значение $XW = X(M)$, печатается сообщение о старом и новом значениях точки сшивки и происходит выход из подпрограммы.

Подпрограмма UNORM нормирует функцию y . Вычисляется новое значение $y = y / \sqrt{\int_a^b y^2(x) dx}$

Обращение:

CALL UNORM(N, H, Y),

где

n – шаг равномерной сетки ω_h

Y – массив размерности N, описанный в главной программе.

Подпрограмма ZEROS определяет число изменений знака в таблице значений функции y . Оно засылается в NZ и отождествляется с числом нулей функции y .

Обращение:

CALL ZEROS(N, Y, NZ),

где

Y – массив размерности n, описанный в главной программе.

3.2. Подпрограммы вычисления начального приближения

NEWTON, TTR И SIGMA

Подпрограмма NEWTON определяет начальные приближения задачи (I) по алгоритму, изложенному в п.2.1.

Обращение:

CALL NEWTON(A, B, P, Q, R, N, XMIN, XMAX, XW, M, EV, X, Y, DX, EPS, ITMIN, ITMAX, LSTEP, MMAX, NIS, NNEC, NEVMAX),

где параметры A, B, P, Q, R, N, XMIN, XMAX, XW, M, X описаны в п.3.1;

параметры DX, EPS, ITMIN, ITMAX, LSTEP, NIS, NNEC, NEVMAX

задаются пользователем. Результат работы программы записывается в массивы Y и EV, где

EV – массив корней полинома (2.1.2) размерности NEVMAX;

Y – массив размерности N, содержащий начальные приближение к решению задачи (I) для последнего найденного корня;

EPS – значение малой величины ε_T из формулы (2.1.4);

ITMIN – число итераций процесса (2.1.3) без проверки условия (2.1.4);

ITMAX – максимально допустимое число итераций. Если точность после этого не достигнута, выдается сообщение, значение найденного приближения, его точность и значение полинома t (2.1.2);

LSTEP – ключ, задающий направление поиска корня на отрезке $[XMIN, XMAX]$. При $LSTEP = \emptyset$ поиск ведется от XMIN, при

LSTEP=1 – от XMAX;

DX – заданная величина. Поиск корней ведется на отрезке $[XMIN-DX, XMAX+DX]$.

MMAX - после завершения работы программы **NEWTON** этому значению присваивается величина, равная количеству найденных корней полинома **T** ,
NIS - количество уже найденных каким-то образом корней полинома (собственных значений). Их значения до обращения к этой подпрограмме должны быть присвоены первым **NIS** элементам массива **EV** ;
NNES - номер корня, до которого включительно надо найти все корни полинома, начиная с (**NIS+1**)-го, и определить функцию **Y** ;
NEVMAX - максимально допустимое число корней $NEVMAX \geq NNES$.

В подпрограмме используется ключ **LOUT** из общего блока /SWITCH/ . Если **LOUT = 0** ; то происходит проверка найденного значения корня нахождение его в интервале (**XMIN-DX** , **XMAX+DX**) , Если **LOUT ≠ 0** , проверки не происходит.

Общий блок:

COMMON/CMINMA/CMIN, SMAX ,

заполняемый пользователем в главной программе, задает диапазон чисел для конкретной ЭВМ (на CDC-6500 $CMIN = 10^{-100}$, $SMAX = 10^{+100}$) .

Подпрограмма **NEWTON** обращается к внутренней подпрограмме **TTR** и функции **SIGMA** .

Подпрограмма **TTR** вычисляет значение полинома **T(λ)** из (2.1.2), его производной **T'(λ)** и функции **Y** .

Функция **SIGMA** вычисляет

$$\sum_{j=1}^{n-1} \frac{1}{\lambda_n^k - \lambda_j}$$

из выражения (2.1.3).

3.3. Подпрограммы уточнения начального приближения **SLIPN4** , **AKBK**, **DELTA**, **PROGRV**, **SIMPYV**, **SKI**, **TAUKA**, **D1F1**, **D2F2**

Подпрограмма **SLIPN4** организует итерационный процесс, описанный в п.2.2 и п.2.3.

Обращение:

CALL SLIPN4(A,B,N,NMAX,P,Q,R,EV0,Y0,LAS,YS, EPS, NT, LST,
 LPR,LX, T0, X,LDEL, AN, BN, V, S) ,

где **A**, **B**, **N**, **P**, **Q**, **R**, **X** - описаны в п.3.1, параметры **NMAX**, **LAS**, **EPS**, **NT**, **LST**, **LPR**, **LX**, **T0**, **LDEL** - задаются пользователем.
NMAX - размерность массивов **P**, **Q**, **R**, **Y0**, **X**, **AN**, **BN**, **V**, **S** в главной программе, $NMAX \geq N$

EV0 и **Y0** - начальное приближение к собственному значению и собственной функции, задаваемое пользователем. Оно может быть получено способом, изложенным в п.3.1 с помощью подпрограммы **NEWTON** или каким-то другим образом. После завершения работы подпрограммы **SLIPN4** в **EV0** и **Y0** находится уточненное решение задачи (1),
LAS - параметр, определяющий наличие асимптотики для собственной функции на $[b, \infty)$. Если **LAS = 0** , задача считается определенной на конечном интервале $[a, b]$. При **LAS ≠ 0** происходит обращение к подпрограмме пользователя **ASIM** для вычисления интегралов, входящих в выражения (2.2.5) и (2.3.3).

YS - рабочий массив размерности **NMAX** в главной программе,

EPS - малое число ϵ (2.2.7),

NT - максимально допустимое количество итераций по **t** ,

LST - параметр, определяющий способ выбора шага τ_k

(см. п. 2.5). τ_k определяется:

при **LST=1** - по алгоритму 2.5.1,

при **LST=2** - по алгоритму 2.5.2,

при **LST=3** - по алгоритму 2.5.3,

при **LST=4** - по алгоритму 2.5.4,

при **LST=5** - по алгоритму 2.5.5,

LPR - параметр, определяющий шаг выдачи на печать результатов промежуточных итераций (2.2.6),

LX - шаг таблицы функции $y_k(x)$ при печати

$h_{печ} = h * LX$,

T0 - заданное начальное значение τ_0 . Рекомендуемые значения $\tau_0 = 0,01; 0,05; 0,1$.

LDEL - параметр, задающий способ вычисления невязки δ_k :

при **LDEL=1** - по формуле (2.2.8)

при **LDEL=2** - по формуле (2.2.9)

AN, **BN**, **V** - массивы размерности **NMAX** для прогоночных коэффициентов и значений функции y_{ki} (2.2.3).

В подпрограмме используются общие блоки:

/CMINMA/ , в котором в главной программе пользователем определен диапазон чисел на данной ЭВМ.

| MOD | MOD, AK, BK ,

в котором значение **MOD ≠ 0** определяет реализацию итерационного процесса п.2.2, а значение **MOD=2** - процесса, описанного в п.2.3.

AK, BK - значения правых частей в граничных условиях (2.2.4) или (2.3.2).

В SLIPN4 происходит обращение к АКВК, DELTA, PROGRV, SKI, SIMPYV, ТАУКА. Обращение к ASIM происходит при LAS ≠ 0.

Подпрограмма ASIM составляется пользователем, если это необходимо. В ней должны быть вычислены значения интегралов на полубесконечном интервале $[b, \infty)$, входящие в выражения (2.2.5) и (2.3.3).

Обращение:

CALL ASIM(A, B, EV, Y(N), S1, S2),

где A, B - границы отрезка $[a, b]$,
 EV - собственное значение λ_k ,
 Y(N) - значение собственной функции $y_k(b)$,
 S1 - значение интеграла $\int_b^\infty \Psi^2(\lambda_k, x) dx$,
 S2 - значение интеграла $\int_b^\infty \Psi(\lambda_k, x) \Psi'_\lambda(\lambda_k, x) dx$.

Подпрограмма АКВК вычисляет величины a_{ik} и b_{ik} по формулам (2.2.4) или (2.3.2). Она, в свою очередь, обращается к подпрограммам пользователя D1F1 и D2F2, которые вычисляют значения $d_1, f_1, d'_{1\lambda}, f'_{1\lambda}, d_2, f_2, d'_{2\lambda}, f'_{2\lambda}$ в выражениях (2.2.4) и (2.3.2).

Обращения:

CALL D1F1(EV, A, D1, DD1, F1, DF1),

где EV - заданное значение λ_k ,
 A - левая граница отрезка $[a, b]$,
 D1 - $d_1(\lambda_k, a)$,
 DD1 - $d'_{1\lambda}(\lambda_k, a)$,
 F1 - $f_1(\lambda_k, a)$,
 DF1 - $f'_{1\lambda}(\lambda_k, a)$,

CALL D2F2(EV, B, D2, DD2, F2, DF2),

где EV - заданное значение λ_k ,
 B - правая граница отрезка $[a, b]$,
 D2 - $d_2(\lambda_k, b)$,
 DD2 - $d'_{2\lambda}(\lambda_k, b)$,
 F2 - $f_2(\lambda_k, b)$,
 DF2 - $f'_{2\lambda}(\lambda_k, b)$,

Подпрограмма DELTA вычисляет значение невязки δ_k по формулам (2.2.8) или (2.2.9). В PROGRV методом прогонки определяются функции $v_k^{(i)}$ в задачах (2.2.3), (2.2.4) и (2.3.1), (2.3.2).

В подпрограмме SIMPYV находятся значения интегралов $\int_a^b y_k^2(x) dx$ и $\int_a^b y_k(x) v_k^{(i)}(x) dx$, используемые в выражениях (2.2.5) и (2.3.3).

Подпрограмма ТАУКА реализует описанные в п.2.5 алгоритмы вычисления параметра τ_k .

Подпрограмма SKI вычисляет значения выражений $s_{2k} \cdot s_k^{(i)}$, $i=1, 2$ по формулам (2.2.4) и (2.3.2).

Подпрограмма DOPORT осуществляет ортогонализацию и нормировку по формулам (2.3.5) и (2.3.6).

4. Примеры использования комплекса SLIPN4

В качестве примера использования программы SLIPN4 было рассмотрено решение задачи (I) для случая потенциала Морзе /I/

$$p(x) \equiv 0, \quad q(x) = -2mD(e^{-2a_1(x-x_0)} - 2e^{-a_1(x-x_0)})$$

на отрезке $[a, b]$ с граничными условиями

$$d_1 = d_2 = 1; \quad f_1 = \sqrt{\lambda} - \sqrt{2mD} e^{ax_0}, \quad f_2 = \sqrt{\lambda} - \sqrt{2mD} e^{-a(b-x_0)}$$

где $a=0, b=20$, константы $a_1=0,67$; $x_0=2,15$; $m=4,69$;
 $D=0,1055$, шаг дискретной сетки $\omega_h, h=0,0125$ ($n=160$).

В таблице I приведены результаты работы комплексов программ SLIPN4, SLIP1 и аналитическое решение. LDEL=1 (невязка δ считается по формуле (2.2.8)), LST=3 (параметр τ_k определяется по формуле (2.5.3)), k - количество итераций.

Таблица I

	Аналит.	SLIPN4	SLIP1
λ	4,353116E-I	4,35311593E-I	4,35317356E-I
x	YANAL	$y^{(k)}, k=11$	$y^{(k)}, k=100$
0	1,88093417E-2	1,88093441E-2	1,880620E-2
2	5,14335066E-I	5,14335081E-I	5,143384E-I
4	4,61811049E-I	4,61811042E-I	4,618091E-I
6	1,69509499E-I	1,69509480E-I	1,695076E-I
8	4,92265882E-2	4,92265749E-2	4,922594E-2
10	1,34452673E-2	1,34452604E-2	1,344509E-2
12	3,61380423E-3	3,61380121E-3	3,613756E-3
14	9,67238461E-4	9,67237232E-4	9,672260E-4
16	2,58597485E-4	2,58597015E-4	2,585942E-4
18	6,91177870E-5	6,91176144E-5	6,911688E-5
20	1,84723677E-5	1,84723066E-5	1,847214E-5
		$\delta = 0,23 E-7$	$\delta = 9,35 E-II$

Таблица I наглядно демонстрирует преимущества комплекса SLIPN4 по сравнению со SLIP1 в смысле точности получаемого решения и скорости сходимости итерационного процесса.

В таблице 2 приведены результаты работы комплекса SLIPN4 на последовательности вдвое сгущающихся сеток ($h=0,05$). Видно, что сходимость разностного решения задачи (I) имеет порядок h^4 .

Таблица 2

	$\lambda_h = 0,435311628$	$\lambda_{h/2} = 0,435311571$	$\lambda_{h/4} = 0,435311568$	σ 16,3
x	y_h	$y_{h/2}$	$y_{h/4}$	σ
0	$0,188097822 \cdot 10^{-1}$	$0,188093685 \cdot 10^{-1}$	$0,188093434 \cdot 10^{-1}$	16,4
3	0,592725832	0,592725648	0,592725849	16,5
6	0,169509462	0,169509497	0,169509499	16,3
9	$0,258172752 \cdot 10^{-1}$	$0,258172844 \cdot 10^{-1}$	$0,258172849 \cdot 10^{-1}$	16,2
12	$0,361380251 \cdot 10^{-2}$	$0,361380418 \cdot 10^{-2}$	$0,361380429 \cdot 10^{-2}$	16,2
15	$0,500156669 \cdot 10^{-3}$	$0,500156968 \cdot 10^{-3}$	$0,500156987 \cdot 10^{-3}$	16,2
18	$0,691177382 \cdot 10^{-4}$	$0,691177892 \cdot 10^{-4}$	$0,691177924 \cdot 10^{-4}$	16,2
δ	$0,34 \cdot 10^{-9}$	$0,36 \cdot 10^{-9}$	$0,44 \cdot 10^{-9}$	

где
$$\sigma = \frac{x_h - x_{h/2}}{x_{h/2} - x_{h/4}}$$

В этом примере $p(x) \equiv 0$. Работа комплекса для случая $p(x) \neq 0$ демонстрируется на уравнении Лежандра

$$y'' - \frac{2x}{1-x^2} y' - \frac{\lambda}{1-x^2} y = 0$$

$-1 < x < 1$

с граничными условиями

$$y' - \frac{\lambda}{2} y \Big|_{x=-1} = 0$$

$$y' + \frac{\lambda}{2} y \Big|_{x=+1} = 0$$

В таблице 3 приведены результаты, полученные на последовательности вдвое сгущающихся сеток ($h = 0,04$) для решения с двумя нулями.

Дальнейшим развитием пакета является добавление к нему блока автоматического нахождения параметра N дискретной сетки ω_h (а также параметров A и B в случае сингулярной граничной задачи) в зависимости от требуемой точности результата.

Таблица 3

	$\lambda_h = -6,00000000$	$\lambda_{h/2} = -6,00000000$	$\lambda_{h/4} = -6,00000000$	I080
x	y_h	$y_{h/2}$	$y_{h/4}$	σ
-1,0	$0,158113581 \cdot 10$	$0,158113864 \cdot 10$	$0,158113882 \cdot 10$	16,0
-0,8	0,727322472	0,727323775	0,727323856	16,0
-0,6	$0,632454326 \cdot 10^{-1}$	$0,632455456 \cdot 10^{-1}$	$0,632455527 \cdot 10^{-1}$	16,0
-0,4	-0,411095310	-0,411096046	-0,411096093	16,0
-0,2	-0,695699755	-0,695701002	-0,695701080	16,0
0,0	-0,790567904	-0,790569320	-0,790569409	16,0
0,2	-0,695699755	-0,695701002	-0,695701080	16,0
0,4	-0,411095310	-0,411096046	-0,411096093	16,0
0,6	$0,632454326 \cdot 10^{-1}$	$0,632455456 \cdot 10^{-1}$	$0,632455527 \cdot 10^{-1}$	16,0
0,8	0,727322472	0,727323775	0,727323856	16,0
1,0	$0,158113581 \cdot 10$	$0,158113864 \cdot 10$	$0,158113882 \cdot 10$	16,0
δ	$0,28 \cdot 10^{-10}$	$0,33 \cdot 10^{-10}$	$0,12 \cdot 10^{-9}$	

Литература

1. Пузынин И.В., Пузынина Т.П. В сб.: Алгоритмы и программы для решения некоторых задач физики. КФКИ -74-34, Будапешт, 1974, с.93-111.
2. Марчук Г.И., Шайдуров В.В. Повышение точности решений разностных схем. "Наука", М., 1979.
3. Акишин П.Г., Пузынин И.В. ОИЯИ, 5-10992, Дубна, 1977.
4. Уилкинсон Дж.Х. Алгебраическая проблема собственных значений. "Наука", М., 1970.
5. Bailey P.B. SLEIGN An Eigenvalue-Eigenfunction Code for Sturm-Liouville Problems. SAND77-2044, Sandia Laboratories, 1978.

6. Faifman M.P. et al. Z. Phys. D, 1986, 2, p.79-85;
 Виноцкий С.И. и др. Акустический журнал, 1985, 31, с.787-790.
 Касчиев М.С. и др. ОИЯИ, РИИ-84-832, Дубна, 1984.
7. Калиткин Н.Н. Численные методы. "Наука", 1978.
8. Современные численные методы решения обыкновенных дифференциальных уравнений. Ред. Дж.Холл, Дж.Уатт "Мир", М., 1979.
9. Дымарский Я.С. и др. Справочник программиста, т.1, Судпром ГИЗ, Л., 1963.
10. Ермаков В.В., Калиткин Н.Н. ЖВМиМФ, 1981, 21, с.491.
11. Марчук Г.И., Шайдулов В.В. Повышение точности решений разностных схем. "Наука", М., 1979.
12. Гавурин М.К., Изв.вузов, Матем., 1958, 5(6), с.18.

НЕТ ЛИ ПРОБЕЛОВ В ВАШЕЙ БИБЛИОТЕКЕ?

Вы можете получить по почте перечисленные ниже книги, если они не были заказаны ранее.

D9-82-664	Труды совещания по коллективным методам ускорения. Дубна, 1982.	3 р. 30 к.
D3,4-82-704	Труды IV Международной школы по нейтронной физике. Дубна, 1982.	5 р. 00 к.
D11-83-511	Труды совещания по системам и методам аналитических вычислений на ЭВМ и их применению в теоретической физике. Дубна, 1982.	2 р. 50 к.
D7-83-644	Труды Международной школы-семинара по физике тяжелых ионов. Алушта, 1983.	6 р. 55 к.
D2,13-83-689	Труды рабочего совещания по проблемам излучения и детектирования гравитационных волн. Дубна, 1983.	2 р. 00 к.
D13-84-63	Труды XI Международного симпозиума по ядерной электронике. Братислава, Чехословакия, 1983.	4 р. 50 к.
D2-84-366	Труды 7 Международного совещания по проблемам квантовой теории поля. Алушта, 1984.	4 р. 30 к.
D1,2-84-599	Труды VII Международного семинара по проблемам физики высоких энергий. Дубна, 1984.	5 р. 50 к.
D17-84-850	Труды III Международного симпозиума по избранным проблемам статистической механики. Дубна, 1984. /2 тома/	7 р. 75 к.
D10,11-84-818	Труды V Международного совещания по проблемам математического моделирования, программированию и математическим методам решения физических задач. Дубна, 1983	3 р. 50 к.
D4-85-851	Труды IX Всесоюзного совещания по ускорителям заряженных частиц. Дубна, 1984 /2 тома/	13 р. 50 к.
D11-85-791	Труды Международной школы по структуре ядра, Алушта, 1985.	3 р. 75 к.
D13-85-793	Труды Международного совещания по аналитическим вычислениям на ЭВМ и их применению в теоретической физике. Дубна, 1985.	4 р.
D3,4,17-86-747	Труды X Международного симпозиума по ядерной электронике. Дубна 1985.	4 р. 80 к.
	Труды У Международной школы по нейтронной физике. Алушта, 1986.	4 р. 50 к.

Рукопись поступила в издательский отдел
 12 мая 1987 года.

Заказы на упомянутые книги могут быть направлены по адресу:
 101000 Москва, Главпочтамт, п/я 79
 Издательский отдел Объединенного института ядерных исследований

ТЕМАТИЧЕСКИЕ КАТЕГОРИИ ПУБЛИКАЦИЙ
ОБЪЕДИНЕННОГО ИНСТИТУТА ЯДЕРНЫХ
ИССЛЕДОВАНИЙ

Индекс	Тематика
1.	Экспериментальная физика высоких энергий
2.	Теоретическая физика высоких энергий
3.	Экспериментальная нейтронная физика
4.	Теоретическая физика низких энергий
5.	Математика
6.	Ядерная спектроскопия и радиохимия
7.	Физика тяжелых ионов
8.	Криогеника
9.	Ускорители
10.	Автоматизация обработки экспериментальных данных
11.	Вычислительная математика и техника
12.	Химия
13.	Техника физического эксперимента
14.	Исследования твердых тел и жидкостей ядерными методами
15.	Экспериментальная физика ядерных реакций при низких энергиях
16.	Дозиметрия и физика защиты
17.	Теория конденсированного состояния
18.	Использование результатов и методов фундаментальных физических исследований в смежных областях науки и техники
19.	Биофизика

Пузынин И.В., Пузынина Т.П., Стриж Т.А.
SLIPH4 - программа для численного решения задачи
Штурма-Лиувилля

P11-87-332

В работе дано описание программного комплекса SLIPH4, предназначенного для численного решения задачи Штурма-Лиувилля для линейного однородного дифференциального уравнения второго порядка с однородными граничными условиями. Используется непрерывный аналог метода Ньютона. Дискретное представление осуществляется разностными схемами четвертого порядка точности относительно шага дискретной сетки для независимой переменной. В комплекс включена программа вычисления начальных приближений к решениям с помощью модифицированного метода Ньютона для нахождения корней полинома с удалением вычисленных корней. В комплексе реализована возможность использования модификации метода с фиксацией собственного значения с дополнительной ортогонализацией найденной приближенной собственной функции. Это позволяет одновременно решить задачи нахождения начального приближения и его уточнения в рамках единой вычислительной схемы. Приведено описание программ и их параметров. Рассмотрены примеры использования комплекса.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1987

Перевод О.С.Виноградовой

Puzynin I.V., Puzynina T.P., Strizh T.A.
SLIPH4 - Program for Numerical Solution of the
Sturm-Liouville Problem

P11-87-332

SLIPH4 program complex for the numerical solution of the Sturm-Liouville problem for the second-order homogeneous linear differential equation with homogeneous boundary conditions is described. The continuous analogue of the Newton method is used. The discrete scheme that has been realized in the approach provides an accuracy of the order of four with respect to the mesh step of the independent variable. The program for the initial approximation to the solution calculations with the help of a modified Newton method for the polynomial roots finding with the calculated root elimination is added to the complex. The possibility of using the modified method with eigenvalue fixing and obtained approximate eigenfunction additional orthogonalization is realized in complex. This permits to solve both the initial approximation problem and its more precise definition by the same scheme. The program write-up is presented. Examples illustrating the program complex using are given.

The investigation has been performed at the Laboratory of Computing Techniques and Automation, JINR.

Communication of the Joint Institute for Nuclear Research. Dubna 1987