

SLIPH4M – программа на языке MAPLE для численного решения частичной проблемы Штурма-Лиувилля

Аннотация: Представлена программа, написанная на языке системы компьютерной алгебры MAPLE, предназначенная для вычисления спектра собственных значений и соответствующих собственных функций разностной задачи Штурма-Лиувилля. В ней реализованы итерационные схемы на основе непрерывного аналога метода Ньютона точности порядка $O(h^4)$, где h – шаг разностной сетки.

Ключевые слова: собственное значение, собственная функция, задача Штурма – Лиувилля, непрерывный аналог метода Ньютона.

SLIPH4M – a program for numerical solution of the Sturm-Liouville problem

Abstract: A program written in the language of the computer algebra system MAPLE is presented to calculate the spectrum of eigenvalues and corresponding eigenfunctions of Sturm-Liouville differential equations. In the program the iteration schemes are realized on the base of a continuous analog of Newton's method with accuracy $O(h^4)$, where h is the step size of the uniform grid.

Keywords: eigenvalue, eigenfunction, Sturm-Liouville problem, continuous analog of Newton's method.

Введение.

Задача Штурма-Лиувилля (Ш-Л) как на конечном, так и на бесконечном интервалах часто возникает при изучении математических моделей колебательных процессов, устойчивости систем, а также в квантовой механике. Особое место в постановке этой задачи занимают такие, в которых коэффициенты уравнения представлены в виде таблиц, полученных или из экспериментальных данных, или из некоторых предварительных расчетов. Для более точного использования этих числовых данных целесообразно применять дискретные аппроксимации исходных задач. Одной из таких аппроксимаций, для которой приближенная задача Ш-Л исследована полностью, является метод конечных разностей [1]. Включение пакета решения задачи Ш-Л в этой постановке в программную среду MAPLE позволяет, с одной стороны, восполнить его отсутствие в этой среде, а с другой стороны, обогатить пользовательский интерфейс пакета современными средствами компьютерной аналитики и графики.

Представляемый пакет программ SLIPH4M использует для решения разностной задачи Ш-Л итерационные схемы четвертого порядка точности относительно шага разностной сетки на основе непрерывного аналога метода Ньютона (НАМН) [2], эффективность которого проверена при решении ряда математических моделей из различных разделов физики, в том числе спектральных задач. В программе реализованы модификации ньютоновских итерационных схем, позволяющие решать указанную задачу с использованием априорной численной и аналитической информации, а возможности графической визуализации промежуточных и окончательных результатов делают более оперативным их анализ. Кроме того, вычисление с помощью пакета

SLIPН4М собственных значений дает возможность для некоторых аналитических потенциалов при дальнейшем обращении к процедуре dsolve в системе MAPLE получить собственные функции в аналитическом виде. В комплексе SLIPM [3] приведен подобный пример для уравнения Уиттекера.

На языке Фортран имеется аналогичный комплекс программ SLIPН4 [4].

1. Постановка задачи.

Для использования НАМН проблема Штурма-Лиувилля записывается в виде системы функциональных уравнений:

$$\varphi(z) = 0, \quad z = (\lambda, y), \quad \varphi = \{\varphi^{(j)}\}, \quad j = 1, 2, 3, 4 \quad (1.1)$$

для линейного дифференциального оператора второго порядка

$$\varphi^{(1)}(\lambda, y) \equiv y''(x) + 2p(x)y'(x) + (q(x) - \lambda r(x))y(x) = 0, \quad a \leq x \leq b, \quad (1.2)$$

краевых условий общего вида

$$\varphi^{(2)}(\lambda, y) \equiv d_1(\lambda, a)y'(a) + f_1(\lambda, a)y(a) = 0, \quad (1.3)$$

$$\varphi^{(3)}(\lambda, y) \equiv d_2(\lambda, b)y'(b) + f_2(\lambda, b)y(b) = 0, \quad (1.4)$$

и условия нормировки

$$\varphi^{(4)}(\lambda, y) \equiv \int_a^b y^2(x)dx + c^2 \int_b^\infty \psi^2(\lambda, x)dx - 1 = 0. \quad (1.5)$$

Здесь коэффициенты $p(x)$, $q(x)$, $r(x)$ - заданные функции, обеспечивающие существование нетривиальных собственных решений $y = y_n(x)$ с указанными граничными условиями, которым соответствуют собственные значения параметра $\lambda = \lambda_n$. Функции d_j, f_j ($j = 1, 2$) непрерывно дифференцируемы по λ и $d_j^2 + f_j^2 > 0$.

Граничные условия (1.3), (1.4) и условие нормировки (1.5) дают возможность учитывать асимптотики собственных функций тех задач, которые рассматриваются на полубесконечном интервале $[a, \infty)$. Здесь $\psi(\lambda, x)$ - асимптотическое выражение для искомой собственной функции $y(x)$ при $x \rightarrow \infty$, c - константа сшивки решения в некоторой достаточно удаленной точке $x = b$, $y(b) = c\psi(\lambda, b)$. Если задача рассматривается на конечном отрезке $[a, b]$, следует положить $c = 0$.

Пакет SLIPН4М позволяет решать следующие задачи для разностной аппроксимации задачи Ш-Л на равномерной сетке узлов $\omega_h = \{x_i = a + (i-1) * h; i = 1, \dots, N; N = (b-a) / (h-1)\}$ отрезка $a \leq x \leq b$.

а) Частичная задача, когда по заданному начальному приближению $\{\lambda_0, y_0(x)\}$ вычисляется ближайшее к нему решение $\{\lambda^*, y^*(x)\}$ задачи (1.2)-(1.5).

б) Вычисление спектра (или его части) разностной задачи, аппроксимирующего соответствующую часть спектра задачи (1.2)-(1.5).

Точность вычисления является величиной $O(h^4)$ и совпадает с точностью используемой разностной аппроксимации задачи (1.2)-(1.5).

Ниже дается краткое описание алгоритмов, используемых в программах комплекса SLIPН4М.

2. Метод решения - Непрерывный аналог метода Ньютона (НАМН).

2.1. Итерационные схемы на основе НАМН.

Задача (1.2)-(1.5) является нелинейным функциональным уравнением (1.1). Согласно НАМН [2], задача (1.1) заменяется эволюционным уравнением по непрерывному параметру t ($0 \leq t < \infty$)

$$\varphi'(z(t))z'(t) = -\varphi(z(t)) \quad (2.1)$$

с начальным условием

$$z(0) = z_0 = \{\lambda_0, y_0(x)\}. \quad (2.2)$$

Здесь φ' – производная Фреше φ ,

$$z'(t) = (\mu(t), v(x, t)), \quad \mu(t) = \lambda'(t), \quad v(x, t) = y'_t(x, t). \quad (2.3)$$

При достаточно общих предположения на $\varphi(z)$ имеем $\lim_{t \rightarrow \infty} \|z(t) - z^*\| = 0$, где $z^* = (\lambda^*, y^*(x))$ - решение задачи (1.2)-(1.5), ближайшее к z_0 . Для приближенного решения задачи (2.1), (2.2) методом Эйлера введем дискретное множество узлов $\{t_k\} = \{t_k = t_{k+1} - t_k, k = 0, 1, 2, \dots\}$.

Один шаг с номером k итерационного процесса заключается в следующем:

1. Решить краевую задачу для функции v_k

$$\begin{aligned} \varphi^{(1)}(\lambda_k, v_k) &= s_{2k}(x), \\ \varphi^{(2)}(\lambda_k, v_k) &= a_{2k}, \\ \varphi^{(3)}(\lambda_k, v_k) &= b_{2k}, \end{aligned} \quad (2.4)$$

$$s_{2k}(x) = r(x)y_k(x),$$

где

$$\begin{aligned} a_{2k} &= -[d'_{1\lambda} y'_k(a) + f'_{1\lambda} y(a)], \\ b_{2k} &= -[d'_{2\lambda} y'_k(b) + f'_{2\lambda} y(b)] \end{aligned} \quad (2.5)$$

при вычисленных на предыдущем шаге $(k-1)$ значениях $\lambda_k, y_k(x)$.

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

$$\mu_k = \frac{1 + \int_a^b y_k^2(x) dx - c^2 \int_b^\infty \psi^2(\lambda_k, x) dx}{2 \left(\int_a^b y_k(x) v_k(x) dx + c^2 \int_b^\infty \psi(\lambda_k, x) \psi'_\lambda(\lambda_k, x) dx \right)}. \quad (2.6)$$

3. Найти новые приближения λ_{k+1} и y_{k+1} , $k = 0, 1, 2, \dots$

$$\begin{aligned} \lambda_{k+1} &= \lambda_k + \tau_k \mu_k, \\ y_{k+1}(x) &= (1 - \tau_k) y_k(x) + \tau_k \mu_k v_k(x), \end{aligned} \quad (2.7)$$

где τ_k определяется специальными алгоритмами.

Итерации прекращаются при выполнении условия

$$\delta_k < \varepsilon, \quad (2.8)$$

где $\varepsilon > 0$ – заданное малое число, невязка δ_k для $x \in \omega_h$ определяется соответствующей разностной аппроксимацией для одной из формул:

$$\delta_k = \max_j \max_x |\varphi^{(j)}(\lambda_k, y_k(x))|, \quad j = 1, 2, 3, \quad (2.9)$$

или

$$\delta_k = \int_a^b [\varphi^{(1)}(\lambda_k, y_k(x))]^2 dx. \quad (2.10)$$

2.2. Модифицированный алгоритм.

Эта процедура является обобщением метода обратных итераций с фиксированным 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 $\bar{\lambda}$ собственное значение. На каждом шаге (с номером k) итерационного процесса для нахождения решения $\{\lambda_n^*, y_n^*(x)\}$ требуется выполнить следующее.

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

$$\begin{aligned} \varphi^{(1)}(\bar{\lambda}, v_k^{(i)}) &= s_k^{(i)}(x), \\ \varphi^{(2)}(\bar{\lambda}, v_k^{(i)}) &= a_{ik}, \\ \varphi^{(3)}(\bar{\lambda}, v_k^{(i)}) &= b_{ik}, \end{aligned} \quad (2.11)$$

$$\begin{aligned} s_k^{(1)}(x) &= -\varphi^{(1)}(\lambda_k, y_k), & s_k^{(2)}(x) &= r(x)y_k(x), \\ a_{1k} &= -\varphi^{(2)}(\lambda_k, y_k(a)), & a_{2k} &= -[d'_{1\lambda}(\bar{\lambda})y'_k(a) + f'_{1\lambda}(\bar{\lambda})y_k(a)], \\ b_{1k} &= -\varphi^{(3)}(\lambda_k, y_k(b)), & b_{2k} &= -[d'_{2\lambda}(\bar{\lambda})y'_k(b) + f'_{2\lambda}(\bar{\lambda})y_k(b)] \end{aligned} \quad (2.12)$$

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

$$\mu_k = \frac{1 - \int_a^b y_k^2(x) dx - c^2 \int_b^\infty \psi^2(\lambda_k, x) dx - 2 \int_a^b y_k(x) v_k^{(1)}(x) dx}{2 \left(\int_a^b y_k(x) v_k^{(2)}(x) dx + c^2 \int_b^\infty \psi(\bar{\lambda}, x) \psi'_\lambda(\bar{\lambda}, x) dx \right)}. \quad (2.13)$$

3. Найти следующие приближения λ_{k+1} и y_{k+1} , $k = 0, 1, 2, \dots$

$$\lambda_{k+1} = \lambda_k + \tau_k \mu_k, \quad y_{k+1}(x) = y_k(x) + \tau_k (v_k^{(1)}(x) + \mu_k v_k^{(2)}(x)). \quad (2.14)$$

4. Для самосопряженного оператора в (1.2) для приближения $y_{k+1}(x)$ к собственной функции $y_n(x)$ выполнить ортогонализацию приближения по отношению ко всем найденным собственным функциям $y_m^*(x)$, $m = 0, 1, 2, \dots, n-1$ по формуле

$$u_{k+1}(x) = y_{k+1}(x) - \sum_{m=0}^{n-1} y_m^*(x) \int_a^b y_{k+1} y_m^*(x) dx. \quad (2.15)$$

5. Нормировать приближенную собственную функцию

$$y_{k+1}(x) = u_{k+1}(x) \left[\int_a^b u_{k+1}^2(x) dx \right]^{-1/2}. \quad (2.16)$$

Операции (2.15), (2.16) позволяют дополнительно подавлять вычислительные погрешности.

Изменение сдвига $\bar{\lambda}$ может обеспечить сходимость 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 (1.2)-(1.5). Далее в качестве сдвига задавать величину

$$\bar{\lambda} = \lambda_m^* + \xi, \quad m = 0, 1, \dots, \quad (2.17)$$

где λ_m^* - вычисленное собственное значение, $\xi > 0$ - параметр сдвига.

2.3. Дискретное представление.

Решение краевых задач (2.4), (2.5) и (2.11), (2.12) осуществляется с помощью метода конечных разностей на равномерной сетке ω_h . Эта задача аппроксимируется с помощью разностных формул с точностью аппроксимации $O(h^4)$. При выводе разностных формул используется прием повышения порядка точности аппроксимации на решении уравнений. 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 (1.2)-(1.4) использовались только во внутренних узлах сетки ω_h . Это важно в тех задачах, где коэффициенты уравнения в граничных точках имеют особенности.

Интегрирование в формулах (2.6), (2.10) и (2.13) выполняется с помощью квадратурной формулы Симпсона точности порядка $O(h^4)$. Полученные дискретные краевые задачи решаются методом прогонки.

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

2.4. Алгоритм вычисления начальных приближений.

Алгоритм вычисления начальных приближений основан на решении двух «встречных» задач Коши. Для первой задачи Коши ($x \geq a$) ставятся начальные условия в точке $x = a$, где условие

(1.3) дополняется условием на производную, если $d_1 = 0$. Аналогичное условие для второй задачи ($x \leq b$) ставится в точке $x = b$. Задача об отыскании собственных значений сводится тогда к отысканию корней функции

$$T(\lambda) \equiv y'_n(x_m)_{left} y_n(x_m)_{right} - y_n(x_m)_{left} y'_n(x_m)_{right} = 0, \quad (2.18)$$

появляющейся из условия равенства логарифмических производных для решений двух задач Коши в некоторой внутренней точке отрезка $x_m \in [a, b]$

$$\frac{y'_n(x_m)_{left}}{y_n(x_m)_{left}} = \frac{y'_n(x_m)_{right}}{y_n(x_m)_{right}}. \quad (2.19)$$

Если задачи Коши решать приближенно с помощью трехточечной разностной схемы точности $O(h^2)$, где h - шаг равномерной сетки ω_h , а функции $d_j(\lambda), f_j(\lambda)$ - полиномы некоторой степени от $\eta = \sqrt{\lambda}$, что справедливо для широкого круга практических задач, то функция $T(\eta)$ является полиномом. Его корни η_n находятся по методу Ньютона с исключением уже найденных корней (k - номер итерации):

$$\eta_n^{k+1} = \eta_n^k - \frac{T(\eta_n^k)}{T'(\eta_n^k) - T(\eta_n^k) \sum_{j=1}^{n-1} \frac{1}{\eta_n^k - \eta_j}}. \quad (2.20)$$

Итерационный процесс (2.20) прекращается при одновременном выполнении двух условий:

$$|\eta_n^{k+1} - \eta_n^k| \leq \varepsilon_\eta \quad \text{и} \quad |T(\eta_n^{k+1})| \leq \varepsilon_T, \quad (2.21)$$

где $\varepsilon > 0, \varepsilon_T > 0$ - заданные малые числа. При определении η_n с заданной точностью мы одновременно получаем и приближенное значение соответствующей сеточной собственной функции $y_n(x), x \in \omega_h$.

2.5. Алгоритмы вычисления итерационного параметра τ_k .

Вычисление итерационного 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, $0 < \tau_0 \leq 1$.

$$1) \quad \tau_k \equiv \tau_0. \quad (2.22)$$

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 $\tau_k \equiv 1$ получается классическая схема Ньютона.

$$2) \quad \tau_k = \begin{cases} \min(1, 2\tau_{k-1}), & \text{если } \delta_k < \delta_{k-1}, \\ \max(1, \tau_{k-1}/2), & \text{если } \delta_k \geq \delta_{k-1}, \end{cases} \quad (2.23)$$

где δ_k определяется по формуле (2.9). Этот алгоритм аналогичен широко распространенному способу выбора шага интегрирования в стандартных программах решения задачи Коши, вычисления интегралов.

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

$$3) \tau_k = \begin{cases} \min(1, \tau_{k-1} \frac{\delta_{k-1}}{\delta_k}), & \text{если } \delta_k < \delta_{k-1}, \\ \max(\tau_0, \tau_{k-1} \frac{\delta_{k-1}}{\delta_k}), & \text{если } \delta_k \geq \delta_{k-1}, \end{cases} \quad (2.24)$$

где δ_k определяется по формуле (2.9). Этот алгоритм более устойчив и обеспечивает сходимость в достаточно широкой области начальных приближений, однако зависит от задания τ_0 .

$$4) \tau_k = \frac{\delta_{k-1}}{\delta_{k-1} + \delta_k(1)}, \quad (2.25)$$

где δ_k определяется по формуле (2.10), $\delta_k(1)$ - невязка на итерации с номером k для $\tau_k = 1$.

5. На равномерной сетке ω_τ на отрезке $[0,1]$ с шагом Δ_τ вычисляется такое значение τ_k , которому соответствует минимальная невязка. Этот алгоритм более общий, чем в п.4, хотя и требует большего объема вычислений.

Недостатком алгоритмов 1-3 является эмпирическое задание начального значения параметра τ_0 .

2.6. Алгоритмы вычисления значения параметра τ_0 .

В программе реализованы два новых алгоритма для вычисления τ_0 :

$$\tau_0 = \frac{\delta_0}{2\delta_1(1)}, \quad (2.26)$$

$$\tau_0 = \frac{1}{2} \left| 1 - \frac{\delta_1(1)}{\delta_0} \right|, \quad (2.27)$$

где невязка δ_k определяется по формуле (2.9), $\delta_1(1)$ - невязка на первой итерации ($k=1$) для $\tau=1$.

Ограничение: $0,1 \leq \tau_0 \leq 1$. Их применение позволяет сократить число итераций, что показано при решении ряда примеров, представленных в работе [3].

3. Численный пример.

В качестве примера использования программы SLIPH4M рассмотрена задача о вычислении колебательных уровней энергии и волновых функций молекулы водорода [5].

Задача о молекуле водорода и ее ионах является фундаментальной в молекулярной физике. В этой области был разработан ряд достаточно точных методов решения обратной спектральной задачи, то есть построения потенциальных энергетических функций по наборам спектроскопических данных, известных обычно с большой точностью. Имея эти потенциальные кривые, можно вычислить волновые функции молекулы и ее основные характеристики. Спектроскопические

данные об уровнях энергии молекулы водорода и вычисленную в соответствии с ними потенциальную энергетическую кривую содержит работа [5].

Рассмотрим решение с помощью обсуждаемого численного метода и пакета SLIPН4М прямой спектральной задачи для молекулы водорода, то есть вычисление уровней энергии и волновых функций по приведенной в работе [5] таблице потенциальной кривой. Наличие большого количества расчетов и измерений уровней энергии H_2 делает этот пример особенно интересным для тестирования пакета, а результаты расчетов можно сравнить со спектроскопическими данными работы [5].

Уровни энергии E и волновые функции $y(x)$ молекулы водорода находятся как собственные значения и собственные решения радиального уравнения Шредингера

$$\frac{d^2 y}{dx^2} + [2ME - \frac{L(L+1)}{x^2} - V(x)]y = 0 \quad (3.1)$$

на полуоси $0 \leq x < \infty$ с граничными условиями

$$y(0) = y(\infty) = 0, \quad (3.2)$$

где M - приведенная масса квантово-механической системы, L - орбитальный момент.

В цитируемой работе [5] содержится таблица III (стр.138), в которой приведены экспериментальные значения уровней энергии молекулы водорода при значениях $2M = 1836.109$, $L = 0$. При этом потенциал уравнения (3.1) имеет вид

$$V(x) = 2M[U(x) + 1], \quad (3.3)$$

где $U(x)$ - потенциальная кривая, восстановленная по экспериментальному спектру и заданная в виде таблицы с неравномерным шагом на интервале $0.4 \leq x \leq 10$ с 7 десятичными знаками после запятой. Для численного решения задачи (3.1)-(3.3) с помощью рассматриваемого в данной работе метода условия ограниченности (3.2) для волновой функции дискретного спектра ($E < 0$) аппроксимируются граничными условиями на отрезке $0 \leq x \leq 10$

$$y(0) = 0, \quad (3.4)$$

$$\left(\frac{d}{dx} + \sqrt{-2ME}\right)y(x)\Big|_{x=10} = 0. \quad (3.5)$$

Краевое условие (3.5) является приближением асимптотического поведения волновой функции

$$y(x) = e^{-\sqrt{-2ME}x} \text{ при } x \rightarrow \infty. \quad (3.6)$$

Рассматривается условие нормировки функции (1.5) при значении $a = 0$, $b = 10$, $c = 0$,

$$\int_0^{10} y(x)^2 dx - 1 = 0. \quad (3.7)$$

Поскольку в таблице III работы [5] значения E_v колебательных уровней энергии H_2 в единицах (eV) приведены с 4 десятичными знаками после запятой, которые будем считать «точными» при тестировании, нами первоначально выбрана на отрезке $0 \leq x \leq 10$ равномерная сетка узлов ω_h с шагом $h = 0.005$. Для приближенного задания функции $U(x)$ в узлах этой сетки

на отрезке $0.4 \leq x \leq 10$ использовались как квадратичная интерполяция, так и сплайн-аппроксимация второго порядка исходных табличных данных. Оба способа дали одинаковые результаты в пределах точности разностной аппроксимации. При $x < 0.4$ функция $U(x)$ экстраполировалась с помощью выражения

$$u(x) = \frac{1}{x} + \rho, \quad (3.8)$$

соответствующего кулоновскому взаимодействию вблизи нуля. Константа ρ выбиралась из условия $U(0.4) = u(0.4)$.

При использовании пакета SLIP4M в задаче (1.2)-(1.4) выполнена замена $\lambda = E$, $p(x) = 0$, $q(x) = -V(x)$, $r(x) = 2M / 27.2107$, $a = 0$, $b = 10$. Вид функции $\Gamma(x)$ позволил осуществить перевод дискретных значений E в значения $E_v(UT) = 2ME$ единиц задачи (units of test (UT)) и получить эти значения в электрон-вольтах по формуле

$$E_v(eV) = -E_v(UT) * 27.2107.$$

Кроме того, проведен пересчет вычисленных значений $E_v(eV)$ к началу отсчета $E_0(eV)$, поскольку это же начало отсчета принято для табличных значений работы [5]. Расчеты выполнены для двух значений приведенной массы $2M$: из работы [5] и уточненной в соответствии с новыми данными.

Результаты расчетов и их сравнение с данными работы [5] представлены в таблице 1. Сравнение показывает, что абсолютные значения разностей между ними

$$\Delta_v(eV) = \left| E_v^{Table} - (E_0(eV) - E_v(eV)) \right| < 6.7643 * E_v^{Table} * 10^{-4},$$

где значения E_v^{Table} считаются «точными».

Разумеется, что на точность решения этой задачи оказывают влияние ошибки, связанные с интерполяцией, а в еще большей мере, с экстраполяцией табличной функции $U(x)$ с помощью формулы (3.8), поскольку ее точное асимптотическое поведение вблизи нуля неизвестно.

Помимо вычисленных значений $E_v(eV)$, соответствующих экспериментальным данным, в таблице 1 приведено собственное значение $E_{14}(eV)$ задачи (3.1) - (3.5), вычисленное с помощью рассматриваемого алгоритма (волновая функция $y_{14}(x)$ и потенциал $V(x)$ изображены на рис. 1), для которого соответствующего экспериментального уровня энергии нет. Следовательно, вычисленная часть спектра задачи (3.1) - (3.5) перекрывает экспериментальный спектр, то есть пакет SLIP4M может быть успешно применен в задачах, аналогичных рассмотренной.

Для $\nu = 14$ точность решения $O(h^p)$ ($p = 4$) реализованной разностной схемы подтверждена вычислениями на последовательности трех сгущающихся сеток с шагами $h = 0.005(N = 2001)$, $h/2$, $h/4$ и получением значения отношения Рунге для $z_h = (\lambda_h, y_h)$

$$\sigma = \frac{z_h - z_{h/2}}{z_{h/2} - z_{h/4}} \approx 27.5 - 45.0, \quad (3.9)$$

то есть порядок $p = \log_2 \sigma > 4$.

Таблица 1.

ν	Работа[5]	SLIPH4M (2M=1836.109)		SLIPH4M (2M=1836.15267247)	
	$E_\nu^{Table}(eV)$	$E_\nu(eV)$	$(E_0 - E_\nu)(eV)$	$E_\nu(eV)$	$(E_0 - E_\nu)(eV)$
0	0.0	4.477241	0.0	4.477244	0.0
1	0.5159	3.961056	0.516185	3.961065	0.516179
2	1.0025	3.474067	1.003174	3.474081	1.003163
3	1.4606	3.015652	1.461588	3.015671	1.461573
4	1.8906	2.585501	1.891739	2.585524	1.891720
5	2.2925	2.183467	2.293774	2.183492	2.293751
6	2.6662	1.809706	2.667535	1.809733	2.667510
7	3.0112	1.464706	3.012535	1.464735	3.012509
8	3.3266	1.149392	3.327849	1.149422	3.327822
9	3.6109	0.865141	3.612100	0.865171	3.612073
10	3.8622	0.613996	3.863244	0.614026	3.863218
11	4.0774	0.398809	4.078432	0.398835	4.078408
12	4.2530	0.223542	4.253699	0.223565	4.253679
13	4.3831	0.093728	4.383513	0.093744	4.383499
14		0.017056	4.460185	0.017064	4.460180

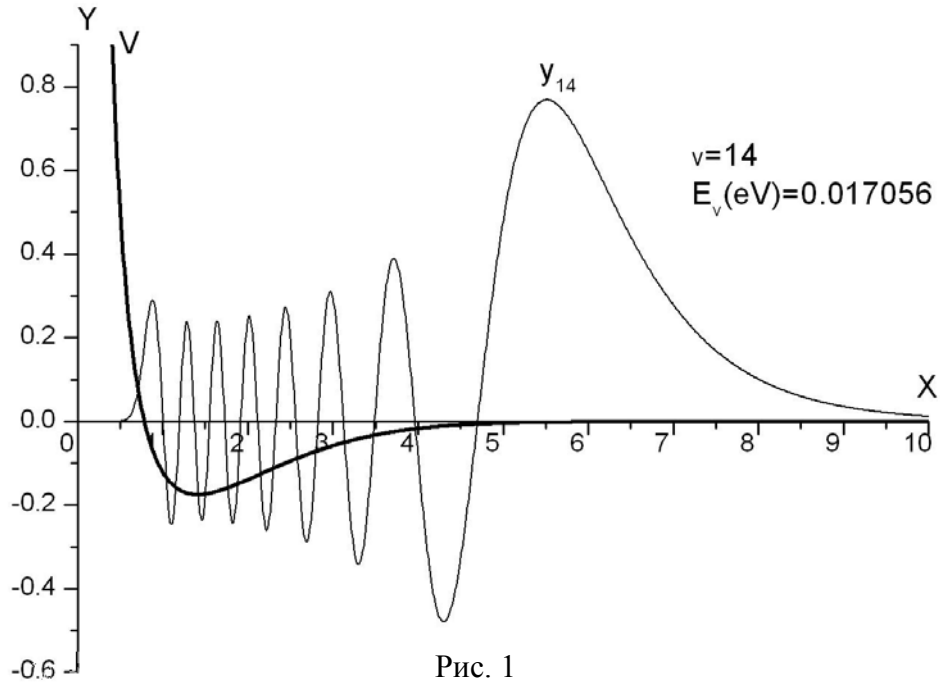


Рис. 1

Расчеты выполнялись на PC (Intel(R) Pentium(R) M processor 1.8GHz) в системе Maple версий 13-15. Переменное окружение Digits управляет числом цифр, которые Maple использует при вычислениях с числами с плавающей запятой. По умолчанию, при Digits=10, алгоритм в пакете SLIPH4M не сходится, поэтому в пакете SLIPH4M Digits=20. Начальное приближение в п.2.4 дает невязку по формуле (2.9) $\delta_k \approx 10^{+2} - 10^{-2}$. Итерации осуществлялись с выбором параметра τ_k согласно формуле (2.24) ($\tau_0 = 0.1$), причем в процессе уточнения, в среднем за 11-12 итераций, невязка достигала величины $\approx 10^{-8} - 10^{-11}$.

4. Описание процедур комплекса и их параметров.

В начале главной программы MAIN помещается ряд необходимых команд:

restart: - перезагрузить все переменные и процедуры,

with(CurveFitting):Support curve-fitting

with(linalg):Load linalg package containing all kinds of linear algebra routines

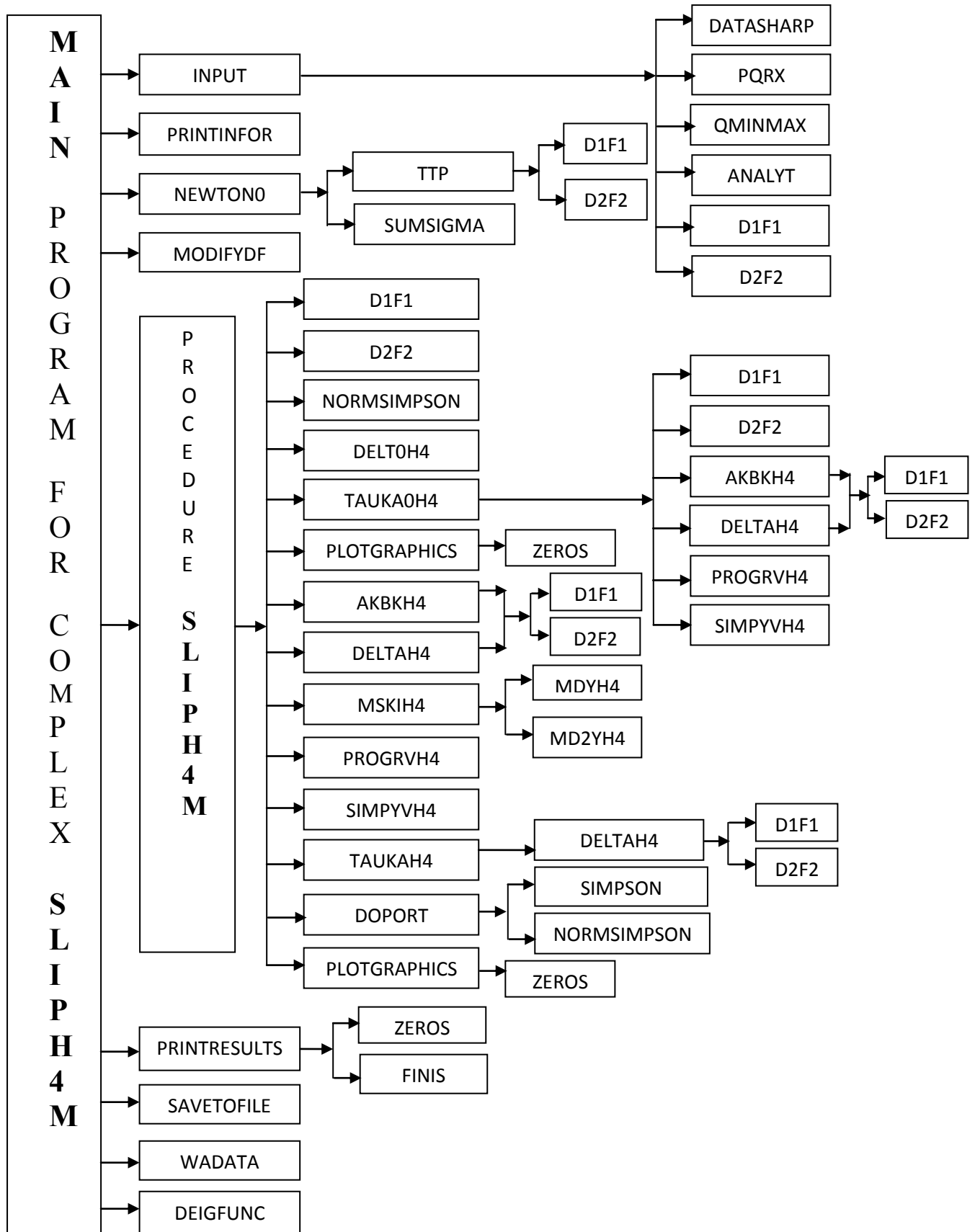
with(plots):Load graphic package

Digits:=20: Digits:=L: - переменная окружения, управляющая числом цифр L, которые Maple использует при вычислениях с числами с плавающей запятой. По умолчанию Digits=10.

ts:=time():определяет реальное время начала вычислений.

На Рис.2 дана блок-схема комплекса программ SLIPH4M для решения задачи, приведенной в качестве численного примера в пункте 3. Результаты работы комплекса записываются в файлы ALLDATA.txt, SLIPH4M.txt, VSHARPP138.txt.

Рис.2. Блок-схема программного комплекса SLIPH4M.



Далее следуют тела всех процедур комплекса и команды обращения к процедурам INPUT, PRINTINFOR, NEWTON0, MODIFYDF, SLIPH4M, PRINTINFOR, PRINTRESULTS, SAVETOFILE, WADATA, DEIGFUNC.

Команда tc:=time()-ts: процедуры PRINTRESULTS дает время выполнения всего вычислительного процесса.

Процедура INPUT и используемые в ней процедуры DATASHARP, PQRX, QMINMAX, ANALYT, D1F1, D2F2 составляются пользователем комплекса с учетом постановки конкретной задачи.

В качестве примера использования программы SLIPH4M рассмотрена задача о вычислении колебательных уровней энергии и волновых функций молекулы водорода [5].

Процедуры NEWTON0, MODIFYDF, SLIPH4M, WADATA, DEIGFUNC являются внутренними стандартными процедурами комплекса. Ниже дается их краткое описание, обращение к ним и к используемым в них процедурам второго-четвертого рядов TTP, SUMSIGMA, NORMSIMPSON, DELT0H4, TAUKA0H4, PLOTGRAPHICS, АКВКН4, DELTAN4, MSKIH4, PROGRVH4, SIMPYVH4, TAUKAN4, DOPORT, ZEROS, FINIS.

4.1. Процедуры INPUT, DATASHARP, PQRX, D1F1, D2F2, ANALYT.

Эти процедуры составляются пользователем в соответствии с поставленной проблемой.

Процедура INPUT. Обращение: INPUT(0).

Global A,B,N,H,P,Q,R,X,W,NZERO,EV0,Y0,EV,Y,EPS,EPS0,NIT,ITMAX,LST0,LST,LDEL, LPR,LX,EV0USER,T0USER,CMIN,CMAX,CEV0,CY0,SHIFTY0,LVISU,LMOD,EVANALYT, YANALYT,EV0NEWTON,Y0NEWTON,EV0ARRAY,OXLG,EQNAME,MODAL,MHD,NZSTART, NZEND,DATAEIGFUNC,EIGFUNC,WRADATA,COEFTR,VSHARP,UGRAPHIC,IGRAPHIC,

где

A, B - левая и правая границы отрезка изменения x.

N - количество узлов равномерной сетки. N должно быть нечетным числом. Если N задано четным числом, то в программе вычисляется новое $N=N+1$.

H - шаг разностной сетки, вычисляется по формуле $H=(B-A)/(N-1)$.

P, Q, R, X, W - массивы размерности N, которые должны быть вычислены в процедуре PQRX.

W - весовая функция W(x).

NZERO - число нулей функции Y.

EV0 и Y0 - начальное приближение к собственному значению и собственной функции.

После завершения работы процедуры SLIPH4M в EV и Y находится вычисленное сеточное решение задачи $\{\lambda, y(x)\}$.

EPS и EPS0 - малые числа, характеризующие точность по невязкам вычисления решения задачи в итерационном процессе и при вычислении начального приближения к решению в NEWTON0).

NIT и ITMAX - максимальное допустимое количество итераций в итерационном процессе и при вычислении начального приближения к решению в NEWTON0.

LST - параметр, определяющий способ вычисления параметра τ_k (п. 2.5):

при LST =1-4 - по алгоритмам (2.22)-(2.25) соответственно,
при LST =5 - по алгоритму 5 п.2.5.

LST0 - параметр, определяющий способ вычисления шага τ_0 (п. 2.6):
при LST0 =1-3 значение $0 < \tau_0 \leq 1$ задается пользователем,
при LST0 =4 - по алгоритму (2.25) при значении k=1,
при LST0 =5 - по алгоритму 5, п.2.5,
при LST0 =6 - по алгоритму (2.26),
при LST0 =7 - по алгоритму (2.27).

LDEL - параметр, задающий способ вычисления невязки δ :
при LDEL=1 - по формуле (2.9),
при LDEL=2 - по формуле (2.10).

LPR - параметр, определяющий шаг выдачи на монитор результатов промежуточных итераций.

LX - параметр, задающий шаг по переменной x в таблицах Y при выводе на монитор.

EV0USER - заданное пользователем начальное собственное значение.

TOUSER - заданное начальное значение τ_0 . Рекомендуемые значения $\tau_0 = 0.01; 0.05; 0.1$.

CMIN - минимальное положительное число с плавающей запятой, равное в тестах 10^{-30} .

CMAK - максимальное положительное число с плавающей запятой, равное в тестах 10^{30} .

LMOD - параметр, задающий при LMOD=1 модификацию начального решения перед обращением к процедуре SLIPH4M по формулам $EV0=EV0+CEV0$, $Y0=CY0*(Y0+SHIFTY0)$.

CEV0 - заданное значение константы. Рекомендуемые значения 1, 2, 3,..1000.

CY0 - заданное значение константы. Рекомендуемые значения 1, 2, 3.

SHIFTY0 - заданное значение константы. Рекомендуемые значения 1, 1.5, 2.

LVISU - параметр, задающий при LVISU=1 графическую визуализацию промежуточных и окончательных результатов итерационного процесса.

EVANALYT и YANALYT - аналитическое собственное значение и аналитическая собственная функция.

EV0NEWTON и Y0NEWTON - собственное значение и аналитическая собственная функция, вычисленные в процедуре NEWTON0.

EV0ARRAY – массив собственных значений.

OXLG – графическая ось OX, в тесте OXLG=A .

EQNAME – название проблемы. В тесте
EQNAME:="Schredinger equation for the Hydrogen Molecule\n":

MODAL- параметр, задающий используемый вычислительный метод.
При MODAL=0 – процесс НАМН, описанный в п.2.1.
При MODAL=1 - Модифицированный НАМН процесс, описанный в п.2.2.

При Modal=0 total CPU time(seconds) used for the current TEST: 44.351. При Modal=1 total CPU time(seconds) used for the current TEST: 210.056.

DATAEIGFUNC, EIGFUNC – параметры, используемые только для случая MODAL=1 и подробно описанные в комментариях в процедуре INPUT.

WRADATA=1 – параметр, дающий возможность записать результаты вычислений в файл "ALLDATA.txt".

MHD = 1836.109/2 – приведенная масса системы (3.1) – (3.5).

COEFTR – коэффициент перевода полученных собственных значений проблемы (3.1) - (3.5) из единиц вычислений в другие:

COEFTR=1 в обычном случае, без перевода.

COEFTR=2*MHD/27.2107 в приложенном тесте – перевод значений дискретного спектра в единицы $E_{\nu}^{table(E_{\nu})}$, используемые в работе [5].

NZSTART, NZEND - начальное и конечное число нулей вычисляемых собственных функций (спектр решений). Должно быть $NZSTART \leq NZEND$.

В тесте NZSTART=0(начальное NZERO), NZEND=14 (конечное NZERO).

VSHARP [i]=2*MHD*(USHARP[i]+1), где USHARP - потенциальная кривая $U(x)$ [5].

UGRAPHIC – потенциал энергии для вывода на график. UGRAPHIC[i]:=USHARP[i].

IGRAPHIC – параметр для вывода потенциала на график, X[IGRAPHIC]=0.4.

Значения параметров LST0,LST,LDEL,LPR,LX,LVISU,LMOD и констант A,B,N,NZERO,EPS,EPS0,NIT,ITMAX,EV0USER,T0USER,CEV0,CY0,SHIFTY0,CMIN,CMAX задаются пользователем.

В процедуре INPUT имеются обращения к процедурам DATASHARP, PQRX, ANALYT, D1F1, D2F2, QMINMAX.

Процедура DATASHARP предназначена для вычисления потенциала SHARP, который записывается в файл SHARPP138.txt.

Обращение: DATASHARP(MHD,N,X,H),
global A,B,VSHARP,UGRAPHIC,IGRAPHIC .

Процедура PQRX предназначена для заполнения массивов P, Q, R, X таблицами значений коэффициентов $p(x)$, $q(x)$, $r(x)$ уравнения (1.2) в узлах разностной сетки. Значения x_i помещаются в массив X.

Обращение: PQRX(0),
global A,B,N,H,P,Q,R,X,MHD,VSHARP,W,COEFTR.

Процедура ANALYT предназначена для вычисления аналитического решения {EVANALYT,YANALYT}.

Обращение: ANALYT(A,B,N,NZERO,X,YANALYT),
global EVANALYT,EV0.

В библиотечном тесте в процедуре ANALYT четырнадцать первым значениям EVANALYT[i], i=0-13, присваиваются известные “точные” значения из работы [5]. Пятнадцатому EVANALYT[14] присваивается значение, вычисленное с помощью пакета SLIP4M при N=2001.

Так как в данной проблеме аналитические значения собственных функций не известны, массивы YANALYT заполняются нулями.

Процедуры D1F1 и D2F2 вычисляют значения коэффициентов d_1, f_1, d_2, f_2 в граничных условиях (1.3),(1.4) в левом и правом концах отрезка [a,b] соответственно и их производных по собственному значению λ .

Обращение: D1F1(EV),
global A, D1, F1, DD1, DF1,

где EV - заданное собственное значение λ_k ,

D1, F1, DD1, DF1 - значения коэффициентов d_1, f_1

и их производных по собственному значению.

Обращение: D2F2(EV),
global B, D2, F2, DD2, DF2,

где EV - заданное собственное значение λ_k ,

D2, F2, DD2, DF2 - значения коэффициентов d_2, f_2

и их производных по собственному значению.

4.2 Процедуры вычисления начального приближения QMINMAX, NEWTON0, MODIFYDF.

Процедура QMINMAX вычисляет значения QMIN,QMAX,равные минимуму и максимуму потенциала Q[i], точку сшивки XW как Q(XW)=QMAX и M - ее номер в массиве X. M должно удовлетворять условию $4 \leq M \leq (N - 3)$.

Обращение: QMINMAX(N,X),
global QMIN,QMAX,M,XW,NZERO,Q.

Процедура NEWTON0 вычисляет начальное приближение к решению задачи (1.2) - (1.5) по алгоритму, изложенному в п.2.4, в интервале $QMIN \leq \lambda_0 \leq QMAX$.

Обращение: NEWTON0(A,B,N,NZERO,QMIN,QMAX,X,H,EPS0,ITMAX),
global CMAX, M, XW, SIGMA, T, TP, EVANALYT, EV0, EV0USER, EV0NEWTON, Y0,
Y0NEWTON, EV0ARRAY, EVID, Y0ARRAY, NZSTART, NZEND, COEFTR, DELLAM.

ITMAX - максимально допустимое число итераций. Если точность после этого не достигнута, выдается сообщение, значение найденного приближения, его точность и значение полинома T
SIGMA - значение суммы из выражения (2.20).

T - значение полинома $T(\lambda)$ из выражения (2.18) .

TP - значение производной $T'_\lambda(\lambda)$.

DELLAM – величина ξ в (2.17) [6].

Остальные величины описаны в процедуре INPUT.

Процедура MODIFYDF предназначена для модификации решения $\{EV0, Y0\}$ по формулам $EV0=EV0+CEV0$; $Y0=CY0(Y0 + SHIFTY0)$, где константы $CEV0$, $CY0$, $SHIFTY0$ заданы пользователем в процедуре INPUT.

Обращение: MODIFYDF(N,CEV0,CY0,SHIFTY0).

Процедура NEWTON0 обращается к внутренним процедурам TTP и SUMSIGMA.

Процедура TTP вычисляет значения полинома $T(\lambda)$ из выражения (2.18),

его производной $T'_\lambda(\lambda)$ и производной Y' функции Y .

Процедура SUMSIGMA вычисляет значение суммы из выражения (2.18).

Процедура для уточнения начального приближения SLIPH4M.

Процедура SLIPH4M организует итерационный процесс, описанный в п.п. 2.1,2.2.

Обращение: SLIPH4M(A,B,N,P,Q,R,W,AN,BN,V,EPS,NT,LST,LPR,LX,X,LDEL,S),
global CMIN,CMAX,KEPS,SY1,SV2,TK,TK1, AK, BK,YANALYT,EV0,Y0,YRS, EVRS, DELK1,
DELTA0,DELTA,MODAL,YNZR,EVAK,FNI,RET,DELTOAK,TAUAK, DELK,RDEL,DELT,AMJU,
T0,LST0,LVISU,KGRAPHIC,EVID, NZERO,EVANALYT,EV0ANALYT,NZSTART,NZEND,INZ,
где

AN, BN, V, AK, BK, YRS – массивы размерности N значений прогоночных коэффициентов, производной от Y в выражениях (2.4),(2.5),(2.11),(2.12), а также для прочих промежуточных вычислений. Двумерный массив $YNZR[INZ,i]$, $INZ=\{NZSTART,..,NZEND\}$, $i=\{1,..,N\}$ предназначен для накопления всех решений.

Из процедуры SLIPH4M происходит обращение к процедурам D1F1, D2F2, NORMSIMPSON, DELT0H4, TAUKA0H4, АКВКН4, DELTAN4, MSKIN4, PROGRVH4, SIMPYVH4, TAUKAN4, DOPORT,PLOTGRAPHICS, FINIS, ZEROS, TK.

Процедура NORMSIMPSON нормирует функцию Y с использованием метода Симпсона.

Процедура TAUKA0H4 вычисляет начальное значение параметра τ_0 , используя один из алгоритмов (2.26), (2.27) в соответствии со значением параметра LST0.

Процедура АКВКН4 вычисляет значения величин, входящих в граничные условия (1.3), (1.4).

Процедура MSKIN4 совместно с процедурами MDYH4, MD2YH4 вычисляет первую и вторую производную функции Y по схемам четвертого порядка точности.

Процедура PROGRVH4 вычисляет функцию $V(x)$ методом прогонки.(Sweep method)

Процедура SIMPYVH4 вычисляет интегралы S1, S2 от функций Y*Y и Y*V, используя метод Симпсона.

Процедура ТАУКАН4 вычисляет значение параметра τ_k , используя один из алгоритмов (2.22)-(2.25) в соответствии со значением параметра LST.

Процедуры DELTAN4 и DELT0H4 вычисляют значение невязки δ_k и δ_0 по формулам (2.9) или (2.10) в соответствии со значением параметра LDEL.

Процедура PLOTGRAPHICS предназначена для наглядного представления в виде графиков результатов итерационного процесса.

При значении параметра LVISU=1 на графики выдаются функции Y0, YANALYT, Y_k

где k - номер итерации, k=0, LPR, 2LPR, ...

Процедура ZEROS определяет число перемен знака в таблице значений функции Y. Оно засылается в NZERO и отождествляется с числом нулей функции Y.

Процедура PRINTINFOR выдает на дисплей данные об авторском коллективе и месте выполнения работы и краткое описание задачи.

Процедуры PRINTRESULTS и FINIS выдают на дисплей результаты промежуточных и конечных вычислений при выполнении итерационного процесса (TEST RUN OUTPUT).

Процедура SAVETOFILE осуществляет запись на файл SLIPH4M.txt информации, аналогичной двум предыдущим процедурам совместно.

Заключение.

Работа выполнена при финансовой поддержке РФФИ, гранты 09-01-00770-а и 10-01-00467-а.

Литература.

1. Самарский А.А. Теория разностных схем. М.: Наука, 1983. стр.258-276.
2. Пузынин И.В., Амирханов И.В., Земляная Е.В., Первушин В.Н., Пузынина Т.П., Стриж Т.А., Лахно В.Д. Обобщенный непрерывный аналог метода Ньютона для численного исследования некоторых квантово-полевых моделей. ЭЧАЯ, 1999, т.30, вып.1, стр.210-265. Phys.of Particles and Nuclei, Vol.30, No.1, 1999, p.87-110.
3. И.В.Пузынин, Т.П. Пузынина, Во Чонг Тхак. SLIPM - программа на языке MAPLE для численного решения частичной проблемы Штурма-Лиувилля на основе непрерывного аналога метода Ньютона. Вестник РУДН, серия "Математика. Информатика. Физика" No.2 вып.2, 2010, стр.90-98. Препринт ОИЯИ 1. Алгоритм. P11-2010-2, Дубна, 2010. Препринт ОИЯИ 2. Алгоритм. P11-2010-95, Дубна, 2010.
<http://wwwinfo.jinr.ru/programs/jinrlib/slipm/index.html>.
4. I.V. Puzynin, T.P. Puzynina, T.A. Strizh. SLIPH4 - Program for numerical solution of the Sturm-Liouville problem. JINR Report, P11-87-332, Dubna, 1987.<http://www.jinr.ru/programs/jinrlib/slip/>.
5. T.E. Sharp. Potential-energy curves for molecular hydrogen and its ions, Atomic data 2,119-169(1971).
6. Во Чонг Тхак, Пузынина Т.П. SLIPH4M – программа для численного решения проблемы Штурма-Лиувилля. Международный журнал "Программные продукты и системы". 2011.3(95).75-80. <http://wwwinfo.jinr.ru/programs/jinrlib/slip4m/>