

Выделим основные свойства разработанного фреймворка:

- использует архитектуру *MVC* (как и приложения, написанные с его применением);
- является шаблоном для разработки архитектуры приложения;
- предоставляет простые средства для выполнения действий над объектами, отображаемыми в таблицы БД (с поддержкой связанных таблиц): поиск, просмотр, добавление, изменение, удаление, работа со списками, экспорт и импорт, изменение свойств и методов объектов;
- содержит встроенные средства кэширования, создания логов и обработки ошибок;
- содержит гибкую систему контроля доступа (*ACL*), механизм авторизации;
- поддерживает темы оформления;
- содержит средства для управления меню.

Авторами предусмотрена возможность подстройки процесса расчета без необходимости изменять накопленные данные о проектах или перепрограммировать систему. К таким методам относятся использование шкал, изменение используемой метрики, добавление подготовки данных (*online*), такой как нормализация или масштабирование. Шкалы, введенные в формулу (1) как группа наборов значений w_j , позволяют изменять степень влияния конкретных признаков в соответствии с их значимостью для проекта. Поскольку проект имеет модульную структуру, изменить метрику можно с помощью реализации дополнительного модуля, не изменяя всю систему. Предварительная обработка данных может происходить с помощью дополнительного обработчика

(перед передачей их для вычисления расстояния близости).

При оценке инвестиционной привлекательности ИТ-проекта наиболее важным и одновременно наиболее сложным для расчета показателем является длительность его разработки. Это связано с тем, что она используется при расчете многих характеристик проекта в качестве основной переменной. Однако на этапе принятия решения о начале инвестирования отсутствует большое количество информации о характеристиках проекта, на основе которых обычно рассчитывают продолжительность разработки, а доступные для этого этапа методы (например экспертные оценки) дают низкую точность. Поэтому авторами разработан метод, позволяющий проводить оценку длительности разработки ИТ-проекта на ранних сроках. Предлагаемый метод основан на комбинировании принципов распознавания образов, на использовании статистических данных, опосредованных экспертных оценок и механизмов обучения и самообучения. Также данный метод имеет возможность адаптивной настройки для специфических условий. На основе этого метода разработано ПО, позволяющее проводить расчет длительности ИТ-проектов.

Литература

1. Сегаран Т. Программируем коллективный разум; [пер. с англ.]. СПб: Символ-Плюс, 2008. 368 с.
2. Чубукова И.А. Data Mining – Интернет-университет информационных технологий. ИНТУИТ.ру. М.: БИНОМ, 2008. 384 с.
3. Метод k-ближайших соседей – BaseGroup Labs. URL: http://www.basegroup.ru/glossary/definitions/nearest_neighbor/ (дата обращения: 02.06.2010).

УДК 517.9, 519.6, 681.3.06

SLIPH4M – ПРОГРАММА ДЛЯ ЧИСЛЕННОГО РЕШЕНИЯ ПРОБЛЕМЫ ШТУРМА–ЛИУВИЛЛЯ

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

Во Чонг Тхак; Т.П. Пузынина, д.ф.-м.н.

(Объединенный институт ядерных исследований, г. Дубна, puzynina@jinr.ru, votrongthach@jinr.ru)

Представлена программа, написанная на языке системы компьютерной алгебры MAPLE и 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.

При изучении математических моделей колебательных процессов, устойчивости систем, а также в квантовой механике часто возникает задача Штурма–Лиувилля (Ш–Л) на конечном и на

бесконечном интервалах. Особое место в ее постановке занимают такие задачи, в которых коэффициенты уравнения представлены в виде таблиц, полученных 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 – обогатить пользовательский интерфейс пакета современными средствами компьютерной аналитики и графики.

Представляемый пакет программ SLIPH4M для решения разностной задачи Ш–Л использует итерационные схемы 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 SLIPH4M собственных значений дает возможность при обращении к процедуре dsolve в системе MAPLE получить собственные функции в аналитическом виде. Созданный пакет программ является продолжением и развитием комплексов программ SLIPM [3] и SLIPH4 [4].

Рассмотрим постановку задачи. Для использования НАМН проблема Ш–Л записывается в виде системы функциональных уравнений

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

для линейного дифференциального оператора второго порядка:

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

краевых условий общего вида:

$$\varphi^{(2)}(\lambda, y) \equiv d_1(\lambda, a)y'(a) + f_1(\lambda, a)y(a) = 0, \quad (3)$$

$$\varphi^{(3)}(\lambda, y) \equiv d_2(\lambda, b)y'(b) + f_2(\lambda, b)y(b) = 0 \quad (4)$$

и условия нормировки:

$$\varphi^{(4)}(\lambda, y) \equiv \int_a^b y^2(x) dx + c^2 \int_b^\infty \psi^2(\lambda, x) dx - 1 = 0. \quad (5)$$

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 (3), (4) и условие нормировки (5) дают возможность учитывать асимпто-

тики 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 шивки решения в некоторой достаточно удаленной точке $x=b$, $y(b)=c\psi(\lambda, b)$. Если задача рассматривается на конечном отрезке $[a, b]$, следует принять $c=0$.

Пакет SLIPH4M позволяет решать следующие задачи для разностной аппроксимации задачи Ш–Л на равномерной сетке узлов ω_h отрезка $a \leq x \leq b$.

- Частичная задача, когда по заданному начальному приближению $\{\lambda^0, y^0(x)\}$ вычисляется ближайшее к нему решение $\{\lambda^*, y^*(x)\}$ задачи (2)–(5).

- Вычисление спектра (или его части) разностной задачи, аппроксимирующего соответствующую часть спектра задачи (2)–(5).

Точность вычисления является величиной $O(h^4)$ и совпадает с точностью используемой разностной аппроксимации задачи (2)–(5).

Методы решения задачи

Итерационные схемы на основе НАМН. Задача (2)–(5) является нелинейным функциональным уравнением (1). Согласно НАМН [2], задача (1) заменяется эволюционным уравнением по непрерывному параметру t ($0 \leq t < \infty$)

$$\varphi'(z(t))z'(t) = -\varphi(z(t)) \quad (6)$$

с начальным условием

$$z(0) = z_0 = \{\lambda_0, y_0(x)\}. \quad (7)$$

Здесь φ' – производная Фреше φ ,

$$z'(t) = (\mu(t), v(x, t)), \quad \mu(t) = \lambda'(t), \quad (8)$$

$$v(x, t) = y'_t(x, t).$$

При достаточно общих предположениях на $\varphi(z)$ имеем $\lim_{t \rightarrow \infty} \|z(t) - z^*\| = 0$, где $z^* = (\lambda^*, y^*(x))$ – решение задачи (2)–(5), ближайшее к z_0 . Для приближенного решения задачи (6), (7) методом Эйлера введем дискретное множество узлов $\{t_k\} = \{t_k = t_{k+1} - \tau_k\}$, $k=0, 1, 2, \dots$

Один шаг с номером k итерационного процесса заключается в следующем:

1. Решить краевую задачу для функции v_k

$$\begin{aligned} \varphi^{(1)}(\lambda_k, v_k) &= s_{2k}(x), \\ \varphi^{(2)}(\lambda_k, v_k) &= a_{2k}, \end{aligned} \quad (9)$$

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

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

$$\text{где } a_{2k} = -[d'_{1\lambda}y'_k(a) + f'_{1\lambda}y_k(a)], \quad (10)$$

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

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

$$\mu_k = \frac{1 + \int_a^b y_k^2(x) dx - c^2 \int_b^\infty \psi^2(\lambda_k, x) dx}{2 \int_a^b y_k(x) v_k(x) dx + c^2 \int_b^\infty \psi(\lambda_k, x) \psi'_\lambda(\lambda_k, x) dx} \quad (11)$$

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

$$y_{k+1}(x) = (1 - \tau_k) y_k(x) + \tau_k \mu_k v_k(x), \quad (12)$$

где τ_k определяется специальными алгоритмами.

Итерации прекращаются, если

$$\delta_k < \varepsilon, \quad (13)$$

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

$$\delta_k = \max_j \max_x |\varphi^{(j)}(\lambda_k, y_k(x))|, \quad j=1, 2, 3, \quad (14)$$

$$\delta_k = \int_a^b [\varphi^{(1)}(\lambda_k, y_k(x))]^2 dx, \quad x \in \omega_h. \quad (15)$$

Модифицированный алгоритм. Эта процедура является обобщением метода обратных итераций с фиксированным сдвигом $\bar{\lambda}$. При довольно грубых приближениях к собственной функции она обеспечивает 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 $v_k^{(2)}(x)$:

$$\begin{aligned} \varphi^{(1)}(\bar{\lambda}, v_k^{(i)}) &= s_k^{(i)}(x), \\ \varphi^{(2)}(\bar{\lambda}, v_k^{(i)}) &= a_{ik}, \\ \varphi^{(3)}(\bar{\lambda}, v_k^{(i)}) &= b_{ik}; \end{aligned} \quad (16)$$

$$\begin{aligned} s_k^{(1)}(x) &= -\varphi^{(1)}(\lambda_k, y_k), \\ a_{1k} &= -\varphi^{(2)}(\lambda_k, y_k(a)), \\ b_{1k} &= -\varphi^{(3)}(\lambda_k, y_k(b)), \\ s_k^{(2)}(x) &= r(x) y_k(x), \\ a_{2k} &= -[d'_{1\lambda}(\bar{\lambda}) y'_k(a) + f'_{1\lambda}(\bar{\lambda}) y_k(a)], \\ b_{2k} &= -[d'_{2\lambda}(\bar{\lambda}) y'_k(b) + f'_{2\lambda}(\bar{\lambda}) y_k(b)] \end{aligned} \quad (17)$$

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 $\mu_k =$

$$\mu_k = \frac{1 - \int_a^b y_k^2(x) dx - c^2 \int_b^\infty \psi^2(\lambda_k, x) dx - 2 \int_a^b y_k(x) v_k^{(1)}(x) dx}{2 \int_a^b y_k(x) v_k^{(2)}(x) dx + c^2 \int_b^\infty \psi(\bar{\lambda}, x) \psi'_\lambda(\bar{\lambda}, x) dx} \quad (18)$$

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

$$y_{k+1}(x) = y_k(x) + \tau_k (v_k^{(1)}(x) + \mu_k v_k^{(2)}(x)). \quad (19)$$

4. Для самосопряженного оператора в (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

$$u_{k+1}(x) = y_{k+1}(x) - \sum_{m=0}^{n-1} y_m^*(x) \int_a^b y_{k+1} y_m^*(x) dx. \quad (20)$$

5. Нормировать приближенную собственную функцию

$$y_{k+1}(x) = u_{k+1}(x) \left[\int_a^b u_{k+1}^2(x) dx \right]^{-1/2}. \quad (21)$$

Операции (20), (21) 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 $\bar{\lambda}$ следует задать соответствующую границу спектра задачи (2)–(5). Далее в качестве сдвига задавать величину

$$\bar{\lambda} = \lambda_m^* + \xi, \quad m=0, 1, 2, \dots, \quad (22)$$

где λ_m^* – вычисленное собственное значение; $\xi > 0$ – параметр сдвига.

Дискретное представление. Решение краевых задач (9), (10) и (16), (17) 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 уравнений (2)–(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 (11), (15) и (18) 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.

Алгоритм вычисления начальных приближений. Данный алгоритм основан на решении

двух встречных задач Коши. Для первой ($x \geq a$) ставятся начальные условия в точке $x=a$, где условие (3) дополняется условием на производную, если $d_1=0$. Аналогичное условие для второй задачи ($x \leq b$) ставится в точке $x=b$. Тогда задача отыскания собственных значений сводится к отысканию корней функции $T(\lambda) \equiv$

$$\equiv y'_n(x_m)_{left} y_n(x_m)_{right} - y_n(x_m)_{left} y'_n(x_m)_{right} = 0, \quad (23)$$

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

$$\frac{y'_n(x_m)_{left}}{y_n(x_m)_{left}} = \frac{y'_n(x_m)_{right}}{y_n(x_m)_{right}}. \quad (24)$$

Если задачи Коши решать приближенно с помощью трехточечной разностной схемы точности $O(h^2)$, где h – шаг равномерной сетки ω_n , 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):

$$\eta_n^{k+1} = \eta_n^k - \frac{T(\eta_n^k)}{T'(\eta_n^k) - T(\eta_n^k) \sum_{j=1}^{n-1} \frac{1}{\eta_n^k - \eta_j}}. \quad (25)$$

Итерационный процесс (25) прекращается при одновременном выполнении двух условий:

$$|\eta_n^{k+1} - \eta_n^k| \leq \epsilon_T \text{ и } |T(\eta_n^{k+1})| \leq \epsilon_T, \quad (26)$$

где $\epsilon_T > 0$ – заданное малое число. При определении η_n с заданной точностью одновременно получается и приближенное значение соответствующей сеточной собственной функции $y_n(x)$, $x \in \omega_n$.

Алгоритмы вычисления итерационного параметра τ_k . Вычисление итерационного параметра τ_k связано с изменением невязки δ_k в ходе итераций. Приведем алгоритмы выбора τ_k , в которых τ_0 – некоторое заданное значение $0 < \tau_0 \leq 1$.

$$1) \tau_k \equiv \tau_0. \quad (27)$$

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

$$2) \tau_k = \begin{cases} \min(1, 2\tau_{k-1}), & \text{если } \delta_k < \delta_{k-1}, \\ \max(1, \tau_{k-1}/2), & \text{если } \delta_k \geq \delta_{k-1}, \end{cases} \quad (28)$$

где δ_k определяется по формуле (14). 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.

$$3) \tau_k = \begin{cases} \min(1, \tau_{k-1} \frac{\delta_{k-1}}{\delta_k}), & \text{если } \delta_k < \delta_{k-1}, \\ \max(\tau_0, \tau_{k-1} \frac{\delta_{k-1}}{\delta_k}), & \text{если } \delta_k \geq \delta_{k-1}, \end{cases} \quad (29)$$

где δ_k определяется по формуле (14). Этот алгоритм более устойчив и обеспечивает сходимость в достаточно широкой области начальных приближений, однако зависит от задания τ_0 .

$$4) \tau_k = \frac{\delta_{k-1}}{\delta_{k-1} + \delta_k(1)}, \quad (30)$$

где δ_k определяется по формуле (15), $\delta_k(1)$ – 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 по формуле (30), хотя и требует большего объема вычислений. Недостатком алгоритмов, использующих формулы (27)–(29), является эмпирическое задание начального значения параметра τ_0 .

Алгоритмы вычисления значения параметра τ_0 . В программе реализованы два новых алгоритма для вычисления τ_0 :

$$\tau_0 = \frac{\delta_0}{2\delta_1(1)}, \quad (31)$$

$$\tau_0 = \frac{1}{2} \left| 1 - \frac{\delta_1(1)}{\delta_0} \right|, \quad (32)$$

где невязка δ_k определяется по формуле (14), $\delta_1(1)$ – 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 [3].

Численный пример

В качестве примера использования программы *SLIPH4M* рассмотрим задачу 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 разработаны достаточно точные методы 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 вычисленная в соответствии с ними потенциальная энергетическая кривая содержатся в работе [5].

Рассмотрим решение с помощью обсуждаемого численного метода и пакета *SLIPH4M* 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 [5].

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

$$\frac{d^2y}{dx^2} + [2ME - \frac{L(L+1)}{x^2} - V(x)]y = 0 \quad (33)$$

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

$$y(0) = y(\infty) = 0, \quad (34)$$

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

В работе [5] содержится таблица III, в которой приведены экспериментальные значения уровней энергии молекулы водорода при значениях $2M=1836,109$, $L=0$. При этом потенциал уравнения (33) имеет вид:

$$V(x) = 2M[U(x)+1], \quad (35)$$

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 (33)–(34) с помощью рассматриваемого в данной работе метода условия ограниченности (34) для волновой функции дискретного спектра аппроксимируются граничными условиями на отрезке $0 \leq x \leq 10$:

$$y(0) = 0, \quad (36)$$

$$\left(\frac{d}{dx} + \sqrt{-2ME}\right)y(x)\Big|_{x=10} = 0. \quad (37)$$

Краевое условие (37) является приближением асимптотического поведения волновой функции

$$y(x) = e^{-\sqrt{-2ME}x} \text{ при } x \rightarrow \infty. \quad (38)$$

Рассматривается условие нормировки функции (5) при значениях $a=0$, $b=10$, $c=0$,

$$\int_0^{10} y(x)^2 dx - 1 = 0. \quad (39)$$

Поскольку в таблице III работы [5] значения E_v колебательных уровней энергии H_2 в единицах (eV) приведены с четырьмя десятичными знаками после запятой, которые будем считать точными при тестировании, авторами первоначально выбрана равномерная сетка узлов ω_h с шагом $h=0,005$ на отрезке $0 \leq x \leq 10$. 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 точно-

сти разностной аппроксимации. При $x < 0,4$ функция $U(x)$ экстраполировалась с помощью выражения

$$u(x) = \frac{1}{x} + \rho, \quad (40)$$

соответствующего кулоновскому взаимодействию вблизи нуля. Константа ρ выбиралась из условия $U(0,4) = u(0,4)$.

При использовании пакета *SLIPH4M* в задаче (2)–(4) выполнена замена $\lambda = E$, $p(x) = 0$, $q(x) = -V(x)$, $r(x) = -2M$, $a = 0$, $b = 10$. Для перевода вычисленных значений E_v из единиц задачи (units of test (UT)) в электрон-вольты использовалась формула $E_v(eV) = -E_v(UT) * 27,2107$.

Кроме того, проведен пересчет вычисленных значений $E_v(eV)$ к началу отсчета $E_0(eV)$, поскольку это же начало 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. Сравнение показывает, что абсолютные значения разностей между ними таковы:

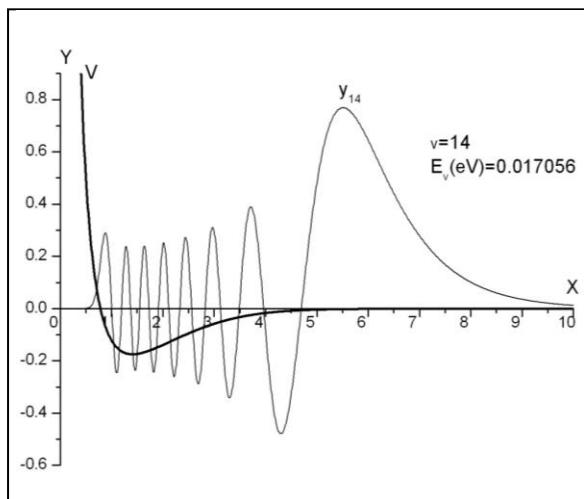
$$\Delta_v(eV) = |E_v^{Table} - (E_0 - E_v)| < 6,7643 * E_v^{Table} * 10^{-4},$$

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 (40), поскольку ее точное асимптотическое поведение вблизи нуля неизвестно.

v	E_v^{Table} (eV) из работ- ты [5]	SLIPH4M (2M=1836,109)		SLIPH4M (2M=1836,15267247)	
		E_v (eV)	$(E_0 - E_v)$ (eV)	E_v (eV)	$(E_0 - E_v)$ (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

Помимо вычисленных значений E_v , соответствующих экспериментальным данным, в таблице приведено собственное значение E_{14} задачи (33), (36), (37), вычисленное с помощью рассматриваемого алгоритма (волновая функция $y_{14}(x)$ и потенциал $V(x)$ изображены на рисунке), для которого



нет соответствующего экспериментального уровня энергии. Следовательно, вычисленная часть спектра задачи (33), (36), (37) перекрывает экспериментальный спектр, то есть пакет *SLIPH4M* может успешно применяться в задачах, аналогичных рассмотренной.

Для $v=14$ точность решения $O(h^p)$ ($p=4$) подтверждена вычислениями на последовательности трех сгущающихся сеток с шагами $h=0,005$ ($N=2001$), $h/2$, $h/4$ и получением значения отношения Рунге для $z_h=(\lambda_h, y_h)$:

$$\sigma = \frac{z_h - z_{h/2}}{z_{h/2} - z_{h/4}} \approx 27,5 - 45,0, \quad (41)$$

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

Расчеты выполнялись на *PC (Intel(R) Pentium(R) M processor 1.8GHz)* в системе *Maple* версии 13. Переменное окружение *Digits* управляет числом цифр, которые *Maple* использует при вычислениях с числами с плавающей запятой. При *Digits=10* алгоритм в пакете *SLIPH4M* по умолчанию не сходится, поэтому в пакете *SLIPH4M Digits=20*. Начальное приближение дает невязку по формуле (14) $\delta_k \approx 10^{+2} - 10^{-2}$. Итерации осуществлялись с выбором параметра τ_k согласно формуле (29) ($\tau_0=0,1$), причем в процессе уточнения невязка достигала величины $\approx 10^{-8} - 10^{-11}$ в среднем за 11 итераций.

Авторы благодарят профессора И.В. Пузынина (ОИЯИ, г. Дубна) за постоянный интерес, помощь и поддержку.

Литература

1. Самарский А.А. Теория разностных схем. М.: Наука, 1983. С. 258–276.
2. Пузынин И.В. [и др.]. Обобщенный непрерывный аналог метода Ньютона для численного исследования некоторых квантово-полевых моделей; Физика элементарных частиц и атомного ядра (ЭЧАЯ). 1999. Т. 30. Вып. 1. С. 210–265.
3. Пузынин И.В., Пузынина Т.П., Во Чонг Тхак. SLIPM – программа на языке MAPLE для численного решения частичной проблемы Штурма–Лиувилля на основе непрерывного аналога метода Ньютона // Вестн. РУДН: сер. Математика. Информатика. Физика. 2010. № 2. Вып. 2. С. 90–98.
4. Пузынин И.В., Пузынина Т.П., Стриж Т.А. SLIPH4 – программа для численного решения задачи Штурма–Лиувилля // Сообщения ОИЯИ, P11-87-332, Дубна, 1987.
5. Sharp T.E. Potential-energy curves for molecular hydrogen and its ions // Atomic Data and Nuclear Data Tables, 1971. Vol. 2, pp. 119–169.

УДК 378.146:519.87

МОДЕЛЬ ДИХОТОМИЧЕСКОЙ МАТРИЦЫ РЕЗУЛЬТАТОВ ТЕСТИРОВАНИЯ

И.Н. Елисеев, к.т.н.

(Южно-Российский государственный университет экономики и сервиса, г. Шахты,
ein@sssu.ru)

Рассматриваются теоретические предпосылки и основные этапы формирования модели дихотомической матрицы результатов тестирования. На основе вычислительного эксперимента сформирована и исследована модель дихотомической матрицы ответов размером 1009×49 , применение которой позволит успешно решать проблемы, связанные с моделированием и параметризацией педагогических тестов.

Ключевые слова: модель дихотомической матрицы, тест, индикатор теста, латентный параметр, модель Раша.

При исследовании широкого круга проблем, связанных с моделированием и параметризацией диагностических тестов [1], необходимо иметь модель дихотомической матрицы результатов тестирования, используя которую, можно было бы оценить генеральные значения β_j и θ_i латентных параметров однопараметрической дихотомической модели Раша [2, 3]. Элементы такой матрицы

должны соответствовать модели Раша, используемой в качестве модели измерения латентных параметров β и θ , а статистические параметры удовлетворять критериям качества виртуального теста-модели, результаты выполнения которого эти элементы матрицы представляют. Строки модели-матрицы – это совокупности нулей и единиц, оценивающие выполнение виртуального набора