

УДК 517.9, 519.6, 539.12

SLIPM — программа на языке MAPLE для численного решения частичной проблемы Штурма–Лиувилля на основе непрерывного аналога метода Ньютона

И. В. Пузынин, Т. П. Пузынина, В. Ч. Тхак

*Лаборатория информационных технологий
Объединённый институт ядерных исследований, Дубна, Россия
Laboratory of Information Technologies
Joint Institute for Nuclear Research
141980, Dubna, Russia*

SLIPM — (The Sturm-Liouville Problem in Maple) — программа на языке системы компьютерной алгебры (СКА) MAPLE, предназначенная для численного решения с помощью непрерывного аналога метода Ньютона (НАМН) частичной проблемы Штурма–Лиувилля, т.е. для вычисления некоторого собственного значения линейного дифференциального оператора второго порядка и соответствующей собственной функции, удовлетворяющей однородным граничным условиям. SLIPM является развитием написанных на языке Фортран программ SLIP1 и SLIPH4 и дополнена двумя новыми способами вычисления начального значения итерационного параметра τ_0 .

Ключевые слова: вычислительная математика, непрерывные аналоги итерационных методов, системы компьютерной алгебры, MAPLE.

1. Постановка задачи

Для линейного дифференциального уравнения

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

требуется вычислить собственное значение λ и соответствующее собственное решение $y(x)$, удовлетворяющее однородным краевым условиям

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

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

Для исключения тривиальных решений $y(x) \equiv 0$ вводится условие нормировки

$$\varphi^{(4)}(\lambda, y) \equiv \int_a^b y^2(x) dx = 1. \quad (4)$$

Здесь $p(x)$, $q(x)$, $r(x)$ — заданные функции, обеспечивающие существование нетривиальных собственных решений $y = y_n(x)$ граничной задачи (1)–(4), которым соответствуют собственные значения параметра $\lambda = \lambda_n$. Функции d_j , f_j непрерывно дифференцируемы по λ и $d_j^2 + f_j^2 > 0$; $j = 1, 2$.

Программа SLIPM по заданному начальному приближению $\{\lambda_n^0, y_n^0\}$ вычисляет частичное решение $\{\lambda_n^*, y_n^*\}$ задачи (1)–(4), где n — фиксировано, $x \in \omega_h$, ω_h — равномерная разностная сетка на отрезке $a \leq x \leq b$ с шагом h . Алгоритмы численного решения задачи, взятые нами за основу, реализованы в SLIP1 [1] и SLIPH4 [2] и хорошо зарекомендовали себя при решении многих сложных физических задач [3, 4].

Статья поступила в редакцию 28 ноября 2009 г.

Работа выполнена при финансовой поддержке РФФИ, грант 09-01-00770.

2. Метод решения (НАМН)

2.1. Алгоритм вычисления начального приближения

Алгоритм вычисления начального приближения основан на решении двух «встречных» задач Коши. Для первой задачи Коши ($x \geq a$) ставятся начальные условия в точке $x = a$, где условие (2) 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

$$T(\lambda) \equiv y'_n(x_m)_{\text{лев}} * y_n(x_m)_{\text{прав}} - y_n(x_m)_{\text{лев}} * y'_n(x_m)_{\text{прав}} = 0, \quad (5)$$

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

$$\frac{y'_n(x_m)_{\text{лев}}}{y_n(x_m)_{\text{лев}}} = \frac{y'_n(x_m)_{\text{прав}}}{y_n(x_m)_{\text{прав}}}. \quad (6)$$

Если задачи Коши решать приближённо с помощью трёхточечной разностной схемы точности $O(h^2)$, где h — шаг равномерной сетки ω_h , а функции $d_j(\lambda)$, $f_j(\lambda)$ являются полиномами от степени λ , что справедливо для широкого круга практических задач, то функция $T(\lambda)$ является полиномом. Его корни λ_n находятся по методу Ньютона с исключением уже найденных корней [5, 6] (k — номер итерации):

$$\lambda_n^{k+1} = \lambda_n^k - \frac{T(\lambda_n^k)}{T'_\lambda(\lambda_n^k) - T(\lambda_n^k) \sum_{j=1}^{n-1} \frac{1}{\lambda_n^k - \lambda_j}}. \quad (7)$$

Итерационный процесс (7) прекращается при одновременном выполнении двух условий:

$$|\lambda_n^{k+1} - \lambda_n^k| \leq \varepsilon_T \quad \text{и} \quad |T(\lambda_n^{k+1})| \leq \varepsilon_T, \quad (8)$$

где $\varepsilon_T > 0$ — заданное малое число. После определения λ_n с заданной точностью мы одновременно получаем и приближенное значение соответствующей сеточной собственной функции $y_n(x_i)$. В SLIPM этот алгоритм реализован в процедуре NEWTON0.

2.2. Итерационные схемы на основе НАМН

Задача (1)–(4) является нелинейным функциональным уравнением

$$\varphi(z) = 0, \quad z = (\lambda, y(x)),$$

где φ — совокупность функций $\varphi^{(j)}$, $j = 1, 2, 3, 4$. Согласно НАМН [3, 4], задача (1)–(4) заменяется эволюционным уравнением

$$\varphi'(z(t))z'(t) = -\varphi(z(t)), \quad 0 \leq t < \infty, \quad (9)$$

с начальным условием

$$z(0) = z_0 = (\lambda_0, y_0(x)). \quad (10)$$

Здесь φ' — производная Фреше оператора φ , $z'(t) = (\mu(t), v(x, t))$, где

$$\mu(t) = \lambda'(t), \quad v(x, t) = y'_t(x, t). \quad (11)$$

При достаточно общих предположениях на $\varphi(z)$ имеем $\lim_{t \rightarrow \infty} \|z(t) - z^*\| = 0$, где $z^* = (\lambda^*, y^*(x))$ — решение задачи (1)–(4), ближайшее к z_0 . Для приближенного решения задачи (9)–(10) методом Эйлера введём дискретное множество узлов

$\{t_k\} = \{\tau_k = t_{k+1} - t_k, k = 0, 1, 2, \dots\}$ и запишем приближенные выражения для соотношений (11)

$$\begin{cases} \left. \frac{dy(x, t)}{dt} \right|_{t=t_k} \approx v(t_k) = \frac{y(x, t_{k+1}) - y(x, t_k)}{\tau_k}, \\ \left. \frac{d\lambda(t)}{dt} \right|_{t=t_k} \approx \mu(t_k) = \frac{\lambda(t_{k+1}) - \lambda(t_k)}{\tau_k}. \end{cases} \quad (12)$$

Введём обозначения $\lambda_k = \lambda(t_k)$, $y_k = y(x, t_k)$, $\mu_k = \mu(t_k)$, $v_k = v(x, t_k)$. При дискретной аппроксимации по переменной x эволюционного уравнения (9) использовались трёхточечные разностные и квадратурные формулы с точностью аппроксимации порядка $O(h^2)$ на равномерной сетке $\omega_h \in [a, b]$. Для вычисления λ_{k+1}, y_{k+1} при известных λ_k, y_k на каждом шаге итерации необходимо:

1) решить относительно w_k уравнение

$$\varphi^{(j)'}(\lambda_k, y_k)w_k = -\varphi_\lambda^{(j)}y_k, \quad j = 1, 2, 3; \quad (13)$$

2) вычислить значение μ_k из уравнения $(\cdot, \cdot) = \int_a^b (\cdot, \cdot) dx$

$$\mu_k = \frac{1 + (y_k, y_k)}{2(y_k, w_k)}; \quad (14)$$

3) найти новые приближения λ_{k+1} и y_{k+1} , используя (12), т.е.

$$\lambda_{k+1} = \lambda_k + \tau_k \mu_k, \quad y_{k+1} = (1 - \tau_k)y_k + \tau_k \mu_k w_k, \quad (15)$$

где τ_k определяется специальными алгоритмами. Итерации прекращаются при выполнении условия

$$\delta_k < \varepsilon, \quad (16)$$

где δ_k определяется по одной из формул:

$$\delta_k = \max_j \max_x |\varphi^{(j)}(\lambda_k, y_k(x))|, \quad j = 1, 2, 3, \quad (17)$$

$$\delta_k = \left[\int_a^b \varphi^{(1)2}(\lambda_k, y_k(x)) dx \right], \quad (18)$$

$x \in \omega_h, \varepsilon > 0$ — заданное малое число.

2.3. 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 $0 < \tau_0 \leq 1$.

$$1. \tau_k \equiv \tau_0. \quad (19)$$

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.

$$2. \tau_k = \begin{cases} \min(1, 2\tau_{k-1}), & \text{если } \delta_k < \delta_{k-1}, \\ \max(\tau_0, \tau_{k-1}/2), & \text{если } \delta_k \geq \delta_{k-1}, \end{cases} \quad (20)$$

где δ_k определяется по формуле (17).

Этот алгоритм аналогичен широко распространённому 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.

$$3. \tau_k = \begin{cases} \min(1, \tau_{k-1} \frac{\delta_{k-1}}{\delta_k}), & \text{если } \delta_k < \delta_{k-1}, \\ \max(\tau_0, \tau_{k-1} \frac{\delta_{k-1}}{\delta_k}), & \text{если } \delta_k \geq \delta_{k-1}, \end{cases} \quad (21)$$

где δ_k определяется по формуле (17). Этот алгоритм более устойчив и обеспечивает сходимость в достаточно широкой области начальных приближений, однако зависит от задания τ_0 .

$$4. \tau_k = \frac{\delta_{k-1}}{\delta_{k-1} + \delta_k(1)}, \quad (22)$$

где δ_k определяется по формуле (18), $\delta_k(1)$ — невязка на k -той итерации для $\tau_k = 1$. Это алгоритм оптимального выбора τ_k [7].

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 объёма вычислений.

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

2.4. Алгоритмы вычисления значения параметра τ_0

В данной работе предложены и программно реализованы два новых алгоритма вычисления τ_0 :

$$I. \tau_0 = \frac{\delta_0}{2\delta_1(1)}, \quad (23)$$

$$II. \tau_0 = \frac{1}{2} \left| 1 - \frac{\delta_1(1)}{\delta_0} \right|, \quad (24)$$

где невязка δ_k определяется по формуле (17), $\delta_1(1)$ — невязка на первой итерации ($k = 1$) для $\tau_1 = 1$. Ограничение $0, 1 \leq \tau_0 \leq 1$.

Их применение позволяет сократить число итераций при решении конкретных задач, что видно из сравнения результатов, представленных в следующем разделе в табл. 1 для уравнения Шрёдингера с потенциалом Морзе и табл. 2 при решении уравнения Лежандра.

3. Численные результаты

Программа SLIPM проверена на ряде тестов [8]. Здесь приведены результаты её применения для решения трёх задач.

Пример 1. Уравнение Шрёдингера с потенциалом Морзе [2]. В этом примере задача (1)–(4) имеет коэффициенты уравнения $p(x) \equiv 0$, $r(x) \equiv 1$,

$$q(x) = -2MD \left(e^{-2a_1(x-x_0)} - 2e^{-a_1(x-x_0)} \right)$$

на интервале (a, b) и граничные условия

$$d_1 = d_2 = 1; \quad f_1 = \sqrt{\lambda - \sqrt{2MD}}e^{a_1x_0}, \quad f_2 = \sqrt{\lambda - \sqrt{2MD}}e^{-a_1(b-x_0)},$$

где значения констант $a_1 = 0.67$, $x_0 = 2.15$, $M = 4.69$, $D = 0.1005$.

Таблица 1

	$\lambda_h = 0,435312$	$\lambda_{h/2} = 0,435312$	$\lambda_{h/4} = 0,435312$	$\sigma_\lambda \simeq 4$
x	y_h	$y_{h/2}$	$y_{h/4}$	$\sigma_y(25)$
0	$0,188079e - 1$	$0,188090e - 1$	$0,188092e - 1$	3,996
4	0,461810	0,461811	0,461811	3,999
8	$0,492263e - 1$	$0,492265e - 1$	$0,492266e - 1$	4,0
12	$0,3613786e - 2$	$0,361380e - 2$	$0,361380e - 2$	4,0
16	$0,258596e - 3$	$0,258597e - 3$	$0,258597e - 3$	4,001
20	$0,184721e - 4$	$0,184723e - 4$	$0,184723e - 4$	4,004
δ_0	60,097359	60,172738	62,171030	
δ_{13}	$\tau_0 = 0,1$ $1,23e - 11$	$k = 13$ $1,35e - 11$	$1,37e - 11$	
δ_3	$\tau_0 = 1$ (23) $2,604e - 8$	$k = 3$ $1,551e - 8$	$1,362e - 8$	
δ_5	$\tau_0 = 0,4499$ (24) $2,877e - 13$	$k = 5$ $2,671e - 13$	$3,063e - 13$	

Таблица 2

	$\lambda_h = -6,00$	$\lambda_{h/2} = -6,00$	$\lambda_{h/4} = -6,00$	$\lambda_a = -6,0$
x	y_h	$y_{h/2}$	$y_{h/4}$	$\sigma(25)$
-1,0	1,581136	1,581138	1,581138	4,0
-0,6	$0,632454e - 1$	$0,632455e - 1$	$0,632455e - 1$	4,0
-0,4	-0,411095	-0,411096	-0,411096	4,001
0,0	-0,790568	-0,790569	-0,790569	4,0
δ_0	39,847	79,521	158,869	
δ_{13}	$\tau_0 = 0,1$ $8,85e - 11$	$k = 13$ $1,78e - 10$	$3,57e - 10$	
δ_4	$\tau_0 = 1$ (23) $2,63e - 9$	$k = 4$ $5,26e - 9$	$1,05e - 8$	
δ_2	$\tau_0 = 0,45$ (24) $2,63e - 9$	$k = 4$ $9,01e - 13$	$2,07e - 11$	

Аналитическое решение с безузловой волновой функцией (число узлов $n = 0$) имеет вид:

$$\lambda = 2MD \left[1 - \frac{a_1}{2\sqrt{2MD}} \right]^2, \quad y(x) = C\xi^\rho e^{-\frac{\xi}{2}},$$

где

$$\xi = 2\delta e^{-a_1(x-x_a)}, \quad \delta = \frac{\sqrt{2MD}}{a_1}, \quad \rho = \frac{\sqrt{\lambda}}{a_1}.$$

Для указанного набора констант a_1 , x_0 , M , D аналитическое значение $\lambda_a = 0,4353114734$.

С помощью процедуры NEWTON0 получается начальное приближения y_0^{Newt} , λ_0^{Newt} , дающее невязку $\delta_0 < \varepsilon$, $\varepsilon = 0,0001$, и выход из процедуры SLIPM происходит сразу же после (0) итераций. Для проверки области сходимости метода и достоинств новых алгоритмов выбора τ_0 начальное приближение z_0 специально сделано плохим: $y_0 = (C_1 * y_0^{\text{Newt}} + C_2; C_1 = 1; C_2 = 0,3); \lambda_0 = 1000,435$. В

первой части табл. 1 приведены вычисленные от этого начального приближения собственные функции $y_h(x)$ и собственные значения λ_h на последовательности $\Delta_h = (h, h/2, h/4)$ вдвое сгущающихся сеток при значениях $a = 0, b = 20, h = 0,008333(N = 2401)$. Далее показаны начальное значение невязки $\delta_0 \approx 60$, эмпирически заданное начальное значение параметра $\tau_0 = 0,1$, количество итераций $k = 13$ и значения δ_k после окончания итерационного процесса по условию (17). Параметр τ_k определяется по формуле (21). В последнем столбце приводятся значения отношения Рунге для $z = (\lambda, y(x))$:

$$\sigma_z = \frac{z_h - z_{h/2}}{z_{h/2} - z_{h/4}}. \quad (25)$$

Величина выражения (25) должна быть порядка $O(2^p)$, где p — точность дискретной аппроксимации задачи (1)–(4). Видно, что точность полученного разностного решения задачи (1)–(4) имеет порядок $O(h^2)$.

Во второй части табл. 1 представлены результаты двух расчётов, отличающихся новым способом счета начального значения τ_0 по формулам (23) и (24). Видно, что использование алгоритмов (23), (24) привело к сокращению числа итераций с 13 для $\tau_0 = 0,1$ до 3 и 5, соответственно.

На рис. 1 показан ход итерационного процесса.

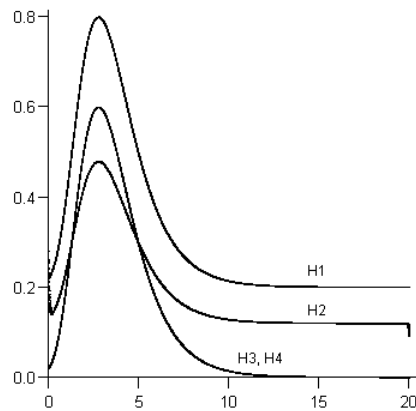


Рис. 1. Аналитика — **Н4**, $\lambda_a = 0.4353114$, $k = 0$ — **Н1**, $\lambda_0 = 1000.435$, $k = 8$ — **Н2**, $\lambda_8 = 270.779$, $k = 13$ — **Н3**, $\lambda_{13} = 0.4353125$, (**Н3** и **Н4** — совпадают), k — номер итерации $\tau_0 = 0.1$, τ_k определяется по формуле (21)

Пример 2. Уравнение Лежандра. В этом примере $p(x) \neq 0$. Задача (1)–(4) имеет вид

$$y''(x) - \frac{2x}{1-x^2}y'(x) - \frac{\lambda}{1-x^2}y(x) = 0, \quad x \in (-1, 1),$$

граничные условия

$$\begin{aligned} y'(x) - \frac{\lambda}{2}y(x) &= 0, & x = -1; \\ y'(x) + \frac{\lambda}{2}y(x) &= 0, & x = +1. \end{aligned}$$

Аналитические собственные значения данной задачи определяются по формуле $\lambda_n = -n(n+1)$, а собственные функции могут быть найдены, например, по следующей аналитической формуле в системе MAPLE: $p_n := \text{LegendreP}[n, x]$. $p_0 = 1$; $p_1 = x$; $p_2 = \frac{1}{2}(3x^2 - 1), \dots$

В первой части табл. 2 приведены результаты вычисления решения для $n = 2$, полученные на последовательности $\Delta_h = (h, h/2, h/4)$ вдвое сгущающихся сеток

$h = 0,0125 (N = 1601)$, подтверждающие порядок $O(h^2)$ сходимости приближенного решения задачи (1)–(4). В качестве начального приближения взято специально ухудшенное известное аналитическое решение, которое можно получить также с помощью процедуры NEWTON0: $y_0 = (C_1 * y_0^{\text{Newt}} + C_2; C_1 = 2; C_2 = 0, 1); \lambda_0 = -5$. Показаны начальные значения невязки δ_0 .

Во второй части табл. 2 представлены результаты трёх расчётов, отличающихся способом счета начального значения τ_0 : $\tau_0 = 0, 1$ и по формулам (23), (24). Здесь k указывает количество итераций, потребовавшихся для вычисления требуемых решений с заданной точностью $\varepsilon = 0,0001$. Видно, что использование алгоритмов (23), (24) привело к сокращению числа итераций с 13 для $\tau_0 = 0, 1$ до 4 для обоих новых алгоритмов. Значения параметра τ_k и невязки δ_k определяются по формулам (21) и (17).

На рис. 2 показан ход итерационного процесса.

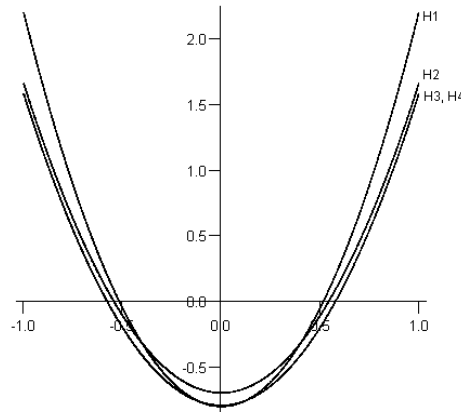


Рис. 2. Аналитика — **H4**, $\lambda_a = -6$, $k = 0$ — **H1** $\lambda_0 = -5$, $k = 4$ — **H2** $\lambda_4 = -5,381$, $k = 13$ — **H3**, $\lambda_{13} = -6,0$ (**H3** и **H4** — совпадают) k — номер итерации $\tau_0 = 0, 1$ τ_k определяется по формуле (21)

Пример 3. Уравнение Шрёдингера

$$y''(x) + \left(\frac{2}{x} - \lambda\right)y(x) = 0 \quad x \in (0, \infty), \quad (26)$$

с краевыми условиями $y(0) = 0; y(\infty) = 0$. Использование процедуры SLIPM от начального приближения $\lambda_0 = 0,9$ на интервале $(0, 20)$ даёт собственное значение $\lambda = 1$. Соответствующая волновая функция выдаётся в дискретном виде на сетке по x . Отметим, что использование процедуры dsolve в СКА Maple для решения этого уравнения даёт общее решение в аналитическом виде, зависящем от λ , через функции Уиттекера:

$$y(x) = C_1 \text{WhittakerM} \left(\frac{1}{\lambda^{1/2}}, 1/2, 2\lambda^{1/2}x \right) + C_2 \text{WhittakerW} \left(\frac{1}{\lambda^{1/2}}, 1/2, 2\lambda^{1/2}x \right). \quad (27)$$

Подстановка значения $\lambda = 1$ в dsolve даёт аналитическое решение $y(x) = xe^{-x}$.

4. Задача Штурма–Лиувилля в СКА Maple

Для решения задачи Штурма–Лиувилля для ряда уравнений (Лежандра, Уиттекера, Эйлера, Чебышева и др.) [9] в системе Maple при использовании процедуры `dsolve` можно получить решение в аналитическом виде, зависящем от λ . Если мы знаем собственное значение, то с помощью `dsolve` можно найти соответствующую собственную функцию. Однако, если не известно собственное значение, что обычно бывает на практике при исследовании многих уравнений, то этот подход не работает. Поэтому представляет интерес использование тандема: сначала с помощью SLIPM найти собственное значение, затем использовать его в крайних условиях `dsolve` для вычисления собственной функции в аналитическом виде, как это показано в третьем примере для уравнения Уиттекера.

5. Заключение

- На базе НАМН создана программа SLIPM на языке MAPLE, предназначенная для численного решения частичной проблемы Штурма–Лиувилля для линейного дифференциального оператора второго порядка с однородными граничными условиями.
- Предложены и программно реализованы два новых алгоритма вычисления начального значения итерационного параметра τ_0 . На приведённых примерах продемонстрированы их достоинства: расширилась область применимости НАМН и сократилось количество итераций.
- На ряде тестов продемонстрирована заданная точность порядка $O(h^2)$ итерационных ньютоновских схем (h — шаг дискретной сетки по x_i). Использование экстраполяции по Ричардсону с использованием результатов на последовательности трёх сгущающихся сеток позволяет повысить точность до порядка $O(h^4)$.
- Возможности графических пакетов СКА MAPLE позволяют организовать хорошую визуализацию итерационного процесса в SLIPM.
- Использование вычисленного с помощью процедуры SLIPM собственного значения в процедуре `dsolve` открывает новые возможности для получения собственной функции в аналитическом виде.

Литература

1. *Puzynin I. V., Puzynina T. P.* SLIP1 — Program for the Numerical Solution of the Sturm–Liouville Problem Basing on the Continuous Analog of the Newton Method // Collection of scientific papers in collaboration of JINR, Dubna, USSR and Central Research Institute for Physics, Budapest, Hungary, KFKI-74-34. — 1974. — Pp. 93–112.
2. *Puzynin I. V., Puzynina T. P., Strizh T. A.* Program for Numerical Solution of the Sturm–Liouville Problem. — JINR Report, P11-87-332. — 1987.
3. *Пузынин И. В. и др.* Обобщённый непрерывный аналог метода Ньютона для численного исследования некоторых квантово-полевых моделей // ЭЧАЯ. — 1999. — Т. 30. — С. 210–265.
4. *Пузынин И. В. и др.* О методах вычислительной физики для исследования моделей сложных физических систем // ЭЧАЯ. — 2007. — Т. 38. — С. 144–232.
5. *Уилкинсон Дж. Х.* Алгебраическая проблема собственных значений. — М.: Наука, 1970.
6. *Акишин П. Г., Пузынин И. В.* Реализация метода Ньютона в разностной задаче Штурма–Лиувилля. — 1977.
7. *Ермаков В. В., Калиткин Н. Н.* Оптимальный шаг и регуляризация метода Ньютона // ЖВМиМФ. — 1981. — Т. 21. — С. 491.
8. *Bailey P. B., Everitt W. N., Zettl A.* The SLEIGN2 Sturm–Liouville Code // ACM Trans. Math. Software. — 2001. — Vol. 21. — Pp. 143–192.

9. *Lopez R. J.* Emeritus Professor of Mathematics and Maple Fellow Classroom Tips and Techniques Eigenvalue Problems for ODEs. Part1-Part3. — Maplesoft.

UDC 517.9, 519.6, 539.12

**SLIPM — MAPLE Programm for Numerical Solution of
Sturm–Liouville Partial Problem with the Help of the
Continuous Analogue of Newton’s Method**

I. V. Puzynin, T. P. Puzynina, V. T. Thach

A computer program SLIPM (Sturm–Liouville Problem in MAPLE) has been designed in the language of computer algebras MAPLE. It is intended for the numerical solution of Sturm–Liouville partial problem with the help of the continuous analogue of Newton’s method, i.e., for calculating the eigenvalue of a linear second-order differential operator and a corresponding eigenfunction satisfying homogeneous boundary conditions. SLIPM is the development of the computer codes SLIP1 and SLIPH4 written in the Fortran language; it is added by a new procedure of calculating an initial approach to the solution and by new ways of calculating the initial value of the iterative parameter τ_0 .

Key words and phrases: computational mathematics, continuous analogue of Newton’s method, computer algebras system, MAPLE.