УДК 519.688

Г. Г. Акниев, Р. М. Гаджимирзаев

Численный метод решения задачи Коши для ОДУ посредством системы полиномов, порожденной системой модифицированных полиномов Лагерра

В настоящей работе рассматривается численная реализация итерационного метода решения задачи Коши для ОДУ, основанного на представлении решения в виде ряда Фурье по системе полиномов $L_{1,n}(x; b)$ ($n = 0, 1, \dots$), ортонормированной относительно скалярного произведения типа Соболева

$$\langle f, g \rangle = f(0)g(0) + \int_0^\infty f'(x)g'(x)\rho(x; b)dx$$

и порожденной системой модифицированных полиномов Лагерра

$\{L_n(x; b)\}_{n=0}^\infty$, где $b > 0$. При приближенном вычислении коэффициентов Фурье искомого решения используется квадратурная формула Гаусса – Лагерра.

Библиография: 3 названия.

In this paper, we consider a numerical realization of an iterative method for solving the Cauchy problem for ordinary differential equations, based on representing the solution in the form of a Fourier series by the system of polynomials $\{L_{1,n}(x; b)\}_{n=0}^\infty$, orthonormal with respect to the Sobolev-type inner product

$$\langle f, g \rangle = f(0)g(0) + \int_0^\infty f'(x)g'(x)\rho(x; b)dx$$

and generated by the system of modified Laguerre polynomials $\{L_n(x; b)\}_{n=0}^\infty$, where $b > 0$. In the approximate calculation of the Fourier coefficients of the desired solution, the Gauss – Laguerre quadrature formula is used.

Bibliography: 3 items.

Ключевые слова: полиномы Лагерра, обыкновенное дифференциальное уравнение (ОДУ), задача Коши, скалярное произведение типа Соболева, квадратурная формула Гаусса – Лагерра.

Keywords: Laguerre polynomials, ordinary differential equation (ODE), Cauchy problem, inner product of Sobolev-type, Gauss – Laguerre quadrature formula.

1. Введение

Пусть $b > 0$, $\rho_b = \rho(x; b) = e^{-bx}$, $\mathcal{L}_{\rho_b}^2$ – пространство измеримых функций f , определенных на полуоси $[0, \infty)$ и таких, что

$$\|f\|_{\mathcal{L}_{\rho_b}^2} = \left(\int_0^\infty |f(x)|^2 \rho(x; b) dx \right)^{1/2} < \infty.$$

Через $W_{\mathcal{L}_{\rho_b}^2}^1$ обозначим пространство Соболева, состоящее из абсолютно непрерывных на произвольном сегменте $[c, d] \subset [0, \infty)$ функций f , для которых $f' \in \mathcal{L}_{\rho_b}^2$. Далее, следуя [1], обозначим через $L_n(x; b) = \sqrt{b} L_n(bx)$ ($n = 0, 1, \dots$) модифицированные полиномы Лагерра, ортонормированные на промежутке $[0, \infty)$ относительно веса $\rho(x; b) = e^{-bx}$, т. е.

$$\int_0^\infty L_m(x; b) L_n(x; b) \rho(x; b) dx = \delta_{mn},$$

где δ_{mn} – символ Кронекера.

В той же работе была введена система функций $L_{1,n}(x; b)$ ($n = 0, 1, \dots$), ортонормированная относительно скалярного произведения типа Соболева

$$\langle f, g \rangle = f(0)g(0) + \int_0^\infty f'(x)g'(x)\rho(x; b)dx$$

и порожденная посредством равенств

$$L_{1,0}(x; b) = 1, \quad L_{1,1+n}(x; b) = \int_0^x L_n(t; b) dt, \quad n = 0, 1, \dots$$

¹Работа выполнена при финансовой поддержке РФФИ (проект 18-31-00477 мол_а)

системой полиномов $\{L_n(x; b)\}_{n=0}^{\infty}$. Кроме того, было показано [1], что ряды Фурье по системам $\{L_n(x; b)\}_{n=0}^{\infty}$ и $\{L_{1,n}(x; b)\}_{n=0}^{\infty}$ являются весьма эффективным инструментом численного решения систем линейных и нелинейных дифференциальных уравнений с помощью метода простых итераций. В настоящей работе мы рассмотрим численную реализацию этого метода на примере задачи Коши для ОДУ следующего вида

$$y'(x) = f(x, y), \quad y(0) = y^0, \quad 0 \leq x < 1. \quad (1)$$

Функцию $f(x, y)$ будем считать непрерывной в некоторой замкнутой области $\bar{G}$ переменных (x, y) , содержащей точку $(0, y_0)$. Кроме того, мы будем предполагать, что $[0, 1] \times \mathbb{R} \subset \bar{G}$. Будем также полагать, что по переменной y функция $f(x, y)$ удовлетворяет условию Липшица

$$|f(x, y_1) - f(x, y_2)| \leq \lambda |y_1 - y_2|, \quad 0 \leq x < 1.$$

Посредством замены переменных $x = 1 - e^{-at}$, $a > 0$ отобразим полуось $[0, \infty)$ на полуинтервал $[0, 1)$. Относительно новой переменной $t \in [0, \infty)$ уравнение (1) принимает следующий вид

$$\eta'(t) = ae^{-at} f(1 - e^{-at}, \eta(t)), \quad \eta(0) = y^0, \quad 0 \leq t < \infty,$$

где $\eta(t) = y(1 - e^{-at})$.

Вышеупомянутый итерационный метод основан на построении сжимающего оператора A , действующего в гильбертовом пространстве l_2 , состоящем из последовательностей $d = (d_0, d_1, \dots)$ с нормой $\|d\|_{l_2} = \sqrt{\sum_{k=0}^{\infty} d_k^2} < \infty$. Оператор A сконструирован таким образом, что координатами его неподвижной точки будут коэффициенты Фурье разложения искомого решения $\eta(t)$ в ряд по системе $\{L_{1,n}(x; b)\}_{n=0}^{\infty}$. Мы не будем останавливаться на подробном описании этого итерационного процесса и построения сжимающего оператора, а только приведем соотношения на которых основана общая идея построения оператора A . Прежде всего заметим, что функция $y = y(x)$ непрерывно дифференцируема и ограничена на $[0, 1]$. Отсюда следует, что функция $\eta(t) = y(1 - e^{-at})$ непрерывно дифференцируема и ограничена на полуоси $[0, \infty)$. Следовательно, $\eta(t) \in W_{\mathcal{L}_{\rho_b}}^1$. Поэтому (см. [1], следствие 2) для функции $\eta(t)$ справедли-

во разложение

$$\eta(t) = \eta(0) + \sum_{k=0}^{\infty} c_{1,k+1}(\eta) L_{1,k+1}(t; b), \quad t \in [0, \infty), \quad (2)$$

где

$$c_{1,k+1}(\eta) = \int_0^{\infty} \eta'(\tau) L_k(\tau; b) \rho_b(\tau) d\tau, \quad k = 0, 1, \dots$$

Далее, поскольку система $\{L_n(t; b)\}_{n=0}^{\infty}$ полна в пространстве $\mathcal{L}_{\rho_b}^2$, то в метрике этого пространства имеют место равенства

$$\eta'(t) = \sum_{k=0}^{\infty} c_{1,k+1}(\eta) L_k(t; b),$$

$$q(t) = e^{-at} f(1 - e^{-at}, \eta(t)) = \sum_{k=0}^{\infty} c_k(q) L_k(t; b),$$

где

$$c_k(q) = \int_0^{\infty} q(\tau) L_k(\tau; b) \rho_b(\tau) d\tau = \frac{c_{1,k+1}(\eta)}{a} \quad (k \geq 0).$$

Отсюда и из (2) для $k \geq 0$ имеем

$$c_{1,k+1}(\eta) = a \int_0^{\infty} e^{-a\tau} f\left(1 - e^{-a\tau}, \eta(0) + \sum_{j=0}^{\infty} c_{1,j+1}(\eta) L_{1,j+1}(\tau; b)\right) L_k(\tau; b) e^{-b\tau} d\tau.$$

Правая часть последнего равенства и есть вышеупомянутый оператор A , сопоставляющий точке $(c_0, c_1, \dots) \in l_2$ точку $(c'_0, c'_1, \dots) \in l_2$ по правилу

$$c'_k = a \int_0^{\infty} e^{-a\tau} f\left(1 - e^{-a\tau}, \eta(0) + \sum_{j=0}^{\infty} c_j L_{1,j+1}(\tau; b)\right) L_k(\tau; b) e^{-b\tau} d\tau.$$

Рассмотрим теперь конечномерный аналог оператора A , сопоставляющий точке $(c_0, c_1, \dots, c_N) \in \mathbb{R}^{N+1}$ точку $(c'_0, c'_1, \dots, c'_N) \in \mathbb{R}^{N+1}$ по правилу

$$c'_k = a \int_0^{\infty} e^{-a\tau} f\left(1 - e^{-a\tau}, \eta(0) + \sum_{j=0}^N c_j L_{1,j+1}(\tau; b)\right) L_k(\tau; b) e^{-b\tau} d\tau, \quad k = \overline{0, N}. \quad (3)$$

В работе [1] было показано, что оператор A , соответственно и его конечномерный аналог (3), будут сжимающими, если

$$\sqrt{I(a, b)}a\lambda < 1,$$

где

$$I(a, b) = \int_0^\infty e^{-(2a+b)t} \sum_{k=0}^\infty (L_{1,k+1}(t; b))^2 dt.$$

В той же работе (см. лемму 2) была получена оценка сверху для величины $I(a, b)$. Но, как будет показано ниже (см. (5)), для этой величины можно получить точное равенство.

2. Вспомогательные утверждения

ЛЕММА 1. Пусть $b > 0$. Тогда имеет место равенство

$$\sum_{k=0}^\infty (L_{1,k+1}(x; b))^2 = \frac{e^{bx} - 1}{b}. \quad (4)$$

ДОКАЗАТЕЛЬСТВО. В работе [1] для полиномов $L_{1,k+1}(x; b)$ была получена следующая формула

$$L_{1,k+1}(x; b) = \frac{\sqrt{bx} L_k^1(bx)}{k+1}.$$

Отсюда и из равенства [2, с. 623, формула 6]

$$\sum_{k=0}^\infty \frac{k! L_k^\alpha(x) L_k^\alpha(y)}{(\alpha+1)_k (k+1)} = \frac{\alpha}{(xy)^\alpha} e^{x+y} \gamma(\alpha, x) \Gamma(\alpha, y), \quad 0 < x \leq y,$$

где $\gamma(\alpha, x)$ и $\Gamma(\alpha, y)$ неполные гамма-функции, определяемые равенствами

$$\gamma(\alpha, x) = \int_0^x t^{\alpha-1} e^{-t} dt, \quad \Gamma(\alpha, x) = \int_x^\infty t^{\alpha-1} e^{-t} dt,$$

получаем

$$\sum_{k=0}^\infty (L_{1,k+1}(x; b))^2 = \sum_{k=0}^\infty \left(\frac{\sqrt{bx} L_k^1(bx)}{k+1} \right)^2 =$$

$$bx^2(bxbx)^{-1}e^{bx+bx}\gamma(1, bx)\Gamma(1, bx) = \frac{e^{2bx}}{b} \int_0^{bx} e^{-t} dt \cdot \int_{bx}^{\infty} e^{-t} dt = \frac{e^{2bx}}{b} (1 - e^{-bx}) e^{-bx} = \frac{e^{bx} - 1}{b}.$$

□

СЛЕДСТВИЕ 1. Для произвольных $a, b > 0$ имеет место равенство

$$I(a, b) = \frac{1}{2a(2a + b)}. \quad (5)$$

ДОКАЗАТЕЛЬСТВО. Из (4) имеем

$$I(a, b) = \int_0^{\infty} e^{-(2a+b)t} \frac{e^{bt} - 1}{b} dt = \frac{1}{b} \left(\int_0^{\infty} e^{-2at} dt - \int_0^{\infty} e^{-(2a+b)t} dt \right) = \frac{1}{b} \left(\frac{1}{2a} - \frac{1}{2a + b} \right) = \frac{1}{2a(2a + b)}.$$

□

3. Численная реализация

Правую часть равенства (3) будем вычислять с помощью квадратурной формулы Гаусса – Лагерра [3, стр. 143]

$$c'_k = a \int_0^{\infty} e^{-a\tau} f \left(1 - e^{-a\tau}, \eta(0) + \sum_{j=0}^N c_j L_{1,j+1}(\tau; b) \right) L_k(\tau; b) e^{-b\tau} d\tau \approx a \sum_{i=1}^M A_i^M g(\tau_i^M), \quad k = \overline{0, N}, \quad (6)$$

где

$$g(\tau) = f \left(1 - e^{-a\tau}, \eta(0) + \sum_{j=0}^N c_j L_{1,j+1}(\tau; b) \right) L_k(\tau; b) e^{(1-a-b)\tau}, \quad (7)$$

τ_i^M – корни полинома Лагерра $L_M(x)$, A_i^M – веса, определяемые равенством

$$A_i^M = \frac{1}{\tau_i^M} \left(\frac{1}{L'_M(\tau_i^M)} \right)^2 = \frac{1}{\tau_i^M} \left(\frac{1}{L_{M-1}^1(\tau_i^M)} \right)^2. \quad (8)$$

Для вычисления значений полинома $L_{1,j+1}(\tau; b)$ в узлах τ_i^M будем пользоваться формулой

$$L_{1,j+1}(\tau; b) = \frac{\sqrt{b\tau} L_j^1(b\tau)}{j+1}, \quad j = 0, 1, \dots$$

Теперь приведем код основных функций, используемых при приближенном вычислении коэффициентов c'_k с помощью квадратурной формулы (6). При этом в коде вместо N и M используются обозначения n_part и $_n$ соответственно.

Нижеприведенная функция **find_all_roots_of_laguerre** находит нули полинома $L_k^\alpha(x)$ с заданной точностью *epsilon*. При этом для отделения нулей используется свойство перемежаемости нулей ортогональных полиномов.

```
def find_all_roots_of_laguerre(k, alpha, epsilon = 1e-12):
    max_bound = 4 * k + 2
    segments = list(range(k+2))
    segments[1] = [(0, max_bound)]
    roots = list(range(k+2))

    for j in range(1, k + 1):
        layer_roots = []
        for s in segments[j]:
            left_end = s[0]
            right_end = s[1]
            segment_root =
                find_root_of_laguerre
                    (j, alpha, left_end, right_end, epsilon)
            layer_roots.append(segment_root)

        next_segments = [(0, layer_roots[0])]
        for i in range(len(layer_roots) - 1):
            next_segments.append((layer_roots[i], layer_roots[i + 1]))
        next_segments.append((layer_roots[-1], max_bound))

        segments[j + 1] = next_segments
        roots[j] = layer_roots
```

```
return roots[k]
```

Используя найденные нули полинома Лагерра, функция **all_weights** вычисляет веса A_i^M квадратурной формулы Гаусса – Лагерра по формуле (8).

```
def root_to_weight(root, k, alpha):
    return 1.0 / root / ((laguerre(k - 1, 1, root)) ** 2)

def all_weights(_k, _alpha):
    _roots = find_all_roots_of_laguerre(_k, _alpha)
    return [root_to_weight(root, _k, _alpha) for root in _roots]
```

Функция **sobolev_laguerre** вычисляет значение полинома $L_{1,j}(x; b)$ в точке τ .

```
def sobolev_laguerre(tau, b, j):
    return math.sqrt(b) * tau * laguerre(j - 1, 1, b * tau) / j
```

Здесь мы вычисляем значение функции g , определенной равенством (7), в точке τ .

```
def g(tau, _k, _a, _b, _c, _alpha, n_part, f, y0):
    first_arg = 1 - math.exp(-_a * tau)
    s0 = sum([_c[j] * sobolev_laguerre(tau, _b, j + 1)
              for j in range(n_part + 1)])
    second_arg_sum_c_sobolev_laguerre = y0 + s0

    s1 = math.sqrt(_b) * laguerre(_k, 0, tau * _b)
        * math.exp((1 - _a - _b) * tau)
    return f(first_arg, second_arg_sum_c_sobolev_laguerre)*s1
```

Функция **perform_iteration_on_cs** производит одну итерацию по вычислению коэффициентов c'_k .


```
def perform_iteration_on_c(c0, _a, _b, _k, _alpha, n_part, _n, f, y0):
    weights = all_weights(_n, _alpha)
    _roots = find_all_roots_of_laguerre(_n, _alpha)
    gs = [g(root, _k, _a, _b, c0, _alpha, n_part, f, y0)
          for root in _roots]

    return _a * sum([weights[i] * gs[i] for i in range(_n)])

def perform_iteration_on_cs(c0, _a, _b, _alpha, n_part, _n, f, y0):
    r=[perform_iteration_on_c(c0, _a, _b, k, _alpha, n_part, _n, f, y0)
      for k in range(n_part+1)]

    return r
```

Функция **find_cs** выполняет итерации по вычислению коэффициентов c'_k , пока расстояние между коэффициентами не окажется меньше порогового значения.

```
def find_cs(_k, _a, _b, _alpha, threshold, n_part, _n):
    c0 = [1/(i+1) for i in range(n_part+1)]
    c1 = perform_iteration_on_cs(c0, _a, _b, _alpha, n_part, _n)

    while distance(c0, c1) > threshold:
        c0 = c1
        c1 = perform_iteration_on_cs(c0, _a, _b, _alpha, n_part, _n)

    return c1
```

Используя коэффициенты c'_k , функция **find_solution** находит приближенное решение задачи Коши.

```
def find_solution(t, _k, _a, _b, _alpha, c, n_part, _n, y0):
    return y0 + sum([c[k] * sobolev_laguerre(t, _b, k+1)
                    for k in range(n_part+1)]).
```

4. Компьютерные эксперименты

В этом пункте на конкретных примерах покажем результат применения вышеприведенной программы при $a = b = 1$.

ПРИМЕР 1. *Решить задачу Коши*

$$y' - y = e^x, \quad y(0) = 2, \quad 0 \leq x < 1. \quad (9)$$

Решение. Точное решение этой задачи имеет вид: $y(x) = (x + 2)e^x$.
Ниже на рисунке 1 приведены первые 4 итерации применения программы к задаче (9) при $N = 15$ и $M = 20$.

Рис. 1: Красным цветом обозначено точное решение, синим – приближенное. На последнем графике иллюстрируется сравнение приближенного решения, полученного после 4 итерации, с точным вблизи правого конца.

ПРИМЕР 2. *Решить задачу Коши*

$$y' + 2xy = xe^{-x^2}, \quad y(0) = 1, \quad 0 \leq x < 1.$$

Решение. Точное решение этой задачи имеет вид: $y(x) = \left(\frac{x^2}{2} + 1\right)e^{-x^2}$.
На рисунке 2 приведен результат применения программы для первых 6 итераций при $N = 20$ и $M = 25$.

Рис. 2: Красным цветом обозначено точное решение, синим – приближенное. На последнем графике иллюстрируется сравнение приближенного решения, полученного после 6 итерации, с точным вблизи правого конца.

ЗАМЕЧАНИЕ 1. Как показывают компьютерные эксперименты, частичные суммы ряда Фурье плохо приближают искомое решение в окрестности точки $x = 1$. Это связано с тем, что ряд Фурье функции $f \in W_{\mathcal{L}_{\rho_b}}^1$ по системе $\{L_{1,n}(t; b)\}_{n=0}^{\infty}$

$$f(t) = f(0) + \sum_{n=0}^{\infty} c_{1,n+1}(f) L_{1,n+1}(t; b), \quad 0 \leq t < \infty,$$

сходится равномерно на любом отрезке $[0, A]$ (см. [1, следствие 2]). А при отображении полуоси $[0, \infty)$ на полуинтервал $[0, 1)$ посредством замены $x = 1 - e^{-at}$, $a > 0$, в окрестность точки $x = 1$ попадают достаточно большие значения переменной t .

Список литературы

- [1] Шарапудинов И.И., Шарапудинов Т.И. Ортогональные по Соболеву полиномы, порожденные модифицированными полиномами Лагерра, и задача Коши для систем ОДУ // Дагестанские электронные математические известия. 2018. Вып. 10. С. 23-40.
- [2] Прудников А.П., Брычков Ю.А., Маричев О.И. Интегралы и ряды. В 3 т. Т. 2. Специальные функции. — 2-е изд., исправ. М.: ФИЗМАТЛИТ. 2003. - 664 с.

- [3] Крылов В.И. Приближенное вычисление интегралов. - 2-е изд., дополненное. М.: Наука. 1967. - 500 с.

Г. Г. Акцииев (G. G. Akniyev)

Отдел математики и информатики Дагестанского федерального исследовательского центра РАН

E-mail: hasan.akniyev@gmail.com

Поступила в редакцию

14.08.19

Р. М. Гаджимирзаев (R. M. Gadzhimirzaev)

Отдел математики и информатики Дагестанского федерального исследовательского центра РАН

E-mail: ramis3004@gmail.com