УДК 518.3 / 681.142.2

ИНТЕГРО-ДИФФЕРЕНЦИАЛЬНЫЙ АЛГОРИТМ А-МЕТОДА В.К.ДЗЯДЫКА И ЕГО РЕАЛИЗАЦИЯ СРЕДСТВАМИ АЛГЕБРАИЧЕСКОГО ПРОГРАММИРОВАНИЯ

П.Н.Денисенко

Побудовано аналітичний наближений метод розв'язування задачі Коші для лінійних диференціальних рівнянь порядку k з многочленними коефіцієнтами -- ЛДУМК. Вихід методу – многочлен. Цей многочлен оптимально апроксимує розв'язок задачі Коши для ЛДУМК і тотожний многочлену який одержують за а-методом В.К.Дзядика. Побудований метод не має найбільш складного перетворення а-методу В.К.Дзядика – перетворення задачі Коші в інтегральне рівняння. Засобами системи алгебраїчного програмування APS метод перетворено в обчислювальну процедуру на мові програмування APLAN.

Analytic approximation method of solving initial-value problem for linear differential equations with polynomial coefficients (LDEPC) of order k is presented. Output of the method is a polynomial. This polynomial optimally approximates the solution of initial-value problem for LDEPC and is identically equal to the polynomial computed by V.K. Dzyadyk a-method. The method presented does not contain the most complex transformation of V.K. Dzyadyk a-method - transformation of initial-value problem into the integral equation. The method is implemented by means of algebraic programming system in a computation procedure in APLAN programming language.

1 Проблема

Линейные дифференциальные уравнения порядка k

$$LDUMK := (D[y] = 0) ; \; D[y] = A * y^{(k)} + . . . + C * y + G ; \qquad (1)$$

где коэффициенты $A , . . . , C , G$ являются известными многочленами независимой переменной x , y – неизвестная (искомая) функция переменной x (ЛДУМК), являются классическим аппаратом математического моделирования. ЛДУМК определяют основную часть элементарных функций и специальных функций математической физики (СМФ). Решение уравнений этого типа в виде конечной композиции элементарных функций или в квадратурах удается найти только для очень узкого подкласса ЛДУМК. Поэтому в связи с потребностями практики постоянно разрабатывались методы приближенного решения ЛДУМК (1). Одним из старейших методов решения уравнений этого типа является метод разложения в ряд Тейлора . Согласно этому методу решение задачи Коши для уравнения (1) – системы из уравнения (1) и условий в начальной точке d

$$init\_cond := \{ y(d) = Y_0 , \; y'(d) = Y_1 , . . . , \; y^{(k-1)}(d) = Y_{k-1} \} ; \qquad (2)$$

в окрестности точки d приближается отрезком ряда Тейлора порядка n

$$T[n , y , d , x] = c_0 + c_1 * (x – d) + . . . + c_n * (x – d)^n$$

Погрешность $y(x) - T[n, y, d, x]$ метода разложения в ряд Тейлора пропорциональна $(x - d)^{n+1}$. При больших значениях $x - d$ эта погрешность медленно стремиться к нулю при $n \to \infty$, а если значение $x - d$ больше радиуса сходимости ряда Тейлора, то эта погрешность вообще не стремиться к нулю при $n \to \infty$. Погрешность метода обычно оценивают в пространстве $C[a,b]$ -- пространстве Банаха функций непрерывных на отрезке аппроксимации $[a, b]$ с равномерной (Чебышевской) нормой

$$|| y ||_{C[a,b]} = max \{ \ | y(x) | , x \in [a , b] \}$$

Коэффициент оптимальности решения задачи Коши для обыкновенных дифференциальных уравнений методом разложения в ряд Тейлора в пространстве $C[a, b]$ с ростом n растет со скоростью геометрической прогрессии

$$Opt(\text{ метод разложения в ряд Тейлора , задача Коши }, [a,b] , n , C[a,b]) =$$
$$|| y - T[n , y , d , x] ||_{C[a,b]} / E[n , y , C[a,b]] = O(q^n) ,$$

где $q > 2$,

$$E[n , y , C[a,b]] := inf_{Ci} || y - (C_0 + C_1 * x + . . . + C_n * x^n) ||_{C[a,b]}$$

– величина наилучшего приближения функции y многочленами степени n в пространстве $C[a, b]$. Эти обстоятельства, а также необходимость вычисления большого числа частных производных, резко ограничивают область применения метода разложения в ряд Тейлора.

Аналитические приближенные методы решения задачи Коши для обыкновенных дифференциальных уравнений (ОДУ) – методы вычисления алгебраических многочленов

$$y_n = c_0 + c_1 * x + . . . + c_n * x^n \qquad (3)$$

приближающих решение y этой задачи -- разработаны для отдельных типов ОДУ и для отдельных уравнений. Для оценки эффективности этих методов наиболее часто используют критерий -- точность аппроксимации в пространстве $C[a,b]$.

Если метод имеет ограниченный коэффициент оптимальности

$$Opt(\text{ метод , задача Коши для ОДУ }, [a,b] , n , C[a,b]) < Const ,$$

то он оптимален по точности аппроксимации в пространстве $C[a,b]$.

Наибольше аналитических приближенных методов построено для решения задачи Коши для ЛДУМК (1), (2). Это тау-метод Ланцоша, а-метод В.К.Дзядыка [1], метод Кленшоу , метод Миллера и другие. Оптимальным по точности методом решения задачи Коши для ЛДУМК является а-метод В.К.Дзядыка . В а-методе выполняют следующие преобразования:

– смещение ЛДУМК (1) и начальных условий (2) к новой начальной точке

$$d := 0; \qquad (4)$$

– преобразование смещенной задачи Коши для ЛДУМК (1) , (2), (4) в линейное интегральное уравнение Вольтерра с коэффициентами и ядром являющимися многочленами (ЛИУМК) эквивалентное задаче Коши

$$LIUMK := (F[y] = 0) := Int\_equation(LDUMK , init\_cond) ; \qquad (5)$$

$$F[\,y\,] = F[\,y\,,x\,] := A * y + V[\,K * y\,] + G\,; \quad V[\,u\,] := \int_0^x u(t)\,dt\,; \tag{6}$$

– решение ЛИУМК (5) на отрезке $[a - d, b - d]$ а-методом В.К.Дзядыка и вычисление аппроксимирующего многочлена вида (3)

$$y_n = solve\_a\text{-}method(\,LIUMK\,, [a - d, b - d]\,, n\,)\,; \tag{7}$$

– линейный перенос многочлена y_n (7) к начальной точке условий (2). Метод (7) решения ЛИУМК -- *solve_a-method* -- предполагает аппроксимацию уравнения (5) – добавление в это уравнение невязки с тау-коэффициентами

$$F[\,y_n\,] + E_m = 0\,; \tag{8}$$

$$m := deg(\,F[\,y_n\,]\,)\,; \tag{9}$$

$$E_m := tau_1 * f(\,n + 1\,, x\,) + \ldots + tau_{m-n} * f(\,m\,, x\,)\,; \tag{10}$$

и базисом $\{\ f(\,i\,, x\,),\ \ i = 0, 1, \ldots\ \}$ -- многочлены Чебышева первого рода линейно перенесенные на отрезок аппроксимации $[a\,, b]$

$$f(\,i\,, x\,) := cos(\,i * arcccos(\,z\,)\,)\,;\ \ z := 2 * (x - a) / (b - a) - 1\,; \tag{11}$$

Преобразование задачи Коши для ЛДУМК (1) , (2), (4) в ЛИУМК (5) является наиболее сложным в а-методе. Формулы этого преобразования полученны В.К.Дзядыком [1] и не реализованы программно в общем виде. Тау-метод Ланцоша, метод Кленшоу, метод Миллера и а-метод В.К.Дзядыка широко применяются для вычисления коэффициентов Фурье-Чебышева специальных функций математической физики (СМФ). Поэтому актуальна следующая задача.

Задача 1. Построить оптимальной по точности аналитический приближенный метод решения задачи Коши для ЛДУМК (1), (2) на отрезке. Метод не имеет преобразования задачи Коши для ЛДУМК (1), (2) в ЛИУМК.

2 Компьютерный аспект проблемы

Системы компьютерной алгебры стали средой математического моделирования. Они широко применяются в практике моделирования. Системы компьютерной алгебры имеют эффективные средства для программирования символьного преобразования многочленов вида (3). Если многочлен размещается в оперативной памяти компьютера, то, как правило, система компьютерной алгебры достаточно эффективно выполняет все известные алгебраические, дифференциальные и интегральные преобразования этого многочлена. Поэтому в системе компьютерной алгебры:

– предпочтительным объектом символьного преобразования является многочлен степени n вида (3) ;

– предпочтительным приближенным методом решения обыкновенных дифференциальных уравнений является аналитический метод: согласно этому методу система вычисляет многочлен вида (3) и этот многочлен оптимально по точности аппроксимирует решение ОДУ на отрезке $[\,a\,, b\,]$.

Средствами алгебраического программирования а-метод В.К.Дзядыка реализован [2] в виде APLAN-процедуры. В системе алгебраического программирования APS эта процедура решает задачу Коши для ЛДУМК общего вида (1), (2) с точкой задания начальных условий вида (4). Эта процедура имеет оператор *Int_equation* . Он выполняет наиболее сложное преобразование а-метода В.К.Дзядыка [1] – задачи Коши (1), (2), (4) в ЛИУМК (5) и требует оперативной памяти больше, чем остальная процедура. Поэтому программы для решения задачи Коши для ОДУ, построенные на основе этого оператора и итерационных методов согласно идеи В.К.Дзядыка, требуют память, как правило, большую чем имеет система APS.

Решение задачи Коши для ЛДУМК (1), (2) процедурой, реализующей дифференциальный алгоритм а-метода В.К.Дзядыка [3], тождественно а-методу [1] на очень важных для практики классах ЛДУМК (1)

$$k = 1 \; ; \quad \text{и} \quad k = 2 \; ; A = Const \; .$$

Но эти классы ЛДУМК не содержат всех ЛДУМК применяемых на практике .

Исходя из изложенного выше, для систем компьютерной алгебры актуальна следующая задача.

Задача 2. В системе компьютерной алгебры построить процедуру для решения задачи Коши для ЛДУМК (1), (2) на отрезке *[a, b]*. Процедура вычисляет многочлен вида (3). Этот многочлен аппроксимирует решение исходной задачи Коши на отрезке *[a , b]* оптимально по точности. Процедура не имеет преобразования задачи Коши для ЛДУМК (1), (2) в ЛИУМК (5).

3 Интегро-дифференциальный алгоритм а-метода

Идея. Вычислить аппроксимирующее уравнение (8) а-метода В.К.Дзядыка (5) – (11) в результате непосредственного преобразования задачи Коши для ЛДУМК (1), (2), (4).

Метод 1:

– вычислить аппроксимацию ЛДУМК (1)

$$SD := (coefTayl(D[y_n] + E_m^{(k)} , i) = 0 \; , \; i := 0 \; , \; . \; . \; . \; , m - k \; ;) \; ; \tag{12}$$

где невязка E_m имеет вид (10) и параметр

$$m = k + q \; ; \quad q = deg(D[y_n]) \; ; \quad k = ord\_equ(D[y]) \; ; \tag{13}$$

– вычислить аппроксимацию начальных условий (2), (4) СЛАУ

$$SI := (D[j , y_n(d) , d] + E_m^{(j)}(d) = D[j , init\_cond , d] , \; j := 1 \; , \; . \; . \; . \; , k \; ;) \; ; \tag{14}$$

где оператор

$$D[j , y , x] := D[j , y] := DV[D[j - 1 , y]]; \tag{15}$$

это дифференциальная часть первообразной порядка j оператора $D[y]$ (1);

$D[j , y_n(d) , d]$ -- подстановка в оператор $D[j , y , x]$ многочлена y_n (3) с символьными коэффициентами и точки задания начальных условий (2) ;

$D[j , init\_cond , d]$ -- подстановка в оператор $D[j , y , x]$ условий (2).

Пример 1. Функция $ln(x + 1)$ является решением задачи Коши

$$(x + 1) * y' = 1 ; \quad y(0) = 0 ; \qquad (16)$$

Процедура *Int_equation* преобразует задачу Коши (16) в ЛИУМК (5) вида

$$(x + 1) * y - V[y] - x = 0 ; \qquad (17)$$

Аппроксимация ЛИУМК (17) a-методом В.К.Дзядыка (8) -- (10) имеет:

1) параметр (9) невязки (10)

$$m := deg(F[y_n]) := n + 1 ;$$

2) невязку (10) с одним тау-коэффициентом

$$E_m := tau * f(n + 1 , x) ;$$

3) уравнение (8) эквивалентное СЛАУ вида

$$\begin{aligned} S := \{ coefTayl(F[y_n] + E_m ,\ i) = 0 ,\ i := 0 , . . . , m ; \} := \\ \{ a(m , 0) * tau + c_0 = 0 , \\ a(m , 1) * tau + c_1 + -1 = 0 , \\ a(m , 2) * tau + c_2 + 1/2 * c_1 = 0 , \\ . . . \\ a(m , i) * tau + c_i + (i - 1) / i * c_{i-1} = 0 , \\ . . . \\ a(m , m) * tau + n/m * c_n = 0 \} ; \end{aligned} \qquad (18)$$

где $m := n + 1 ;$

$a(m , i)$ -- коэффициенты многочлена базиса невязки $f(m , x)$.

Аппроксимация задачи Коши для ЛДУМК (16) методом (12) – (14) имеет:

1) параметр m (13) невязки (10) тождественный параметру невязки a-метода

$$k := ord\_equ(D[y]) := 1 ; \quad m := k + deg(D[y_n]) := n + 1 ;$$

2) невязку с одним тау-коэффициентом – вида тождественного виду невязки аппроксимации ЛИУМК (17) a-методом ;

3) СЛАУ (12) вида

$$\begin{aligned} SD :=\{ coefTayl(D[y_n] + E_m' , i) = 0 ,\ i := 0 , . . . , n ; \} := \\ \{ a(m , 1) * tau + c_1 + -1 = 0 , \\ 2 * a(m , 2) * tau + 2 * c_2 + c_1 = 0 , \\ . . . \\ i * a(m , i) * tau + i * c_i + (i - 1) * c_{i-1} = 0 , \\ . . . \\ m * a(m , m) * tau + n * c_n = 0 \}; \end{aligned}$$

и эта СЛАУ тождественна части $(i = 1 , . . . , m ;)$ СЛАУ S *(18);*

4) СЛАУ (14) вида

$$SI := (y_n(0) + E_m(0) = 0) := (c_0 + a(m , 0) * tau = 0) ;$$

и эта СЛАУ тождественна первому уравнению СЛАУ (18) $SI = S , (i = 0);$

4 Обоснование интегро-дифференциального алгоритма а-метода

Теорема 1. *Пусть:*
– точка задания начальных условий (2) имеет вид (4) -- $d = 0$ *,*
– ЛДУМК (1) не имеет особенности вида $A(d) = 0$*.*

Тогда тождественны: СЛАУ а-метода В.К.Дзядыка

$$S := (\ coefTayl(\ F[\ y_n\ ] + E_m\ ,\ i\ ) = 0\ ,\quad i := 0\ ,\ ...\ ,\ m\ ;\ )\ ; \tag{19}$$

и объединение СЛАУ (12) и (14) метода 1 -- $conc(\ SI\ ,\ SD\ )$*.*

Доказательство. Согласно определения ЛИУМК (5), для оператора этого ЛИУМК и оператора ЛДУМК (1) имеет место тождество

$$(\ F[\ y\ ]\ )^{(k)} = D[\ y\ ]\ ; \tag{20}$$

Поэтому теорема 1 является непосредственным следствием лемм 1 – 5. Леммы доказаны ниже.

Лемма 1. *Пусть:*
– порядок а-метода В.К.Дзядыка (4) -- (11) и интегро-дифференциального алгоритма а-метода (12) – (14) $n > k - 1$ *,*
– для операторов $D[\ y\ ]$ *ЛДУМК (1) и* $F[\ y\ ]$ *(6) ЛИУМК (5) имеет место тождество (20).*

Тогда вид невязки E_m *(10) для уравнения (8) тождественен виду этой невязки для СЛАУ (12), (14) .*

Доказательство. Согласно правил дифференцирования, для многочленов P_m степени $m > k - 1$ имеет место тождество

$$deg(\ P_m^{(k)}\ ) = deg(\ P_m\ ) - k\ ;$$

Согласно этого тождества и тождества (20) имеет место тождество

$$deg(\ F[\ y_n\ ]\ ) = deg(\ D[\ y_n\ ]\ ) + k\ , \tag{21}$$

где многочлены $D[\ y_n\ ]$ и $F[\ y_n\ ]$ являются результатами преобразований операторами $D[\ y\ ]$ (2) и $F[\ y\ ]$ (6) многочлена y_n вида (3) с символьными коэффициентами. Тождество (21) является тождеством значений параметра m (9) невязки (10) уравнения (8) и параметра m (13) невязки (10) СЛАУ (12) и (14). Параметры n , m однозначно определяют вид невязки (10).

Лемма 2. *Пусть для оператора* $D[\ y\ ]$ *ЛДУМК (1) и оператора* $F[\ y\ ]$ (6) *ЛИУМК (5) имеет место тождество (20) .*

Тогда СЛАУ (12) тождественна уравнениям $i := k\ ,\ ...\ ,\ m\ ;$ *СЛАУ* (19)

$$SD = S\ (\ i := k\ ,\ ...\ ,\ m\ ;\ )\ ;$$

Доказательство. Согласно правил дифференцирования, для многочленов P_m степени $m > i$ и $j < i$ имеет место тождество

$$coefTayl(\ P_m\ ,\ i\ ) * i! := coefTayl(\ P_m^{(j)}\ ,\ i - j\ ) * (i - j)!$$

Согласно этого тождества и тождества (20), если $i > k$, то имеет место тождество

$$coefTayl(\ F[\ y_n\ ]\ ,\ i\ ) * i! := coefTayl(\ D[\ y_n\ ]\ ,\ i - k\ ) * (i - k)!$$

где многочлены $D[\ y_n\ ]$ и $F[\ y_n\ ]$ являются результатами преобразований операторами $D[\ y\ ]$ (1) и $F[\ y\ ]$ (6) многочлена y_n вида (3) с символьными коэффициентами. Согласно лемме 1, вид невязки уравнения (8) и СЛАУ (12) тождественен. Поэтому, если $i > k$, то имеет место тождество

$$coefTayl(\, E_m \, , \, i \,) * i! := coefTayl(\, E_m \, , \, i - k \,) * (i - k)!$$

Из этих тождеств непосредственно следует тождество СЛАУ (12) и части $(\, i = k \, , \, . \, . \, . \, , \, m \, ; \,)$ СЛАУ (19).□

Лемма 3. *Пусть:*

– для оператора $D[\, y \,]$ *(1) и оператора* $F[\, y \,]$ *(6) имеет место тождество (20),*

– операторы $D[\, j \, , \, y \, , \, x \,]$ *(15) удовлетворяют начальным условиям (2), (4).*

Тогда имеют место тождества

$$(\, F[\, y \, , \, x = 0 \,] \,)^{(k - i)} = D[\, i \, , \, y \, , \, x = 0 \,] - D[\, i \, , \, init\_cond \,] \, , \;\; i = 1 \, , \, . \, . \, . \, , \, k \qquad (22)$$

Доказательство. Значение линейного интегрального оператора Вольтерра $V[\, u \,]$ (6) в точке ноль, очевидно, равно нулю. Поэтому оператор $F[\, y \,]$ (6) в точке ноль имеет значение

$$F[\, y \, , \, x = 0 \,] = A(0) * y(0) + G(0) \, ; \quad G(0) = - A(0) * Y_0 \, ;$$

Согласно тождества (20) оператор $D[\, k \, , \, y \, , \, x \,]$ имеет вид

$$D[\, k \, , \, y \, , \, x \,] = A * y$$

Поэтому справедливы тождества

$$D[\, k \, , \, y \, , \, x \,] = A(0) * y(0) \, , \quad D[\, k \, , \, init\_cond \,] = A(0) * Y_0$$

и, следовательно, тождество (22) имеет место в случае $i = k$. Тождество (22) для производных оператора $F[\, y \,]$ доказывается аналогично. □

Лемма 4. *Операторы (15) имеют вид*

$$D[\, i \, , \, y \,] = A * y^{(k-i)} + . \, . \, . + E * y \, ; \quad i = 1 \, , \, . \, . \, . \, , \, k \, ; \qquad (23)$$

Доказательство. Согласно определения (15) и лемме 3, операторы $D[\, i \, , \, y \,]$, $i = 1 \, , \, . \, . \, . \, , \, k$, являются дифференциальным слагаемым производной порядка $k - i$ оператора $F[\, y \,]$ (6) . Согласно определения оператора $F[\, y \,]$ (6), дифференциальное слагаемое его производной порядка $k - i$ имеет вид (23). □

Лемма 5. *Пусть ЛДУМК (1) не имеет особенности вида* $A(d) = 0$ *в точке задания начальных условий (2) вида (4)* $d = 0$.

Тогда тождественны СЛАУ (14) и первые k *уравнений СЛАУ (19)*

$$SD := S \, (\, i := 0 \, , \, . \, . \, . \, , \, k - 1 \, ; \,) \, ;$$

Доказательство. Доказываемое тождество является непосредственным следствием тождеств (22), (23) и тождества вида невязки СЛАУ (14) и (19). Последнее тождество доказано в лемме 1. □

5 Реализация метода (12) – (14) в APS

Вход: – ЛДУМК (1) ,

– начальные условия (2) с начальной точкой вида (4) ,

– отрезок аппроксимации $[a, b]$,

– порядок искомого аппроксимирующего многочлена вида (3).

Выход.

Многочлен вида (3) – решение задачи Коши (1), (2), (4) методом (12) – (14).

Алгоритм 1. Операции преобразования -- вычислить:

1. Объединение СЛАУ (12) и (14)

$$S := conc(SI\ ,\ SD\ )\ ; \qquad (24)$$

2. Решение СЛАУ (24) -- коэффициенты многочленов y_n , E_m

$$Coef := solve(\ S\ )\ ; \qquad (25)$$

3. Многочлен y_n вида (3) с коэффициентами $c_0, \ldots, c_n$ из списка (25)

$$y_n := ser(\ n\ ,\ Coef\ )\ ; \qquad (26)$$

Решение задачи Коши (16) по алгоритму 1 с базисом невязки тождественным базису ряда Тейлора $\{ x^i \}$, $n := 2$; $E_m := tau * x^3$;

1. СЛАУ (14) аппроксимирующая начальное условие $\{ y = 0 \}$

$$SI := \{ D[\ 1\ ,\ y_n(d)\ ,\ d\ ] + E_m(d) = D[\ 1\ ,\ init\_cond\ ,\ d\ ]\ \} := \{ c_0 = 0 \}\ ;$$

СЛАУ (12) аппроксимирующая ЛДУМК $\{ (x + 1) * y' - 1 = 0 \}$

$$SD := \{ coefTayl(D[\ y_n\ ] + E_m^{(k)}\ ,\ i\ ) = 0\ ,\ i := 0\ , \ldots ,\ m - k\ ;\ \} :=$$

$$\{ \quad c_1 - 1 = 0\ ,$$
$$2 * c_2 + c_1 = 0\ ,$$
$$3 * tau + 2 * c_2 = 0\ \}\ ;$$

2. Решение объединения (24) этих СЛАУ

$$Coef := solve(\ S\ ) := \{ tau = 1/3\ ,\ c_0 = 0\ ,\ c_1 = 1\ ,\ c_2 = 1/2\ \};$$

3. $y_n := ser(\ n\ ,\ Coef\ ) := x - 1/2 * x^2$;

Решение задачи Коши (16) по алгоритму 1 с базисом невязки (11)

$n := 2$; $interval := (0\ ,\ 1)$; $E_m := tau * cheb(n+1\ ,\ 2x - 1)$;

1. СЛАУ (14) аппроксимирующая условие $init\_cond := (\ y(0) = 0\ )$;

$$SI := \{ D[1\ ,\ y_n(d)\ ,\ d\ ] + E_m(d) = D[1\ ,\ init\_cond\ ,\ d\ ]\ \} := \{ -\ tau + c_0 = 0\ \}\ ;$$

СЛАУ (12) аппроксимирующая ЛДУМК $\{ (x + 1) * y' - 1 = 0 \}$

$$SD := \{ coefTayl(\ D[\ y_n\ ] + E_m^{(k)}\ ,\ i\ ) = 0\ ,\ i := 0\ , \ldots ,\ m - k\ ;\ \} :=$$

$$\{ 18 * tau + \quad c_1 - 1 = 0\ ,$$
$$-96 * tau + 2 * c_2 + c_1 = 0\ ,$$
$$96 * tau + 2 * c_2 = 0\ \}\ ;$$

2. Решение объединения (24) этих СЛАУ

$$Coef := solve(\ S\ ) := \{ tau = 1/210\ ,\ c_2 = -8/35\ ,\ c_1 = 32/35\ ,\ c_0 = 1/210\ \}\ ;$$

3. $y_n := ser(\ n\ ,\ Coef\ ) := -8/35 * x^2 + 32/35 * x + 1/210$;

Структура данных. Структура данных построенной процедуры тождественна структуре данных процедур [2] и [3].

1. Слагаемые ЛДУМК (1) упорядочены по порядку производных и ЛДУМК имеет вид принятый в системе Maple III

```
LDUMK := ( A * dif(y , k) + ...+ C * y + g = 0 );
```

где y -- атом, коэффициенты $A, \ldots, B, C, g$ -- многочлены переменной (атома) x естественного вида .

2. Начальные условия (2), (4) задаются в виде принятом в системе Maple III. Начальная точка (4) определяется оператором присвоения

```
InitPoint := 0 ;
```

и значения производных в этой точке определяются в виде списка

```
init_cond :=
( y = Yo , dif(y,1) = Y1 ,..., dif(y,s) = Ys );
```

где y - атом, $s := k - 1$;

3. Описание отрезка аппроксимации $[\ a\ ,\ b\ ]$ имеет вид

```
interval := (a , b);
```

4. Процедура вычисляет объединение (24) СЛАУ (12), (14). Неизвестными СЛАУ (24) являются коэффициенты многочленов y_n (3) и E_m (10). Для коэффициентов невязки принято обозначение $tau_i = c_{n+i}$. Поэтому неизвестные СЛАУ (24) имеют вид -- атом c с индексом

```
c(0),...,c(n),...,c(m)
```

Для вычисления СЛАУ (24) процедура преобразует многочлены y_n , E_m с символьными коэффициентами этого вида.

5. Результат (25) решения СЛАУ (24) процедура вычисляет в виде списка тождеств

```
c(0)= d ,...,c(n) = e ,...,c(m) = f
```

6. Многочлен (26) процедура вычисляет в естественном виде.

Алгебраическая спецификация алгоритма 1.

```
S := aprox_Dzyadyk( LDUMK , InitPoint , init_cond ,
interval , n); /* аппроксимация задачи Коши СЛАУ */
Xn := c;                      /* имя неизвестных в СЛАУ */
Coef := solve(S);                      /* решение СЛАУ */
y_n := ser(n , Coef);  /* y_n с коэффициентами Coef */
```

Выводы из алгебраической спецификации алгоритма 1.

1. Вид спецификации тождественен виду спецификации процедуры [2] для тау-метода Ланцоша.
2. Спецификация процедуры имеет один новый, относительно спецификации процедуры тау-метода Ланцоша [2], оператор -- *aprox_Dzyadyk* .
3. Операторы процедуры (согласно их спецификации):
   – выполняют вычисления в арифметике рациональных чисел;
   – имеют по параметру n полиномиальную сложность

$$Q(\,solve\_ode\,,\,n\,) = O(\,n * Q(\,canplf\,,\,n^2\,)\,) + O(\,n^3\,)\ ;$$

$$Q(\,solve\,,\,n\,) = O(\,n^3\,)\ ;$$

$$Q(\,ser\,,\,n\,) = O(\,n\,)\ ;$$

где $Q(\ canplf\ ,\ m\ )$ -- сложность преобразования оператором *canplf* многочлена P к виду каноническому для системы APS – сумма слагаемых вида **c(i) * x^j $ b** , m -- число слагаемых многочлена *canplf(P)*.

4. Процедура имеет по параметру n полиномиальную сложность

$$O(\,n * Q(\,canplf\,,\,n^2\,)\,) + O(\,n^3\,)\ ;$$

APLAN-программа.

1. Файлы выполнения программы:

D_3_1.BAT – для программы с базисом невязки – базисом ряда Тейлора,
D_3_2.BAT – для программы с базисом невязки вида (11).

2. Файлы с оператором *aprox_Dzyadyk* -- информатива программы:
D_3_1.AP -- базисом невязки является базис ряда Тейлора;
D_3_2.AP -- базисом невязки являются многочлены Чебышева (11).

3. Файл с операторами *process* и *task* -- TASK_d_1.AP -- директива программы. Эта директива тождественна директиве программы реализующей тау-метод Ланцоша [2].

Решение задачи Коши (16) построенной процедурой.

APLAN-описание задачи Коши (16) и отрезка аппроксимации *[0,1]* имеет вид

```
process[1]:=(
    LDUMK     := ( (x + 1) * dif(y , 1) + (-1) = 0 ) ;
    InitPoint := 0 ;
    init_cond := ( y = 0 ) ;
    interval  := (0 , 1);  ... );
```

Результаты преобразования этой задачи Коши построенной выше APLAN-процедурой при *n := 2 ;* и базисе невязки тождественном базису ряда Тейлора имеют вид:

```
S := aprox_Dzyadyk ( LDUMK , InitPoint , init_cond ,
        interval , n ) :=
     ( c 0 = 0 , c 1 + -1 = 0 , c 2 $ 2 + c 1 = 0 );
Xn := c; Coef := solve(S) :=
     ( c 0  = 0 , c 1  = 1 , c 2  = rat(-1 , 2) );
y_n := ser(n , Coef):= x + -1/2 * x ^ 2 ;
```

Результаты преобразования этой задачи Коши построенной выше APLAN-процедурой при *n := 2 ;* и базисе невязки (11) имеют вид:

```
S := aprox_Dzyadyk ( LDUMK , InitPoint , init_cond ,
               interval , n ) :=
          (  c 3  $ -1                 + c 0  = 0 ,
             c 3  $ 18 +            c 1 +  -1 = 0 ,
             c 3  $-96 + c 2  $ 2 + c 1       = 0 ,
             c 3  $ 96 + c 2  $ 2             = 0 ) ;
Xn := c ; Coef := solve(S) :=
          ( c 0  = rat(1,210) , c 1  = rat(32,35) ,
            c 2  = rat(-8,35) , c 3  = rat(1,210) );
y_n := ser(n , Coef):=
     x ^ 2 $ rat(-8,35) + x $ rat(32,35) + rat(1,210) ;
```

В приведенных выше результатах:

$ -- операция языка APLAN умножения терма на число,

rat(p , q) := p / q ; -- рациональное число,

c 0 , c 1 , c 2 , c 3 -- результат вывода атома с индексом `c(0) , c(1) , c(2) , c(3)` оператором *prn* системы APS.

6 Оператор *aprox_Dzyadyk*

Вход: – ЛДУМК (1) ,
– условия (2) в начальной точке вида (4) ,
– отрезок аппроксимации $[a, b]$,
– порядок искомого многочлена вида (4).
Выход – СЛАУ (24).
Метод.

1. Многочлен y_n (3) с символьными коэффициентами подставляют в оператор $D[\,y\,]$ ЛДУМК (1) и вычисляют многочлен
$D[\,y_n\,] := D_0 + \ldots + D_q * x^q$; $D_i := a(i\,,\,0) * c_0 + \ldots + a(i\,,\,n) * c_n + b_i$;
2. Вычисляют параметры $q\,,\,k\,,\,m$ (13) СЛАУ (12) и невязку E_m (10) с символьными коэффициетами.
3. Вычисляют СЛАУ (12).
4. Вычисляют дифференциальную часть первообразных $D[\,i\,,\,y\,]$ порядка $i := 1\,,\,\ldots\,,\,k$; оператора $D[\,y\,]$ (1) и СЛАУ (14).

Алгоритм 1. Операции преобразования -- вычислить:

1. Оператор $D[\,y\,]$ ЛДУМК (1).
2. Порядок дифференциального оператора $D[\,y\,]$.
3. Многочлен y_n (3) с символьными коэффициентами.
4. Преобразование $D[\,y_n\,]$ многочлена y_n оператором $D[\,y\,]$.
5. Порядок многочлена $D[\,y_n\,]$ и параметр m невязки.
6. Невязку E_m (10) с символьными коэффициентами для отрезка $[\,-1\,,\,1\,]$.
7. Линейное преобразование $z : [\,a\,,\,b\,]$ --> $[\,-1\,,\,1\,]$.
8. Невязку E_m (10) с символьными коэффициентами для отрезка $[\,a\,,\,b\,]$.
9. Многочлен $D[\,y_n\,] + E_m^{(k)}$.
10. Систему линейных алгебраических уравнений (12).
11. Систему линейных алгебраических уравнений (14) --
для $i := 1\,,\,\ldots\,,\,k$; интегрировать оператор $D[\,y\,]$ и вычислить:

11.1. $D[\,i\,,\,y\,,\,x\,]$ --
дифференциальный оператор i-той первообразной оператора $D[\,y\,]$.
11.2. $D[\,i\,,\,y\,,\,d\,]$ -- значение оператора $D[\,i\,,\,y\,,\,x\,]$ в точке $d := 0$; (4)
-- дифференциальный оператор с постоянными коэффициентами.
11.3. $D[\,i\,,\,y_n(x)\,,\,d\,] := D_n(x)$; -- преобразование многочлена y_n (3) с символьными коэффициентами оператором $D[\,i\,,\,y\,,\,d\,]$.
11.4. $D_i(x) := D[\,i\,,\,y_n(x)\,,\,d\,] + E_m^{(k-i)})$; -- многочлен.
11.5. $D_i(d)$ -- значение многочлена $D_i(x)$ в точке d .
11.6. $D[\,i\,,\,init\_cond\,,\,d\,]$ -- подстановка в оператор $D[\,i\,,\,y\,,\,d\,]$ условий (2), (4) -- число.
11.7. Уравнение СЛАУ (14) -- аппроксимацию $k - i$ -- того условия (2).

12. Объединение СЛАУ (14) и СЛАУ (12).

Преобразование задачи Коши (16) по алгоритму 2 с базисом невязки (11), interval := (0 , 1) ; n := 2 ;

1. $D[y] := (x + 1) * y' + 1 ;$
2. $k := 1 ;$
3. $y_n := c_2 * x^2 + c_1 * x + c_0 ;$
4. $D[y_n] := (x + 1) * (2 * c_2 * x + c_1);$
5. $q := deg(D[y_n]) := 2 ; \quad m := q + k := 3 ; \quad m - n := 1 ;$
6. $E_m := tau * cheb(3 , x) := tau * (4 * x^3 - 3 * x) ;$
7. $z := 2 * x - 1 ;$
8. $Em(z) := tau * cheb(3 , 2 * x - 1) := tau * (32 * x^3 - 48 * x^2 + 18 * x - 1) ;$
9. $D[y_n] + E_m^{(k)} := (x + 1) * (2 * c_2 * x + c_1) + tau * (96 * x^2 - 96 * x + 18) ;$
10. $SD := \{ coefTayl(D[y_n] + E_m^{(k)} , i) = 0 , i := 0 , . . . , m - k ; \}$

$$
\begin{aligned}
:= \{ & 18 * tau + c_1 - 1 = 0 , \\
& -96 * tau + 2 * c_2 + c_1 = 0 , \\
& 96 * tau + 2 * c_2 = 0 \} ;
\end{aligned}
$$

11.1. $D[1 , y , x] := (x + 1) * y ;$
11.2. $D[1 , y , 0] := y ;$
11.3. $D[1 , y_n(x) , 0] := c_2 * x^2 + c_1 * x + c_0 ;$
11.4. $D[1 , y_n , 0] + E_m :=$
$c_2 * x^2 + c_1 * x + c_0 + tau * (32 * x^3 - 48 * x^2 + 18 * x - 1) ;$
11.5. $D[1 , y_n(0) , 0] + E_m(0) := - tau + c_0 ;$
11.6. $D[1 , init\_cond , 0] := 0 ;$
11.7. $\{ D[1 , y_n(0) , 0] + E_m(0) = D[1 , init\_cond , 0] \} := \{ -tau + c_0 = 0 \} ;$
12. $conc(SI , SD) :=$

$$
\begin{aligned}
\{ & -1 * tau + c_0 = 0 , \\
& 18 * tau + c_1 - 1 = 0 , \\
& -96 * tau + 2 * c_2 + c_1 = 0 , \\
& 96 * tau + 2 * c_2 = 0 \} ;
\end{aligned}
$$

Структура данных. Структуру данных на входе и выходе построенного оператора определяет основная процедура. Для вычисления СЛАУ (24) оператор преобразует многочлены y_n , E_m с символьными коэффициентами . Аппроксимирующий многочлен (3) оператор вычисляет в виде

```
y_n := c(0) + c(1) * x + ... + c(n) * x ^ n ;
```

Невязку (10) с базисом (11) оператор вычисляет на отрезке *[-1,1]* в виде

```
E_m :=
c(n + 1) * cheb(n + 1 , x) + ... + c(m) * cheb(m , x);
```

и линейно переносит на отрезок аппроксимации *[a , b]* -- заменой переменных

*subs(x = z , E_m) , z := 2 * (x - a) / (b - a) - 1 ;*

Алгебраическая спецификация алгоритма 2.

```
   aprox_Dzyadyk := proc( LDUMK , InitPoint , init_cond
, interval , n )loc(y_n,k,SD,S,Dy,Dn,Di,D0,Ri,E_m,m)(
   let( LDUMK , Dy = 0 );          /* оператор D[y]  */
   k := ord_equ(Dy);              /*  порядок ЛДУМК  */
   y_n := main_pol(n);        /*  y_n с коэфф. c(i)  */
 /*       АППРОКСИМАЦИЯ ЛДУМК                        */
   Dn :=  canplf(sub_du(Dy,y_n));         /*  D[y_n]  */
   m := deg(canplf(ein_pol(Dn))); /* порядок  D[y_n] */
   E_m :=  Enl(n,m+k-n);                  /*  E_m(x)  */
   S   := canplf( (2/(arg(interval,2) + (-1) * /* z(x)*/
arg(interval,1))*(x + (-1) * arg(interval,1) ) + (-1);
E_m --> canplf( subs( x = S, E_m ));      /*  E_m(z)  */
   Dn --> canplf(Dn + nd_x(E_m,k));  /* аппрок. D[y] */
   SD := pol_equ(Dn,m);       /*  СЛАУ аппр. Dy = 0  */
 /*       АППРОКСИМАЦИЯ   НАЧАЛЬНЫХ   УСЛОВИЙ        */
    for (i:=1, i<=k, i:=i+1,
        Dy --> intDy( Dy );               /*  D[i,y,x]  */
   D0 --> canplf(subs(x=InitPoint ,Dy));/*  D[i,y,d] */
   Dn --> sub_du(D0,y_n) ;              /* D[i,y_n,d] */
   Dn --> Dn + nd_x(E_m,k-i);/*D[i,y_n,d]+D^(k-i)E_m */
   Di --> canplf(subs(x = InitPoint , Dn)); /*  Dn(d)*/
   Ri --> canplf(subCondDy(D0,init_cond));/*D[i,cond]*/
   S --> copy(Di = Ri);/* аппроксимация начальн. усл.*/
   SD -->conc(S,SD)/*объединение с полученной СЛАУ */);
    return(SD)          /*  возврат СЛАУ  */   );
```

Выводы из алгебраической спецификации алгоритма 2.

1. Процедура *aprox_Dzyadyk* имеет только известные [2] APLAN-операторы.
2. Операторы процедуры *aprox_Dzyadyk* :
   -- выполняют вычисления в арифметике рациональных чисел ;
   -- имеют по параметру n полиномиальную сложность

$$Q(\, let(\, equation,\, Dy = 0\,)\,,\, n\,) = O(\,1\,)\,;$$
$$Q(\, k := ord\_equ(\, Dy\,)\,,\, n\,) = O(\,1\,)\,;$$
$$Q(\, main\_pol(\, n\,)\,,\, n\,) = O(\, n\,)\,;$$
$$Q(\, canplf(\, sub\_du(\, Dy\,,\, y\_n\,)\,)\,,\, n\,) = O(\, n^2\,) + O(\, Q(\, canplf\,,\, n^2\,)\,)\,;$$
$$Q(\, deg(\, canplf(\, ein\_pol(\, Dn\,)\,)\,)\,,\, n\,) = O(\, n^2\,) + O(\, Q(\, canplf\,,\, n^2\,)\,)\,;$$
$$Q(\, Enl(\, n\,,\, m + k - n\,)\,,\, n\,) = O(\, n^2\,)\,;\ \text{(для классического базиса)};$$
$$Q(\, canplf(\, subs(\, x = S\,,\, E\_m\,)\,)\,,\, n\,) = O(\, n^2\,) + O(\, Q(\, canplf\,,\, n^2\,)\,)\,;$$
$$Q(\, canplf(\, Dn + nd\_x(\, E\_m\,,\, k\,)\,)\,,\, n\,) = O(\, n^2\,) + O(\, Q(\, canplf\,,\, n^2\,)\,)\,;$$
$$Q(\, pol\_equ(\, Dn\,,\, m\,)\,,\, n\,) = n * O(\, Q(\, canplf\,,\, n^2\,)\,) + O(\, n^3\,)\,;$$
$$Q(\, intDy(\, Dy\,)\,,\, n\,) = O(\,1\,)\,;$$
$$Q(\, canplf(\, subs(\, x{=}InitPoint\,,Dy\,)\,)\,,\, n\,) = O(\,1\,)\,;$$

$$Q(\, sub\_du(D0\,,\, y\_n)\,,\, n\,) = O(\, n^2\,)\,;$$
$$Q(\, Dn + nd\_x(E\_m\,,\, k - i)\,,\, n\,) = O(\, n\,)\,;$$
$$Q(\, canplf(\, subs(\, x = InitPoint\,, Dn\,)\,)\,,\, n\,) = O(\,1\,)\,;$$
$$Q(\, canplf(\, subCondDy(\, D0\,,\, init\_cond\,)\,)\,,\, n\,) = O(\,1\,)\,;$$
$$Q(\, copy(\, Di = Ri\,)\,,\, n\,) = O(\,1\,)\,;$$
$$Q(\, conc(S\,,\, SD)\,,\, n\,) = O(\,1\,)\,;$$

3. Процедура *aprox_Dzyadyk :*

-- выполняет вычисления в арифметике рациональных чисел ;

-- имеет по параметру n полиномиальную сложность

$$Q(\, solve\_ode\,,\, n\,) = O(\, n * Q(\, canplf\,,\, n^2\,)\,) + O(\, n^3\,)\,;$$

Результаты преобразования оператором solve_ode задачи Коши (16) .
n := 2 ;

```
let( LDUMK , Dy = 0 );
        Dy := (x + 1) * dif(y , 1) + -1;
k := ord_equ(Dy) := 1;
y_n := main_pol(n) := c 2 * x^2 + c 1 * x + c 0;
Dn := canplf( sub_du(Dy, y_n) ) :=
c 2  * x ^ 2 $ 2 + c 1 * x + c 2* x $ 2 + c 1 + -1 ;
m := deg(canplf(ein_pol(Dn))) := 2;
E_m :=  Enl(n,m+k-n) := c 3  * ( x^3 $ 4 + x $ -3 );
S := canplf((2/(arg(interval,2)+(-
1)*arg(interval,1))
 * (x + (-1) * arg(interval,1) ) + (-1) := x $ 2 + -1;
E_m --> canplf( subs(x=S, E_m) ) := c 3 * x ^ 3 $ 32+
        c 3 * x ^ 2 $ -48 + c 3 * x $ 10 + c 3 $ -1 ;
 Dn --> canplf(Dn + nd_x(E_m,k)) :=
        c 3 * x^2 $ 96 + c 3  * x $ -96 + c 3  $ 18 +
c 2 * x ^ 2 $ 2 + c 1 * x + c 2 * x $ 2 + c 1 + -1 ;
SD := pol_equ(Dn,m) :=
         ( c 3  $ 18 +            c 1 +  -1 = 0 ,
           c 3  $-96 + c 2  $ 2 + c 1       = 0 ,
           c 3  $ 96 + c 2  $ 2             = 0 ) ;
 for( i := 1 )
      Dy --> intDy( Dy ) := (x + 1) * y ;
      D0 --> canplf(subs(x=InitPoint ,Dy)) := y ;
      Dn --> sub_du(D0,y_n) :=
             c 2 * x^2 + c 1 * x + c 0;
      Dn --> Dn + nd_x(E_m,k-i) :=
        c 2 * x^2 + c 1 * x + c 0 + c 3* x ^ 3 $ 32+
        c 3 * x ^2 $ -48 + c 3 * x $ 10 + c 3 $ -1 ;
      Di --> canplf(subs(x= InitPoint ,Dn)) :=
```

```
                         c 3 $ -1 + c 0 ;
        Ri --> canplf(subCond(D0,init_cond)) := 0;
        S --> copy(Di = Ri) := ( c 3 $ -1 + c 0 = 0 );
```

7 Заключение

Построенная в работе APLAN-процедура :

– вычисляет алгебраический многочлен тождественный решению задачи Коши для ЛДУМК a-методом В.К.Дзядыка – оптимальную по точности аппроксимацию решения задачи Коши для ЛДУМК;

– имеет естественный для математики вид;

– выполняет вычисления в арифметике рациональных чисел – не вносит дополнительные вычислительные погрешности;

– не выполняет наиболее сложное преобразование a-метода В.К.Дзядыка – преобразование задачи Коши в ЛИУМК и, поэтому, легко включается в процедуры решающие задачу Коши для ОДУ общего вида ;

– имеет по параметру n полиномиальную сложность

$$O(n * Q(canplf , n^2)) + O(n^3) ;$$

БИБЛИОГРАФИЯ

1. Дзядык В.К. Аппроксимационные методы решения дифференциальных и интегральных уравнений. – Киев: Наукова думка, 1988 – 387 с.
2. Денисенко П.Н., научная редакция Летичевский А.А. Алгебраическое программирование. Учебное пособие. – Кировоград: КННПК, 2002 – 120 с.
3. Денисенко П.Н. Модифікований метод Дзядика розв'язування задачі Коші. // Наукові записки Кіровоградського державного педагогічного університету, вип. 12 (1997) , с. 44-51.

Кировоградский Национальный технический университет

Надійшло 19 октября 2004