ПРОГРАММИРОВАНИЕ, 2013, № 3, с. 38–46

КОМПЬЮТЕРНАЯ АЛГЕБРА

УДК 512.715+519.61

СИМВОЛЬНО-ЧИСЛЕННОЕ РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ С ТРЕБУЕМОЙ ТОЧНОСТЬЮ

 Н.А. Малашонок, М.А. Рыбаков
Институт математики, физики и информатики
Тамбовского государственного университета им. Г.Р. Державина
392021 Тамбов, ул. Интернациональная, д. 33
E-mail: namalaschonok@gmail.com, mixail08101987@mail.ru
Поступила в редакцию 02.07.2012 г.

Приводится символьно-численный алгоритм получения решения системы обыкновенных линейных дифференциальных уравнений с постоянными коэффициентами и составной правой частью. Алгоритм основан на применении преобразования Лапласа. Частью алгоритма является определение погрешности вычислений, достаточной для требуемой точности решения системы. Алгоритм эффективен для решения систем дифференциальных уравнений большого размера, позволяет выбирать методы решения алгебраической системы – образа преобразования Лапласа – в зависимости от ее типа, а тем самым оптимизировать эффективность решения исходной системы. Алгоритм входит в состав библиотеки алгоритмов системы Mathpar [15].

1. ВВЕДЕНИЕ

Современные системы компьютерной алгебры предоставляют некоторые методы решения систем линейных дифференциальных уравнений – численных и символьно-численных. Нередкое обращение к преобразованию Лапласа в применении к решению дифференциальных уравнений в последние годы (см. например, [2, 3, 13, 14]) не случайно. Это приводит к получению аналитического решения, что особенно важно для приложений.

Мы представляем символьно-численный алгоритм, основанный на применении преобразования Лапласа.

Рассматриваются системы линейных дифференциальных уравнений с постоянными коэффициентами, правые части которого составные, компоненты которых представляют собой полиномы экспонент, что обеспечивает символьное применение преобразования Лапласа. В результате задача сводится к решению алгебраической системы линейных уравнений с полиномиальными коэффициентами. Решение исходной системы получается в результате обратного преобразования Лапласа. Именно на этом этапе возникает необходимость численных методов для вычисления корней полиномов. Мы даем алгоритм определения погрешности вычислений, достаточной для получения требуемой точности решения исходной системы.

Преимущества построенного алгоритма в следующем:

- размер исходной системы и порядок производных не играют существенной роли;
- в зависимости от вида (размера, плотности и т.д.) выбирается наиболее эффективный способ решения алгебраической системы – тем самым регулируется эффективность решения этим методом исходной системы дифференциальных уравнений;
- требуемая точность решения гарантируется выбором соответствующей погрешности численного фрагмента вычислений.

2. АЛГОРИТМ РЕШЕНИЯ СИСТЕМ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ МЕТОДОМ ПРЕОБРАЗОВАНИЯ ЛАПЛАСА

Рассмотрим систему

$$\sum_{j=1}^{n}\sum_{k=0}^{N} a_{kj}^{l} x_j^{(k)} = f_l, \quad l = 1,\ldots,n, \quad a_{kj}^{l} \in \mathbf{R}, \quad (1)$$

n дифференциальных уравнений порядка N (N – старший по всем неизвестным функциям) n неизвестных функций с начальными условиями $x_j^{(k)}(0) = x_{0j}^{k}, k = 0,\ldots,N-1$. В правой части системы функции, приводимые к виду

$$f_l(t) = f_l^i(t),\ t_l^i < t < t_l^{i+1}, \qquad (2)$$

$$i = 1,\ldots,I_l,\ t_l^1 = 0, t_l^{I_l+1} = \infty,$$

где

$$f_l^i(t) = \sum_{s=1}^{S_l^i} P_{ls}^i(t) e^{b_{ls}^i t}, \quad i = 1,\ldots,I_l, \quad l = 1,\ldots,n,$$

и $P_{ls}^i(t) = \sum_{m=0}^{M_{ls}^i} c_{sm}^{li} t^m$.

Здесь обозначены через $x_j, j = 1,\ldots,n$ неизвестные функции аргумента $t, t \geq 0$, через $x_j^{(k)}$ производные этих x_j, $k = 0,\ldots,N$ порядка k.

Предлагаемый нами алгоритм представим в трех основных этапах.

Этап 1. Преобразование Лапласа данной системы дифференциальных уравнений.

Обозначим образы преобразования Лапласа функций $x_j(t)$ и $f_l(t)$ соответственно через $X_j(p)$ и $F_l(p)$.

Этап 1.1. Преобразование Лапласа левой части системы (1) с учетом начальных условий выполнятся формально согласно свойствам преобразования Лапласа. Получим

$$\sum_{j=1}^{n}\sum_{k=0}^{N} a_{kj}^{l} p^k X_j(p) - \sum_{j=1}^{n}\sum_{k=0}^{N-1} d_{jk}^{l}(p) x_{0j}^{k}, \ l = 1,\ldots,n,$$

где

$$d_{jk}^{l}(p) = \sum_{i=k}^{N-1} a_{i+1,j}^{l} p^{i-k}, \qquad (3)$$

l – номер уравнения.

Этап 1.2. Подготовка функций $f_l(t)$, стоящих в правых частях системы (1), к преобразованию Лапласа проводится применением функции Хевисайда $\eta(t)$, принимающей значение 1 для неотрицательных аргументов и значение 0 для остальных.

Представим $f_l(t)$ в виде суммы функций

$$f_l^i(t) = \phi_l^i(t - t_l^k) = \sum_{s=1}^{S_l^i} \psi_{ls}^i(t - t_l^k) e^{b_{ls}^i t_l^k} e^{b_{ls}^i (t - t_l^k)}.$$

Здесь $\psi_{ls}^i(t - t_l^k) = P_{ls}^i(t)$ and $\psi_{ls}^i(t - t_l^k) = \sum_{j=0}^{J_{ls}^i} \gamma_{lsj}^{ik}(t - t_l^k)^j$, а коэффициенты γ_{lsj}^{mk} вычисляются по формуле

$$\gamma_{lsj}^{ik} = \sum_{r=0}^{M_{ls}^k - j} c_{s,i+r}^{lk} \binom{i+r}{r} (t_l^k)^r.$$

Окончательно, функция $f_l(t)$ представляется в виде

$$f_l(t) = \sum_{i=1}^{I_l-1} [\phi_l^i(t - t_l^i)\eta(t - t_l^i) - \phi_l^i(t - t_l^{i+1})\eta(t - t_l^{i+1})] +$$

$$\phi_l^{I_l}(t - t_{I_l})\eta(t - t_{I_l}).$$

Этап 1.3. Обозначим через $\Phi_l^{i,\tilde{i}}(p)$ образ функции $\phi_l^i(t - t_l^{\tilde{i}})\eta(t - t_l^{\tilde{i}})$. Так как образ преобразования Лапласа функции $(t - t^*)^n e^{\alpha(t - t^*)}\eta(t - t^*)$ равен $\frac{n!}{(p-\alpha)^{n+1}} e^{-t^* p}$, то преобразование Лапласа $f_l(t)$ приводит к следующему:

$$F_l(p) = \sum_{i=1}^{I_l-1} \left[\Phi_l^{i,i}(p) - \Phi_l^{i,i+1}(p)\right] + \Phi_l^{I_l,I_l}(p). \quad (4)$$

Приводя для каждого $l = 1,\ldots,n$ к общему знаменателю выражение (4), получим в числителях суммы экспонент с полиномиальными коэффициентами.

Этап 2. Решение линейной алгебраической системы с полиномиальными коэффициентами.

Рассмотрим алгебраическую систему n линейных уравнений с полиномиальными коэффициентами относительно неизвестных функций $X_j, j = 1,\ldots,n$:

$$\sum_{j=1}^{n}\sum_{k=0}^{N} a_{kj}^{l} p^k X_j(p) = \sum_{j=1}^{n}\sum_{k=0}^{N-1} d_{jk}^{l}(p) x_{0j}^{k} + F_l(p), \qquad (5)$$

$l = 1,\ldots,n.$

Этап 2.1. Обозначим

$$\sum_{j=1}^{n}\sum_{k=0}^{N-1} d_{jk}^{l}(p)x_{0j}^{k} = S_l(p).$$

Систему (5) в этих обозначениях приводим к виду

$$\sum_{j=1}^{n}\sum_{k=0}^{N} a_{kj}^{l}p^{k}X_j(p) = S_l(p) + F_l(p),\ l = 1,\ldots,n. \tag{6}$$

Для каждого $l = 1,\ldots,n$ выражения в правых частях системы (6) приводим к общему знаменателю.

Этап 2.2. Напомним, что в системе (6) полиномы основной матрицы имеют коэффициентами действительные числа (что видно из (1)). Метод решения выбирается в зависимости от вида системы (например, p-адический [9], модулярные методы, методы, основанные на детерминантных тождествах [7, 8] в случае, если коэффициенты рациональные, и другие).

Этап 3. Обратное преобразование Лапласа.

Этап 3.1. Обозначим определитель основной матрицы системы (6) через $D(p)$. Решение системы (6), т.е. функции $X_j(p)$, $j = 1,\ldots,n$, представлены в виде произведения $\frac{1}{D(p)}$ на сумму дробей, числители которых – суммы экспонент с полиномиальными коэффициентами, а знаменатели – натуральные степени биномов вида $(p-\alpha)$. Для символьного выполнения обратного преобразования Лапласа представим $X_j(p)$ как суммы экспонент, коэффициентами которых являются суммы простых дробей.

Этап 3.2. Подготовка к обратному преобразованию Лапласа сводится к разложению рациональных коэффициентов перед экспонентами на простые дроби $A/(p-p^*)^v$, $p^* \in \mathbf{C}$. Для этого требуется найти корни многочлена $D(p)$. Это именно тот шаг алгоритма, который требует численных методов. На этом этапе делается оценка точности этих вычислений (см. Главу **2**). В итоге каждая функция $X_j(p)$ представляется в виде

$$X_j(p) = \sum_m\sum_k \frac{A_{mk}}{(p-p_{jk})^{\beta_{mk}}}e^{-\alpha_m p}. \tag{7}$$

Этап 3.4. Оригиналы преобразования Лапласа для функций (7) записываются формально:

$$x_j(t) =$$

$$\sum_m\sum_k \frac{A_{mk}}{(\beta_{mk}-1)!}(t-\alpha_m)^{\beta_{mk}-1}e^{p_{ik}(t-\alpha_m)}\eta(t-\alpha_m),$$

$j = 1,\ldots,n.$

Заметим, что, вообще говоря, функции $\tilde{x}_l(t)$, полученные как оригиналы $X_j(p)$, построенные по приближенным корням $D(p)$, комплекснозначные. Можно либо перейти к действительным функциям преобразованием экспонент, либо взять их действительные части в качестве решения данной системы. Возникающая при этом погрешность можно учесть в требуемой оценке точности.

3. ОЦЕНКА ТОЧНОСТИ

На **Этапе 3.2** алгоритма численно находятся корни многочлена $D(p)$. Задача – определить, какая погрешность вычисления корней достаточна, чтобы получить требуемую точность решения системы дифференциальных уравнений.

Будем рассматривать все функции и делать вычисления и оценки на отрезке $[0,T]$, где $T > t_l^{I_l}$ для всех $l = 1,\ldots n$, и достаточно велико для поставленной задачи. Обозначим через $\tilde{x}_l(t)$, $l = 1,\ldots n$, решение системы (1), полученное с использованием приближенных корней $D(p)$. Будем говорить, что решение исходной системы на $[0,T]$ имеет точность ε, если

$$\max_{t\in[0,T]}|x_l(t)-\tilde{x}_l(t)| < \varepsilon,\ l = 1,\ldots,n. \tag{8}$$

Требуется найти погрешность Δ вычисления корней $D(p)$, достаточную для получения точности ε.

Обозначим через $T_i^l(p)$ минор (i,l) матрицы системы (6). Решение $X_l(p)$ системы (6) может быть записано в виде

$$X_l(p) = \frac{D_l(p)}{D(P)},$$

где

$$D_l(p) = \sum_{i=1}^{m}\left[F_l(p) + S_l(p)\right]T_i^l(p).$$

Обозначим через p_r точные корни и через $p_r^*, r = 1,\ldots,n$, приближенные корни полинома

$D(p)$. Если $|p_r - p_r^*| < \Delta$, то погрешность вычислений меньше Δ.

Теорема *Для любого ε существует Δ такое, что при условии $|p_r - p_r^*| < \Delta$, имеет место (8)*

Доказательство

Рассмотрим многочлен $D(p+\Delta e^{i\alpha}), \alpha \in [0, 2\pi]$. Обозначим

$$\tilde{X}_l(p) = \frac{D_l(p)}{D(p+\Delta e^{i\alpha})}. \tag{9}$$

Требуется найти Δ для которого выполняется (8).

Оценим оригинал разности

$$\frac{D_l(p)}{D(p)} - \frac{D_l(p)}{D(p+\Delta e^{i\alpha})}.$$

Линейность преобразования Лапласа позволяет оценить независимо оригиналы

$$\sum_{i=1}^{n} F_l(p) \frac{T_i^l(p)}{D(p)} - \sum_{i=1}^{n} F_l(p) \frac{T_i^l(p)}{D(p+\Delta e^{i\alpha})} \tag{10}$$

и

$$\sum_{i=1}^{n} S_l(p) \frac{T_i^l(p)}{D(p)} - \sum_{i=1}^{n} S_l(p) \frac{T_i^l(p)}{D(p+\Delta e^{i\alpha})}. \tag{11}$$

Запишем выражение (10) в форме:

$$\sum_{i=1}^{n} F_l(p) \left(\frac{T_i^l(p)}{D(p)} - \frac{T_i^l(p)}{D(p+\Delta e^{i\alpha})} \right). \tag{12}$$

Обе функции – $F_l(p)$ и разность, стоящая в скобках – имеют оригиналы. Оригинал $\Psi_l(t)$ выражения (10) можно найти как сумму сверток $f_l(t)$ и оригинала $\Omega_l^i(t)$ функции

$$\frac{T_i^l(p)}{D(p)} - \frac{T_i^l(p)}{D(p+\Delta e^{i\alpha})}. \tag{13}$$

Представим выражение (13) в виде

$$\frac{T_i^l(p)}{D(p)} - \frac{T_i^l(p+\Delta e^{i\alpha})}{D(p+\Delta e^{i\alpha})} + \frac{T_i^l(p+\Delta e^{i\alpha}) - T_i^l(p)}{D(p+\Delta e^{i\alpha})}. \tag{14}$$

Обозначим через $Q_l^i(t)$ и $\widetilde{Q_l^i(t)}$ оригиналы функций $\frac{T_i^l(p)}{D(p)}$ и $\frac{T_i^l(p)}{D(p)} - \frac{T_i^l(p+\Delta e^{i\alpha})}{D(p+\Delta e^{i\alpha})}$ соответственно. Из свойств преобразования Лапласа следует, что:

$$\widetilde{Q_l^i(t)} = \left(1 - e^{\Delta T}\right) Q_l^i(t). \tag{15}$$

Теперь оценим $Q_l^i(t)$. Функция $Q_l^i(t)$ может быть записана в форме

$$Q_l^i(t) = \sum_{r=1}^{R} \left(\sum_{\mu=1}^{\mu_r} \mathcal{B}_{r\mu}^i \frac{t^{\mu_r-\mu}}{(\mu_r-\mu)!)} \right) e^{p_r t}, \tag{16}$$

где p_r – корни $D(p)$ кратности μ_r, R – количество различных корней.

Коэффициенты $\mathcal{B}_{r\mu}^i$ можно вычислить и оценить как Тейлоровские коэффициенты функции $(p-p_r)^{\mu_r} \frac{T_i^l(p)}{D(p)}$ в δ-окрестности точки p_r. Пусть $\delta < 1/2 \min |p_r - p_s|, r, s = 1, \ldots R, \quad \delta < 1$. Мы должны получить $\Delta \leq \delta$, в противном случае возьмем $\Delta = \delta$.

Из неравенств Коши для тейлоровских коэффициентов следует:

$$|\mathcal{B}_{r\mu}^i| \leq \left(\max\nolimits_{|p-p_r|=\delta} (p-p_r)^{\mu_r} \frac{T_i^l(p)}{D(p)} \right) / \delta^{\mu}.$$

Рассмотрим многочлен $D(p) = \sum_{\nu=0}^{mn} c_\nu p^\nu$. Так как

$$\left| (p-p_r)^{\mu_r} \frac{T_i^l(p)}{D(p)} \right| = \frac{|T_i^l(p)|}{|c_{mn}| \prod_{s=1, s\neq r}^{R} |p-p_r|^{\mu_s}},$$

то нам нужно оценить $|T_i^l(p)|$ в δ-окрестности p_r. Обозначим через ρ радиус круга, содержащего все нули $D(p)$. В качестве такого ρ можно взять число

$$\rho = \max \left\{ 1; \frac{\sum_{\nu=0}^{mn} |c_\nu|}{|c_{mn}|} \right\},$$

причем его можно выбрать так, чтобы

$$\max\nolimits_{|p-p_r|=\delta} |T_i^l(p)| \leq \max\nolimits_{|p|=\rho} |T_i^l(p)|.$$

Элемент матрицы, определитель которой обозначен $T_i^l(p)$, – это многочлен $\gamma_{\eta,\lambda}(p)$, где λ – номер строки, а $\lambda \neq l$, η – номер столбца, $\eta \neq i$. Обозначим $m_{\eta,\lambda} = \max_{|p|=\rho} |\gamma_{\eta,\lambda}(p)|$. Из неравенств Адамара следует

$$\max\nolimits_{|p|=\rho} |T_i^l(p)| \leq \sqrt{\prod_{\eta=1,\eta\neq i}^{n} \sum_{\lambda=1,\lambda\neq l}^{n} m_{\eta,\lambda}^2}.$$

Обозначим

$$M_{il} = \sqrt{\prod_{\eta=1,\eta\neq i}^{n} \sum_{\lambda=1,\lambda\neq l}^{n} m_{\eta,\lambda}^2}. \tag{17}$$

Принимая во внимание $|p - p_s| \geq \delta$ и $\mu \leq |mu_r$, получим:

$$|\mathcal{B}^i_{r\mu}| \leq \frac{M_{il}}{|c_{mn}|\delta^{mn}}.$$

В результате получаем оценку $Q^i_l(t)$ на отрезке $[0, T]$:

$$|Q^i_l(t)| \leq \frac{mn}{|c_{mn}|\delta^{mn}} M_{il} e^{\rho T}. \quad (18)$$

Рассмотрим последнее слагаемое в выражении (14).

По формуле конечных приращений

$$T^l_i(p) - T^l_i(p + \Delta e^{i\alpha}) =$$

$$\Delta e^{i\alpha} \frac{1}{2\pi i} \int_{|\xi - p| = \delta} \frac{T^l_i(\xi)}{(\xi - p - \Delta e^{i\alpha})(\xi - p)} d\xi. \quad (19)$$

Из (19) получим:

$$|T^l_i(p) - T^l_i(p + \Delta e^{i\alpha})| \leq \frac{\Delta}{\delta - \Delta} M_{il}. \quad (20)$$

Как в предыдущем случае, получим неравенство для оригинала $\widetilde{\widetilde{Q^i_l(t)}}$ функции $\frac{T^l_i(p + \Delta e^{i\alpha}) - T^l_i(p)}{D(p + \Delta e^{i\alpha})}$:

$$\left|\widetilde{\widetilde{Q^i_l(t)}}\right| \leq \frac{\Delta}{\delta - \Delta} |Q^i_l(t)|. \quad (21)$$

Пользуясь (15), (18), (20) и (21) оценим $\Omega^i_l(t)$:

$$|\Omega^i_l(t)| \leq \left(e^{\Delta T} - 1 + \frac{\Delta}{\delta - \Delta}\right) \frac{mn}{|c_{mn}|\delta^{mn}} M_{il} e^{\rho T}. \quad (22)$$

Оригинал $f_i(t)$ функции $F_i(p)$ – составная функция. Каждая компонента $f_i(t)$ – это сумма экспонент с полиномиальными коэффициентами. Оцениваем ее, находя максимумы коэффициентов на соответствующих отрезках и значения экспонент на их правых концах. Обозначим

$$\max_{t \in [0,T]} |f_i(t)| = Mf_i. \quad (23)$$

Оценивая свертку оригиналов функции (12), получим оценку для $\Psi_l(t)$:

$$\max_{t \in [0,T]} |\Psi_l(t)| \leq (e^{\Delta T} - 2) \frac{mnT}{|c_{mn}|\delta^{mn}} \sum_{i=1}^{n} Mf_i \, M_{il} \, e^{\rho T}.$$

Обозначим

$$\Lambda_l = \frac{mnT}{|c_{mn}|\delta^{mn}} \sum_{i=1}^{n} Mf_i \, M_{il} \, e^{\rho T}. \quad (24)$$

Рассмотрим (11). Многочлен $S_i(p)$ не имеет оригинала. Обозначая $\sum_{i=1}^{n} S_i(p) T^l_i(p) = K_l(p)$, $\frac{K_l(p)}{D(p)} = N_l(p)$, запишем (11) в форме

$$\frac{K_l(p)}{D(p)} - \frac{K_l(p)}{D(p + \Delta e^{i\alpha})} = \frac{K_l(p)}{D(p)} - \frac{K_l(p + \Delta e^{i\alpha})}{D(p + \Delta e^{i\alpha})} +$$

$$\Delta e^{i\alpha} \frac{1}{2\pi i} \int_{|\xi - p| = \delta} \frac{N_l(p)}{(\xi - p - \Delta e^{i\alpha})(\xi - p)} d\xi. \quad (25)$$

Оригиналы $R_l(t)$ функций $N_l(p)$ можно представить в виде (16). Оценки, аналогичные предыдущим, приводят к неравенствам:

$$\max_{t \in [0,T]} |R_l(t)| \leq \frac{mnT}{|c_{mn}|\delta^{mn}} \sum_{i=1}^{n} \max_{|p| \leq \rho} | \, S_i(p)| M_{il} \, e^{\rho T}. \quad (26)$$

Из (5) очевидно, что

$$\max_{|p| \leq \rho} |S_i(p)| \leq \sum_{\nu = k}^{m-1} |a^i_{\nu+1,j}| \rho^{\nu - k} |x^k_{oj}|. \quad (27)$$

Обозначим

$$L_l = \frac{mnT}{|c_{mn}|\delta^{mn}} \sum_{i=1}^{n} \max_{|p| \leq \rho} |S_i(p)| \; M_{il} \; e^{\rho T}. \quad (28)$$

Аналогично предыдущему, получим оценки для $\Theta_l(t)$ оригинала функции (11):

$$|\Theta_l(t)| \leq (e^{\Delta T} - 1 + \frac{\Delta}{\delta - \Delta}) \; L_l. \quad (29)$$

Окончательно, из (22) и (29) получим оценки для $x_l(t)$:

$$\max_{t \in [0,T]} |x_l(t) - \tilde{x}_l(t)| \leq \left(e^{\Delta T} - 1 + \frac{\Delta}{\delta - \Delta}\right) (\Lambda_l + L_l). \quad (30)$$

Предположим, что $\frac{k}{k+1}\delta < \Delta < \delta, k \geq 0$. Тогда неравенство (30) принимает вид

$$\max_{t \in [0,T]} |x_l(t) - \tilde{x}_l(t)| \leq \left(e^{\Delta T} - 1 + k\right) (\Lambda_l + L_l). \quad (31)$$

Исходя из (9) и (31), потребуем

$$\left(e^{\Delta T} - 1 + k\right) (\Lambda_l + L_l) < \varepsilon. \quad (32)$$

Из (32) получаем оценку для Δ при каждом l, обозначим ее Δ_l:

$$\Delta_l < \frac{1}{T} \ln \left(\varepsilon \, (\Lambda_l + L_l)^{-1} + 1 - k\right). \quad (33)$$

Возьмем

$$k < \min_l (\Lambda_l + L_l)^{-1}. \qquad (34)$$

Окончательно, в качестве искомого Δ возьмем $\Delta = \min_l \Delta_l$.

Теорема доказана.

Таким образом, алгоритм получения погрешности, достаточной для обеспечения требуемой точности решения, состоит в следующем.

Этап 4. Алгоритм вычисления погрешности

Этап 4.1. Для матрицы системы (6) вычисление чисел $m_{\eta,\lambda} = \max_{|p|=\rho}|\gamma_{\eta,\lambda}(p)|$ и M_{il} (см. (17)).

Этап 4.2. Вычисление Mf_i (см. (23)).

Этап 4.3. Вычисление Λ_l (см. (24)) согласно значениям δ и ρ.

Этап 4.4. Вычисление L_l (см. (28)), оценка многочленов $S_l(p)$ как в (27).

Этап 4.5. Окончательный результат для Δ_l и $\Delta = \min_l \Delta_l$ согласно формулам (33) и (34).

Таким образом, в процессе доказательства Теоремы были получены формулы для вычисления погрешности Δ по заданной точности ε решения системы (1). Они представлены на Этапе 4.

4. О СЛОЖНОСТИ АЛГОРИТМА

Сложность алгоритма зависит от сложности трех основных операций: решение алгебраической линейной системы с полиномиальными коэффициентами на этапе (2.2), решение алгебраической линейной системы с постоянными коэффициентами на этапе (3.3), вычисление корней многочлена на этапе 3.2.

На этапе (2.2) существенными для определения сложности являются вычисление обратной матрицы, умножение полиномиальной матрицы на полиномиальный вектор. Сложность различных алгоритмов для этих вычислений так же, как для решения системы на шаге 3.3, вычисляется соответствующим образом (см., например, [6]). Здесь один из наиболее эффективных методов – p-адический [9], для распараллеливания вычислений – модулярные методы. Сложность на этапе 3.2 вычисляется, например в [1].

5. ПРИМЕР

Дана система дифференциальных уравнений:

$$\begin{cases} x_1''' - x_1' - 2x_1 - x_2''' + x_2 &= f_1 \\ 3x_1''' + x_1'' - 2x_1' + x_2''' + x_2 &= f_2. \end{cases}$$

Функции f_1 и f_2 и числа t_l^i:
$f_1^1 = e^t$, $f_1^2 = t^2e^{2t}$, $t_1^1 = 0$, $t_1^2 = 1$;
$f_2^1 = te^t$, $f_2^2 = e^{2t}$, $t_2^1 = 0$, $t_2^2 = 1$.
Начальные условия:
$x_{01}^0 = 5$, $x_{01}^1 = 10$, $x_{01}^2 = 30$, $x_{02}^0 = 4$, $x_{02}^1 = 14$, $x_{02}^2 = 20$.

Этап 1.1.

$$-2X_1 - pX_1 + p^3X_1 + X_2 - p^3X_2 - (10 - 4p - 4p^2 + 5(-1 + p^2));$$
$$-2pX_1 + p^2X_1 + 3p^3X_1 + X_2 + p^3X_2 - (110 + 14p + 4p^2 + 10(1 + 3p) + 5(-2 + p + 3p^2)).$$

Этап 1.2.
$f_1 := (f_1^2 - f_1^1)\mathrm{UnitStep}[(t - t_1^2)] + f_1^1\mathrm{UnitStep}[t]$;
$f_2 := (f_2^2 - f_2^1)\mathrm{UnitStep}[(t - t_2^2)] + f_2^1\mathrm{UnitStep}[t]$.
Этап 1.3.

$$F_1 = \frac{1}{-1+p} - \frac{e^{1-p}}{-1+p} + \frac{e^{2-p}(p^2-2p+2)}{(-2+p)^3};$$
$$F_2 = \frac{e^{2-p}}{-2+p} + \frac{1}{(-1+p)^2} - \frac{e^{1-p}p}{(-1+p)^2}.$$

Этап 2.1.

$$-2X_1 - pX_1 + p^3X_1 + X_2 - p^3X_2 = 10 - 4p - 4p^2 + 5(-1 + p^2) + \frac{1}{-1+p} - \frac{e^{1-p}}{-1+p} + \frac{e^{2-p}(p^2-2p+2)}{(-2+p)^3};$$

$$-2pX_1 + p^2X_1 + 3p^3X_1 + X_2 + p^3X_2 = 110 + 14p + 4p^2 + 10(1 + 3p) + 5(-2 + p + 3p^2) + \frac{e^{2-p}}{-2+p} + \frac{1}{(-1+p)^2} - \frac{e^{1-p}p}{(-1+p)^2}.$$

Этап 2.2.

$$X_1(p) = e^{-p}(8e + 2e^2 - 4ep - 4e^2p + 2ep^2 + 2e^2p^2 + 9ep^3 - 6e^2p^3 - 19ep^4 + 12e^2p^4 + 11ep^5 - 8e^2p^5 - 2ep^6 + 2e^2p^6)/((-2 + p)^3(-1 + p)(-2 + p - p^2 - 4p^3 - 3p^4 + p^5 + 4p^6)) + (-856 + 1692p - 982p^2 + 1061p^3 - 1991p^4 +$$

$\mathtt{1398p^5 - 412p^6 + 160p^7 - 95p^8 + 20p^9)/((-2+p)^3(-1+p)(-2+p-p^2-4p^3-3p^4+p^5+4p^6))};$

$$\mathtt{X_2(p)} = \mathtt{e^{-p}(-8e^2 - 32ep + 24e^2p + 64ep^2 - 28e^2p^2 - 32ep^3 + 17e^2p^3 - 24ep^4 - 5e^2p^4 + 34ep^5 - 3e^2p^5 - 14ep^6 + 5e^2p^6 + 2ep^7 - 2e^2p^7)/((-2+p)^3(-1+p)^2(-2+p-p^2-4p^3-3p^4+p^5+4p^6)) + (1776 - 4576p + 3568p^2 - 1404p^3 + 2465p^4 - 2751p^5 + 841p^6 + 133p^7 + 2p^8 - 68p^9 + 16p^{10})/((-2+p)^3(-1+p)^2 (-2+p-p^2-4p^3-3p^4+p^5+4p^6))}.$$

`Этап 3.1.`

$\mathtt{D(p)} = -2 + \mathtt{p} - \mathtt{p}^2 - 4\mathtt{p}^3 - 3\mathtt{p}^4 + \mathtt{p}^5 + 4\mathtt{p}^6.$

Приведем окончательный результат для произвольного T:

`Этап 4.3.`

$\Lambda_1 = (6240\mathtt{T}^3 + 20160\mathtt{T}^2)\mathtt{e}^{5\mathtt{T}}$,

$\Lambda_2 = (6240\mathtt{T}^3 + 6720\mathtt{T}^2)\mathtt{e}^{5\mathtt{T}}$.

`Этап 4.4.`

$\mathtt{L}_1 = 9438240\mathtt{e}^{4\mathtt{T}}$; $\mathtt{L}_2 = 3874080\mathtt{e}^{4\mathtt{T}}$.

Возьмем, например, $k = \frac{1}{2}\min_l (\Lambda_l + L_l)^{-1}$, $T = 2$. Для $\varepsilon = 0.01$ получаем $\Delta = 8.06 \cdot 10^{-14}$, для $\varepsilon = 0.001$ $\Delta = 7.99 \cdot 10^{-15}$.

Возьмем $\Delta = 10^{-15}$, найдем приближенные корни $D(p)$.

$\mathtt{p}_1^* = -1.00000000000000$,

$\mathtt{p}_2^* = -0.594937842169665 - \mathtt{i}0.830713582043548$,

$\mathtt{p}_3^* = -0.594937842169665 + \mathtt{i}0.830713582043548$,

$\mathtt{p}_4^* = 0.355937297682321 - \mathtt{i}0.513128324882554$,

$\mathtt{p}_5^* = 0.355937297682321 + \mathtt{i}0.513128324882554$,

$\mathtt{p}_6^* = 1.22800108897469$.

Окончательно получаем решение данной системы дифференциальных уравнений

$$\mathtt{x_1(t)} = 10.031249\mathtt{e}^{-\mathtt{t}} - 1.25\mathtt{e}^{\mathtt{t}} + 5.538602\mathtt{e}^{1.228001\mathtt{t}} + 2\mathtt{e}^{0.355937\mathtt{t}}(-3.735568\cos(0.513128\mathtt{t}) + 15.529795\sin(0.513128\mathtt{t})) + 2\mathtt{e}^{-0.594937\mathtt{t}}(-0.924357\cos(0.830713\mathtt{t}) + 0.061193\sin(0.830713\mathtt{t}));$$

$$\mathtt{x_2(t)} = 10.03125e^{-t} + 0.5e^{t} - 8.948223e^{1.228001t} + 0.5\mathtt{e}^{\mathtt{t}}\mathtt{t} + 2\mathtt{e}^{0.355937\mathtt{t}}(-0.493116\cos(0.513128\mathtt{t}) + 33.959275\sin(0.513128\mathtt{t})) + 2\mathtt{e}^{-0.594938\mathtt{t}}(1.701602\cos(0.830713\mathtt{t}) + 0.929609\sin(0.830713\mathtt{t})).$$

6. ПАКЕТЫ ПРОГРАММ ДЛЯ РЕШЕНИЯ СИСТЕМ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

Приведенный здесь алгоритм был программно реализован в системе компьютерной алгебры Mathpar

Саму систему и „Руководство пользователя“ к ней можно смотреть на сайте http: \\ mathpar.com.

Разработанный программный **пакет** ***laplaceTransform*** состоит из трёх основных классов.

1. **Класс SystemLDE** включает в себя процедуры, предназначенные для решения систем дифференциальных уравнений.

Процедура `systDifEquationToMatrix` вычисляет полиномиальную матрицу $A(p)$, которая получается в результате преобразования Лапласа левой части системы (1).

Процедура `initCondLaplaceTransform` вычисляет вектор-функцию $\mathcal{S}(p)$, с элементами $S_l(p)$, $l = 1..n$, которая получается в результате преобразования Лапласа начальных условий.

Процедура `primeFractionForSingle` предназначена для разложения дроби $1/D(p)$ с факторизованным знаменателем $D(p)$ в сумму простых дробей. Числители этих дробей находятся методом неопределённых коэффициентов.

Процедура `primeFractions` предназначена для разложения суммы дробей $\sum_{v=1}^{V} 1/w_v(p)$, у которых факторизованы знаменатели $w_v(p)$, в сумму простых дробей, для этапа 3.1.

2. **Класс LaplaceTransform** включает в себя процедуры, предназначенные для вычисления прямого преобразования Лапласа функций. Процедура `vectorLaplaceTransform` вычисляет преобразования Лапласа (этап 1.2, 1.3) правой части системы (1).

3. **Класс InverseLaplaceTransform** включает в себя процедуры, предназначенные для вычисления обратного преобразования Лапласа.

Процедура `vectorInverseLaplaceTransform` вычисляет обратное преобразование Лапласа для вектор-функции $X(p)$, с элементами $X_j(p)$. В результате получаем решение $x(t)$ системы (1).

Программный пакет *laplaceTransform* использует следующие программные пакеты системы Mathpar: *number*, *matrix*, *polynom*, *func*.

Пакет *number* предназначен для задания числовых множеств и операций над ними. Кроме того, в пакете *number* имеются два дополнительных класса — Product и SumOfProducts.

1. **Класс Product** включает в себя процедуры, предназначенные для операций над объектами вида:

$$\prod_i \alpha_i^{n_i},$$

где $n, i \in \mathbb{N}$, а α_i — скалярные объект системы Mathpar (числа, полиномы, функции). Все элементы в произведении упорядочены.

2. **Класс SumOfProducts** включает в себя процедуры, предназначенные для операций над объектами вида:

$$\sum_j \prod_i \alpha_{ij}^{n_{ij}}.$$

Процедура `primeFractions` класса SumOfProducts производит разложение объектов в сумму простых дробей.

Процедура `reduction` приводит подобные в объекте SumOfProducts.

Пакет *matrix* предназначен для операций над числовыми и функциональными матрицами.

Процедура `matrixAdjointDet` пакета *matrix* позволяет вычислить определитель и присоединённую матрицу для матрицы полиномов на этапе 2.

Процедура `solveLAE` вычисляет решение системы на этапе 2.

Пакет *polynom* предназначен для операций над полиномами. Одна из процедур пакета – процедура `factorOfPol` раскладывает полином одной переменной на множители в поле комплексных чисел.

Пакет *func* предназначен для операций над суперпозициями трансцендентных и рациональных функций (см. [15]).

Описание этих пакетов можно смотреть в [15].

7. ЗАКЛЮЧЕНИЕ

Сформулируем еще раз преимущества алгоритма, приведенного в статье:

1. Преобразование Лапласа – это путь к символьному решению систем дифференциальных уравнений, так как приводит к решению алгебраической системы линейных уравнений.

2. Для алгебраической системы выбирается наиболее эффективный метод решения в зависимости от ее свойств.

3. Размер системы и порядок производных может быть произвольным.

4. На численном этапе решения оценивается погрешность, достаточная для получения требуемой точности решения системы дифференциальных уравнений.

5. Алгоритм реализован в системе компьютерной алгебры Mathpar.

8. ЗАМЕЧАНИЯ.

Решение приведенного примера в Mathpar на PC с 1 гб оперативной памяти занимает время порядка секунды.

Тот же пример был реализован в среде Maple на PC с 4 гб оперативной памяти.

$[>\ \mathtt{dsys} := \{\frac{\mathtt{d}}{\mathtt{dt}}\left(\frac{\mathtt{d}}{\mathtt{dt}}\left(\frac{\mathtt{d}}{\mathtt{dt}}\mathtt{x(t)}\right)\right) - \left(\frac{\mathtt{d}}{\mathtt{dt}}\mathtt{x(t)}\right) - 2\cdot\mathtt{x(t)} - \frac{\mathtt{d}}{\mathtt{dt}}\left(\frac{\mathtt{d}}{\mathtt{dt}}\left(\frac{\mathtt{d}}{\mathtt{dt}}\mathtt{y(t)}\right)\right) + \mathtt{y(t)} =$
$(\mathtt{t}^2\cdot\mathtt{e}^{2\cdot\mathtt{t}}-\mathtt{e}^{\mathtt{t}})\cdot\mathtt{Heaviside(t-1)}+\mathtt{e}^{\mathtt{t}}\cdot\mathtt{Heaviside(t)},$
$3\cdot\frac{\mathtt{d}}{\mathtt{dt}}\left(\frac{\mathtt{d}}{\mathtt{dt}}\left(\frac{\mathtt{d}}{\mathtt{dt}}\mathtt{x(t)}\right)\right) + \frac{\mathtt{d}}{\mathtt{dt}}\left(\frac{\mathtt{d}}{\mathtt{dt}}\mathtt{x(t)}\right) - 2\cdot\left(\frac{\mathtt{d}}{\mathtt{dt}}\mathtt{x(t)}\right) + \frac{\mathtt{d}}{\mathtt{dt}}\left(\frac{\mathtt{d}}{\mathtt{dt}}\left(\frac{\mathtt{d}}{\mathtt{dt}}\mathtt{y(t)}\right)\right) + \mathtt{y(t)} =$
$(\mathtt{e}^{2\cdot\mathtt{t}}-\mathtt{t}\cdot\mathtt{e}^{\mathtt{t}})\cdot\mathtt{Heaviside(t-1)}+\mathtt{t}\cdot\mathtt{e}^{\mathtt{t}}\cdot\mathtt{Heaviside(t)},$
$\mathtt{D(D(x))(0)} = 30, \mathtt{D(D(y))(0)} = 20, \mathtt{D(x)(0)} = 10, \mathtt{D(y)(0)} = 14, \mathtt{x(0)} = 5,$
$\mathtt{y(0)} = 4\}$

$[>$ `dsolve(dsys)`

После нескольких минут счета было получено сообщение о переполнении оперативной памяти.

СПИСОК ЛИТЕРАТУРЫ

1. *Akritas A.G.* Elements of Computer Algebra with Applications. J. Wiley Interscience, New York, 1989, 448 p.

2. *Burghelea D., Haller S.* Laplace transform, dynamics and spectral geometry. arXiv:math.DG/0405037v2, 17 Jan., 2005.

3. *Dahiya R.S., Jabar S.-N.* Theorems on n-dimensional Laplace transforms and their applications. In: 15th Annual Conf. of Applied Math., Univ. of Central Oklahoma, Electr. J. of Differential Equations, Conf. 02, 1999. P. 61–74.

4. *Давенпорт Дж., Сире И., Турнье Э.* Системы и алгоритмы алгебраических вычислений. М., МИР, 1991, 352 с.

5. *Von zur Gathen J., Gerhard J.* Modern Computer Algebra. Cambridge University Press, 1999, 771 p.

6. *Malaschonok G.I.* Algorithm for the solution of systems of Linear Equations in commutative rings. In: Effective Methods in Algebraic Geometry. T. Mora and C. Traverso (Eds.). Progress in Mathematics. V. 94. Birkhauser. 1991. P. 289–298.

7. *Malaschonok G.I.* Recursive Method for the Solution of systems of Linear Equations. In: Computational Mathematics (15th IMACS World Congress V. I, Berlin, August 1997), Wissenschaft und Technik Verlag. Berlin. 1997. P. 475–480.

8. *Malaschonok G.I.* Effective Matrix Methods in Commutative Domains. In: Formal Power Series and Algebraic Combinatorics, Springer, Berlin. 2000. P. 506–517.

9. *Малашонок Г.И.* Решение систем линейных уравнений p-адическим методом. Программирование. 2003. Т. 29. № 2. P. 8–22.

10. *Malaschonok N.* An algorithm to settle the necessary exactness in Laplace transform method. In: Computer Science and Information Technologies. Yerevan. 2005. P. 453–456.

11. *Malaschonok N.* Estimations in Laplace transform method for solutions of differential equations in symbolic computations. In: Differential Equations and Computer Algebra Systems. Minsk. 2005. P. 195–199.

12. *Malaschonok N.* Parallel Laplace method with assured accuracy for solutions of differential equations by symbolic computations. In: Computer Algebra and Scientific Computing, CASC 2006, LNCS 4196, Springer, Berlin. 2006. P. 251–261.

13. *Mizutani N., Yamada H.* Modified Laplace transformation method and its applications to anharmonic oscillator. International J. of Modern Physics A, December. 1997. V. 12. № 31. P. 5687–5709.

14. *Podlubny I.* The Laplace transform method for linear differential equations of the fractional order. arXiv:funct-an/9710005v1, 30 Oct. 1997.

15. *Malaschonok G.I.* Project of Parallel Computer Algebra // Tambov University Reports. Series: Natural and Technical Sciences. 2010. V. 15. I. 6. P. 1724–1729.