Содержание

Введение

В данной дипломной работе рассматривается коэффициентная обратная задача уравнения аномальной диффузии с лево-и правосторонними дробными производными типа Римана-Лиувилля по пространственной переменной. Определению подлежит коэффициент диффузии $k(U)$.

Дробное исчисление [2] является эффективным и широко используемым инструментом для изучения процессов с аномальной кинетикой. Широкий спектр дифференциальных уравнений с дробными производными по времени и пространству был предложен и исследован большим числом ученых за последние два десятилетия для моделирования разнообразных процессов релаксации и переноса. Примерами таких уравнений являются уравнения суб- и супердиффузии [4], уравнение Фоккера-планка [6], дробные уравнения адвекции-дисперии [7].

Однако, для практического использования этих уравнений необходимо знать соответствующие свойства среды, входящие в эти уравнения в качестве коэффициентов. В частности, дифференциальное уравнение аномальной диффузии одним из важнейших параметров является коэффициент диффузии. Как правило, этот коэффициент зависит от концентрации. Поэтому, представляется актуальной задача разработки алгоритмов параметрической идентификации для уравнения с производными дробного порядка. В данной дипломной работе объектом исследования является уравнение аномальной диффузии, содержащее лево- и правосторонние дробные производные типа Римана-Лиувилля про пространственной переменной. Разработка алгоритма идентификации основывается на сведении задачи к функционально-интегральному уравнению. Таким образом, целью работы является решение коэффициентной обратной задача уравнения аномальной диффузии с лево-и правосторонними дробными производными типа Римана-Лиувилля по пространственной переменной.

Для достижения данной цели в работе решаются следующие задачи

1. Для уравнения аномальной диффузии с лево- и правосторонними дробными производными типа Римана-Лиувилля по пространственной переменной вывести функционально-интегральное уравнение относительно коэффициента аномальной диффузии, рассматриваемого как функция концентрации.

2. Разработать численный алгоритм решения построенного функционально-интегрального уравнения.

3. Выполнить программную реализацию алгоритма.

4. Провести серию вычислительных экспериментов, подтверждающих работоспособность предложенного алгоритма.

1 Функционально-интегральное уравнение для идентификации коэффициента аномальной диффузии

1.1 Постановка обратной задачи

Рассматривается уравнение аномальной диффузии:

$$U_t = \frac{\partial}{\partial x}\left[k(U) \cdot ({}_0D_x^\alpha(U) -_x D_L^\alpha(U))\right], \tag{1}$$

где

$$U = U(t,x), x \in [0,L], t \in [0,T], \alpha \in (0,1), \tag{2}$$

со следующими начальным и граничным условиями:

$$U(0,x) = \phi(x), x \in [0,L], \tag{3}$$

$$k(U) \cdot ({}_0D_x^\alpha(U) -_x D_L^\alpha(U)) = \begin{cases} q_0(t), x = 0, \\ q_L(t), x = L \end{cases}, t \in [0,T]. \tag{4}$$

В (1)

$${}_0D_x^\alpha(U(t,x)) = \frac{1}{\Gamma(1-\alpha)}\frac{\partial}{\partial x}\int\limits_0^x \frac{U(t,\tau)}{(x-\tau)^\alpha}d\tau, \alpha \in (0,1), n \in R, \tag{5}$$

$${}_xD_L^\alpha(U(t,x)) = \frac{-1}{\Gamma(1-\alpha)}\frac{\partial}{\partial x}\int\limits_x^L \frac{U(t,\tau)}{(\tau-x)^\alpha}d\tau, \alpha \in (0,1), n \in R, \tag{6}$$

- лево- и правосторонние дробные производные типа Римана-Лиувилля порядка $\alpha \in (0,1)$ [2],

$$\Gamma(\alpha) = \int\limits_0^\infty t^{\alpha-1}e^{-t}dt, \alpha \in C : Re(\alpha) > 0 \tag{7}$$

- гамма - функция.

Ставится коэффициентная обратная задача восстановления коэффициента аномальной диффузии $k(U)$ по полю концентрации, известному в дискретном множестве N внутренних точек $x = d_i$ во все моменты времени:

$$U(t,x)\Big|_{x=d_i} = f_i(t), i = 1 \dots N \tag{8}$$

1.2 Вывод функционально-интегрального уравнения

Для решения коэффициентой обратной задачи используется подход, описанный в [1], который адаптируется для одномерной области $\Omega = [0, L]$, и уравнения с дробными производными типа Римана-Лиувилля.

Пусть Ω - конечная область евклидового пространства R^1, ограниченная границей $\Gamma = \partial\Omega$; v - нормаль к поверхности Γ, внешняя по отношению к Ω;

$$g(t) = \min_{x\in\bar{\Omega}} U(t,x), \ \ h(t) = \max_{x\in\bar{\Omega}} U(t,x), \ \ (\bar{\Omega} = \Omega \cup \Gamma) \tag{9}$$

- минимальная и максимальная концентрации (экстремальные концентрации) в замкнутой области $\bar{\Omega}$ в момент времени t; $\underline{g} = \min\limits_{0\le t\le T} g(t)$, $\bar{h} = \max\limits_{0\le t\le T} h(t)$ - наименьшая и наибольшая концентрации за время эксперимента T ; U_r - множество значений функции $r(t), r = g, h$.

Принято, что концентрация $U(t,x)$ является непрерывной функцией на $\bar{\Omega} \times (0,T)$. Следовательно, при каждом фиксированном $t \in (0,T)$ множество значений концентрации есть отрезок $[g(t), h(t)]$, причем для каждой экстремальной концентрации $r, r = g, h$ существует по крайней мере одна точка $x_r(t)$ в которой $U(t, x_r(t)) = r(t)$.

Введем изотерму

$$x(s,t) = \{x : x \in \Omega, U(t,x) = s, g(t) \le s \le h(t)\}. \tag{10}$$

Она разделяет всю область Ω на два множества, Ω_1 и Ω_2:

$$\begin{aligned} \Omega_1 = \Omega_1(t,s) = \{x : x \in \Omega, g(t) \le U(t,x) < s \le h(t)\}, \\ \Omega_2 = \Omega_2(t,s) = \{x : x \in \Omega, h(t) \ge U(t,x) > s \ge g(t)\}. \end{aligned} \tag{11}$$

Введем также характеристическую функцию следующего вида:

$$\chi(r(t), U(t,x), s) = \begin{cases} 1, s \in (r(t), U(t,x)), \\ 0, s \notin (r(t), U(t,x)). \end{cases} \tag{12}$$

Утверждение 1. *Пусть функции $U(t,x)$ и $k(U)$ удовлетворяют уравнению (1) и граничному условию (4); экстремальные концентрации $g(t)$, $h(t)$ дифференцируемы на $(0,T)$. Пусть $\rho(t,x)$, $U(t,x)$ таковы, что каждый член уравнения (1), умноженный на $\rho(t,x)$, интегрируем по $x \in \Omega = [0,L]$. Тогда $\{U(t,x), k(U(t,x)), t\}$ удовлетворяют следующему функционально - интегральному уравнению:*

$$\int\limits_{\Omega} \rho(t,x) U_t(t,x)\, dx = \int\limits_{g(t)}^{h(t)} J_k(s,t,\rho) k(s)\, ds + Q(t;\rho),\ 0 < t < T, \tag{13}$$

в котором ядром является взвешенная поверхностная интегральная характеристика поля концентрации:

$$J_k(s,t,\rho) = \frac{\partial \rho(t,x)}{\partial x} \frac{{}_0D_x^{\alpha}(U) -_x D_L^{\alpha}(U)}{U_x(t,x)}\bigg|_{x=x(t,s)}. \tag{14}$$

Свободный член представляет собой взвешенную поверхностную интегральную характеристику потока массы на границе $\Gamma = \partial\Omega = \{0\} \cup \{L\}$:

$$Q(t;\rho) = \int\limits_{\Gamma} \rho(t,x) q(t,x)\, d\Gamma \equiv \rho(t,L)\cdot q_L(t) - \rho(t,0)\cdot q_0(t). \tag{15}$$

Доказательство. Умножим уравнение (1) на $\rho(t,x)$ и проинтегрируем по области Ω, фиксируя $t \in (0,T)$:

$$\int\limits_0^L \rho(t,x) U_t(t,x)\, dx = \int\limits_0^L \rho(t,x) \frac{\partial}{\partial x}\left[k(U)\cdot\left({}_0D_x^{\alpha}(U) -_x D_L^{\alpha}(U)\right)\right] dx. \tag{16}$$

Рассмотрим интеграл стоящий в правой части:

$$\begin{aligned}\int\limits_0^L \rho(t,x) \frac{\partial}{\partial x}\left[k(U)\cdot\left({}_0D_x^{\alpha}(U) -_x D_L^{\alpha}(U)\right)\right] dx = \\ = \rho(t,x)\left(k(U)\left({}_0D_x^{\alpha}(U) -_x D_L^{\alpha}(U)\right)\right)_x\Big|_0^L - \\ - \int\limits_0^L \frac{\partial \rho(t,x)}{\partial x} k(U)\left({}_0D_x^{\alpha}(U) -_x D_L^{\alpha}(U)\right) dx.\end{aligned} \tag{17}$$

На основании граничного условия (4) получим:

$$\rho(t,x)\left(k(U)\left({}_0D_x^\alpha(U) -_x D_L^\alpha(U)\right)\right)_x \Big|_0^L = Q(t;\rho). \tag{18}$$

Затем введем в интеграл $U_x(t,x)$:

$$\begin{gathered}\int\limits_0^L \frac{\partial \rho(t,x)}{\partial x} k(U)\left({}_0D_x^\alpha(U) -_x D_L^\alpha(U)\right) dx = \\ = \int\limits_0^L \frac{\partial \rho(t,x)}{\partial x} \frac{{}_0D_x^\alpha(U) -_x D_L^\alpha(U)}{U_x} k(U) U_x \, dx.\end{gathered} \tag{19}$$

Учитывая, что

$$k(U)U_x = \frac{\partial}{\partial x} \int\limits_{r(t)}^{U(t,x)} \lambda(s)\, ds, \;\; r = g, h, \tag{20}$$

получим

$$\begin{gathered}\int\limits_0^L \frac{\partial \rho(t,x)}{\partial x} \frac{{}_0D_x^\alpha(U) -_x D_L^\alpha(U)}{U_x} k(U) U_x \, dx = \\ = \int\limits_0^L \frac{\partial \rho(t,x)}{\partial x} \frac{{}_0D_x^\alpha(U) -_x D_L^\alpha(U)}{U_x} \frac{\partial}{\partial x} \int\limits_{r(t)}^{U(t,x)} k(s)\, ds\, dx = \\ = \frac{\partial \rho(t,x)}{\partial x} \frac{{}_0D_x^\alpha(U) -_x D_L^\alpha(U)}{U_x} \int\limits_{r(t)}^{U(t,x)} k(s)\, ds \Big|_0^L - \\ - \int\limits_0^L \frac{\partial}{\partial x}\left[\frac{\partial \rho(t,x)}{\partial x} \frac{{}_0D_x^\alpha(U) -_x D_L^\alpha(U)}{U_x}\right] \int\limits_{r(t)}^{U(t,x)} k(s)\, ds\, dx.\end{gathered} \tag{21}$$

Уравнение в итоге примет вид:

$$\int\limits_0^L \rho(t,x)U_t(t,x)\,dx = Q(t;\rho) - \frac{\partial\rho(t,x)}{\partial x}\frac{{}_0D_x^\alpha(U) -_x D_L^\alpha(U)}{U_x}\int\limits_{r(t)}^{U(t,x)} k(s)\,ds\Big|_0^L + \\ + \int\limits_0^L \frac{\partial}{\partial x}\left[\frac{\partial\rho(t,x)}{\partial x}\frac{{}_0D_x^\alpha(U) -_x D_L^\alpha(U)}{U_x}\right]\int\limits_{r(t)}^{U(t,x)} k(s)\,ds\,dx. \tag{22}$$

Выполним следующую замену:

$$\frac{\partial\rho(t,x)}{\partial x}\frac{{}_0D_x^\alpha(U) -_x D_L^\alpha(U)}{U_x} = \frac{\partial\tilde{\rho}(t,x)}{\partial x}, \tag{23}$$

тогда

$$\int\limits_0^L \rho(t,x)U_t(t,x)\,dx = Q(t;\rho) - \frac{\partial\tilde{\rho}(t,x)}{\partial x}\int\limits_{r(t)}^{U(t,x)} k(s)\,ds\Big|_0^L + \\ + \int\limits_0^L \frac{\partial^2\tilde{\rho}(t,x)}{\partial x^2}\int\limits_{r(t)}^{U(t,x)} k(s)\,ds\,dx. \tag{24}$$

Преобразуем интеграл из (24):

$$\int\limits_0^L \frac{\partial^2\tilde{\rho}(t,x)}{\partial x^2}\int\limits_{r(t)}^{U(t,x)} k(s)\,ds\,dx = \int\limits_0^L\int\limits_{g(t)}^{h(t)} \frac{\partial^2\tilde{\rho}(t,x)}{\partial x^2}\chi(r(t),U(t,x),s)k(s)\,ds\,dx; \tag{25}$$

используя следующее соотношение:

$$\int\limits_0^L \frac{\partial^2\tilde{\rho}(t,x)}{\partial x^2}\chi(r(t),U(t,x),s)\,dx = (-1)^i \int\limits_{\Omega_i(s,t)} \frac{\partial^2\tilde{\rho}(t,x)}{\partial x^2}\,dx,\ i=1,2\,, \tag{26}$$

где $\Omega_1 = [0, x(s,t)], \Omega_2 = [x(s,t), L]$. Получим:

$$\begin{gathered}
\int\limits_0^L \int\limits_{g(t)}^{h(t)} \frac{\partial^2 \tilde{\rho}(t,x)}{\partial x^2} \chi(r(t), U(t,x), s) k(s)\, ds\, dx = \\
= \int\limits_{g(t)}^{h(t)} (-1)^i \int\limits_{\Omega_i(s,t)} \frac{\partial^2 \tilde{\rho}(t,x)}{\partial x^2}\, dx\, k(s)\, ds = \\
= \int\limits_{g(t)}^{h(t)} (-1)^{2i} \int\limits_{\partial\Omega_i(s,t)} \frac{\partial \tilde{\rho}(t,x)}{\partial x}\, d\Gamma\, k(s)\, ds = \int\limits_{g(t)}^{h(t)} \int\limits_{\partial\Omega_i(s,t)} \frac{\partial \tilde{\rho}(t,x)}{\partial x}\, d\Gamma\, k(s)\, ds.
\end{gathered} \tag{27}$$

, где $\partial\Omega_1 = \{0\} \cup \{x(s,t)\}, \Omega_2 = \{x(s,t)\} \cup \{L\}$. В итоге уравнение примет вид:

$$\begin{gathered}
\int\limits_0^L \rho(t,x) U_t(t,x)\, dx = Q(t;\rho) - \frac{\partial \tilde{\rho}(t,x)}{\partial x} \int\limits_{r(t)}^{U(t,x)} k(s)\, ds \Big|_0^L + \\
+ \int\limits_{g(t)}^{h(t)} \int\limits_{\partial\Omega_i(s,t)} \frac{\partial \tilde{\rho}(t,x)}{\partial x}\, d\Gamma\, k(s)\, ds,
\end{gathered} \tag{28}$$

$$\begin{gathered}
\int\limits_0^L \rho(t,x) U_t(t,x)\, dx = Q(t;\rho) - \frac{\partial \tilde{\rho}(t,x)}{\partial x} \int\limits_{g(t)}^{h(t)} \chi(r(t), U(t,x), s) k(s)\, ds \Big|_0^L + \\
+ \int\limits_{g(t)}^{h(t)} \int\limits_{\partial\Omega_i(s,t)} \frac{\partial \tilde{\rho}(t,x)}{\partial x}\, d\Gamma\, k(s)\, ds,
\end{gathered} \tag{29}$$

$$\begin{gathered}
\int\limits_0^L \rho(t,x) U_t(t,x)\, dx = Q(t;\rho) - \int\limits_{g(t)}^{h(t)} \int\limits_{\Gamma} \chi(r(t), U(t,x), s) \frac{\partial \tilde{\rho}(t,x)}{\partial x}\, d\Gamma k(s)\, ds + \\
+ \int\limits_{g(t)}^{h(t)} \int\limits_{\partial\Omega_i(s,t)} \frac{\partial \tilde{\rho}(t,x)}{\partial x}\, d\Gamma\, k(s)\, ds,
\end{gathered} \tag{30}$$

Учитывая, что

$$\int\limits_{\partial\Omega_i(s,t)} \frac{\partial\tilde{\rho}(t,x)}{\partial x}\,d\Gamma - \int\limits_{\Gamma} \chi(r(t),U(t,x),s)\frac{\partial\tilde{\rho}(t,x)}{\partial x}\,d\Gamma = \int\limits_{x=x(s,t)} \frac{\partial\tilde{\rho}(t,x)}{\partial x}\,dx(s,t), \tag{31}$$

приходим к искомому уравнению (13). $\square$

2 Численный алгоритм решения функционально-интегрального уравнения

2.1 Численная схема решения прямой задачи

Численная схема решения уравнения (1) строится с помощью аппроксимации дробной производной по формуле Грюнвальда-Летникова [2]:

$$ {}_0D_x^{\alpha}(U(t,x)) = \frac{1}{\Delta x^{\alpha}} \sum_{j=0}^{\frac{x-0}{\Delta x}} (-1)^j \begin{pmatrix} \alpha \\ j \end{pmatrix} U(t, x - j \cdot \Delta x), \tag{32} $$

$$ {}_xD_L^{\alpha}(U(t,x)) = \frac{1}{\Delta x^{\alpha}} \sum_{j=0}^{\frac{L-x}{\Delta x}} (-1)^j \begin{pmatrix} \alpha \\ j \end{pmatrix} U(t, x + j \cdot \Delta x). \tag{33} $$

Уравнение для $U(x,t)$ имеет вид:

$$ U_t = \left[k(U) \cdot ({}_0D_x^{\alpha}(U) -_x D_L^{\alpha}(U))\right]_x, \tag{34} $$

$$ U(0,x) = f(x), U(t,0) = q_0(t), U(t,L) = q_L(t), x \in (0,L), t \in (0,T). \tag{35} $$

Обозначим: $G_j^{\alpha} = \frac{(-1)^j}{\Delta x^{\alpha}} \cdot \begin{pmatrix} \alpha \\ j \end{pmatrix}$.

Разностная схема будет иметь вид:

$$ \begin{gathered} (U_i^n - U_i^{n-1}) \cdot \frac{\Delta x}{\Delta t} = \\ = k_i^{n-1} \cdot \left[\sum_{j=0}^{i} G_j^{\alpha} U_{i-j}^n - \sum_{j=0}^{N-i} G_j^{\alpha} U_{i+j}^n \right] - \\ -k_{i-1}^{n-1} \cdot \left[\sum_{j=0}^{i-1} G_j^{\alpha} U_{i-j}^n - \sum_{j=0}^{N-i+1} G_j^{\alpha} U_{i+j}^n \right]. \end{gathered} \tag{36} $$

2.2 Метод регуляризации

Полученное в предыдущей главе ФИУ может быть сведено к интегральному уравнению Фредгольма I рода.

Задача отыскания решения такого уравнения является некорректной, поэтому для её решения будет использоваться метод регуляризации А. Н. Тихонова [4]. Уравнение имеет вид:

$$\begin{gathered}\int\limits_{\underline{g}}^{\bar{h}} J_k(s,t;\rho)k(s)\,ds = \int\limits_0^L \rho(t,x)U_t(t,x)\,dx+ \\ + \int\limits_{\Gamma} \rho(t,x)k(U(t,x)\left[{}_0D_x^{\alpha}(U(t,x)) -_x D_1^{\alpha}(U(t,x))\right]\,d\Gamma.\end{gathered} \tag{37}$$

Переобозначим:

$$\begin{gathered}f(t) = \int\limits_0^L \rho(t,x)U_t(t,x)\,dx+ \\ + \int\limits_{\Gamma} \rho(t,x)k(U(t,x)\left[{}_0D_x^{\alpha}(U(t,x)) -_x D_1^{\alpha}(U(t,x))\right]\,d\Gamma.\end{gathered} \tag{38}$$

Уравнение примет вид:

$$\int\limits_{\underline{g}}^{\bar{h}} J_k(s,t;\rho)k(s)\,ds = f(t), \tag{39}$$

Умножим обе части уравнения на $J_k(\xi,t;\rho)$ и проинтегрируем по t:

$$\int\limits_0^T J_k(\xi,t;\rho)\int\limits_{\underline{g}}^{\bar{h}} J_k(s,t;\rho)k(s)\,ds\,dt = \int\limits_0^T J_k(\xi,t;\rho)f(t)\,dt. \tag{40}$$

В таком случае ядро Шмидта и симметризованная правая часть будут выглядеть как:

$$K(\xi, s; \rho) = \int\limits_0^T J_k(\xi, t; \rho) J_k(s, t; \rho)\, dt, \qquad f^*(\xi) = \int\limits_0^T J(\xi, t; \rho) f(t)\, dt, \tag{41}$$

Применяя метод регуляризации приходим к интегральному уравнению Фредгольма II рода:

$$\Theta \cdot k(\xi) + \int\limits_{\underline{g}}^{\bar{h}} K(\xi, s; \rho) k(s)\, ds = f^*(\xi). \tag{42}$$

Данное уравнение считается численно.

Разностные аппроксимации здесь имеют вид:

$$f(t_n) = \sum_{i=0}^{N} \rho_i^n \tfrac{U_i^n - U_i^{n-1}}{\Delta t} \Delta x + \rho_N^n \cdot q_N^n - \rho_0^n \cdot q_0^n, \tag{43}$$

$$J_k(\xi, t_n; \rho) = \frac{\rho_i^n - \rho_{i-1}^n}{U_i^n - U_{i-1}^n} \cdot \left[\sum_{j=0}^{i} G_j^\alpha U_{i-j}^n - \sum_{j=0}^{N-i} G_j^\alpha U_{i+j}^n \right] \Bigg|_{i=}, \tag{44}$$

$$f^*(\xi) = \sum_{j=1}^{M} J_k(\xi, t_n; \rho) f(t_j) \Delta t, \tag{45}$$

$$K(\xi, s; \rho) = \sum_{j=0}^{M} J_k(\xi, t_n; \rho) J_k(s, t_n; \rho) \Delta t. \tag{46}$$

Уравнение в таком случае примет вид:

$$\Theta \cdot k(\xi_l) + \sum_{c=0}^{P} K(\xi_l, s_c; \rho) k(s_c)\, \Delta s = f^*(\xi_l), l \in [0, P], c \in [0, P], \tag{47}$$

где Θ - параметр регуляризации.

3 Вычислительный эксперимент

Численные эксперименты проводились по классической схеме: сначала решалась прямая задача с заданным(истинным) значением $k(U)$ и определялось поле концентрации, которое затем использовалось в качестве априорной информации при решении обратной задачи по описанному выше алгоритму.

Параметры тестовой задачи:

$$\begin{gathered}\alpha = 0.99,\ L = 1,\ T = 0.25,\ \rho(x) = x \cdot (L - x),\ \Theta = 10^{-6} \\ k(U) = U^2,\ U_0(x) = \frac{x}{2},\ q_0(t) = 0,\ q_L(t) = \frac{1}{2\sqrt{(1-2t)}}.\end{gathered} \tag{48}$$

Поле концентрации прямой задачи показано на рис 1. Результаты восстановления коэффициента аномальной диффузии показаны на рис. 2, что говорит о работоспособности алгоритма.

Таким образом предложенный алгоритм может быть использован для восстановления коэффициента диффузии при некоторых частных случаях поля концентрации.

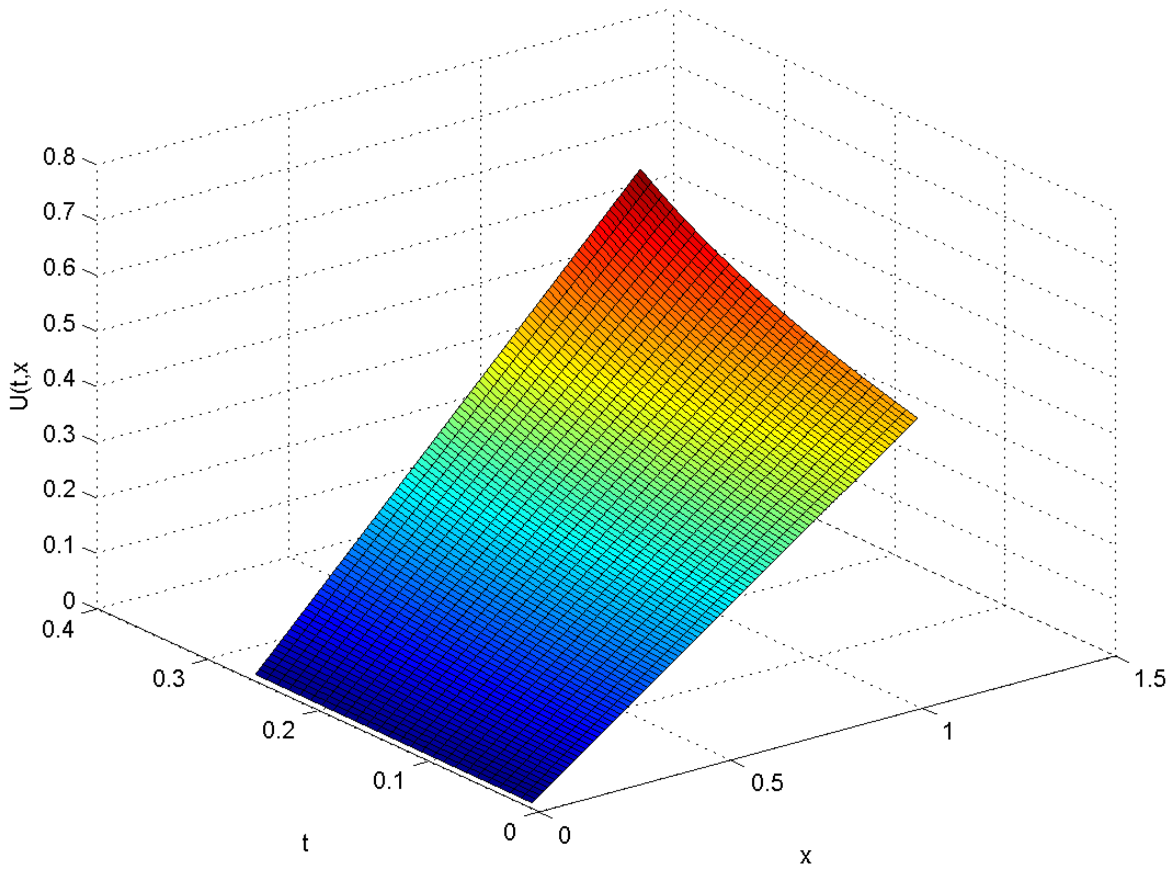

Рис. 1: Решение задачи при истинном значении коэффициента аномальной диффузии.

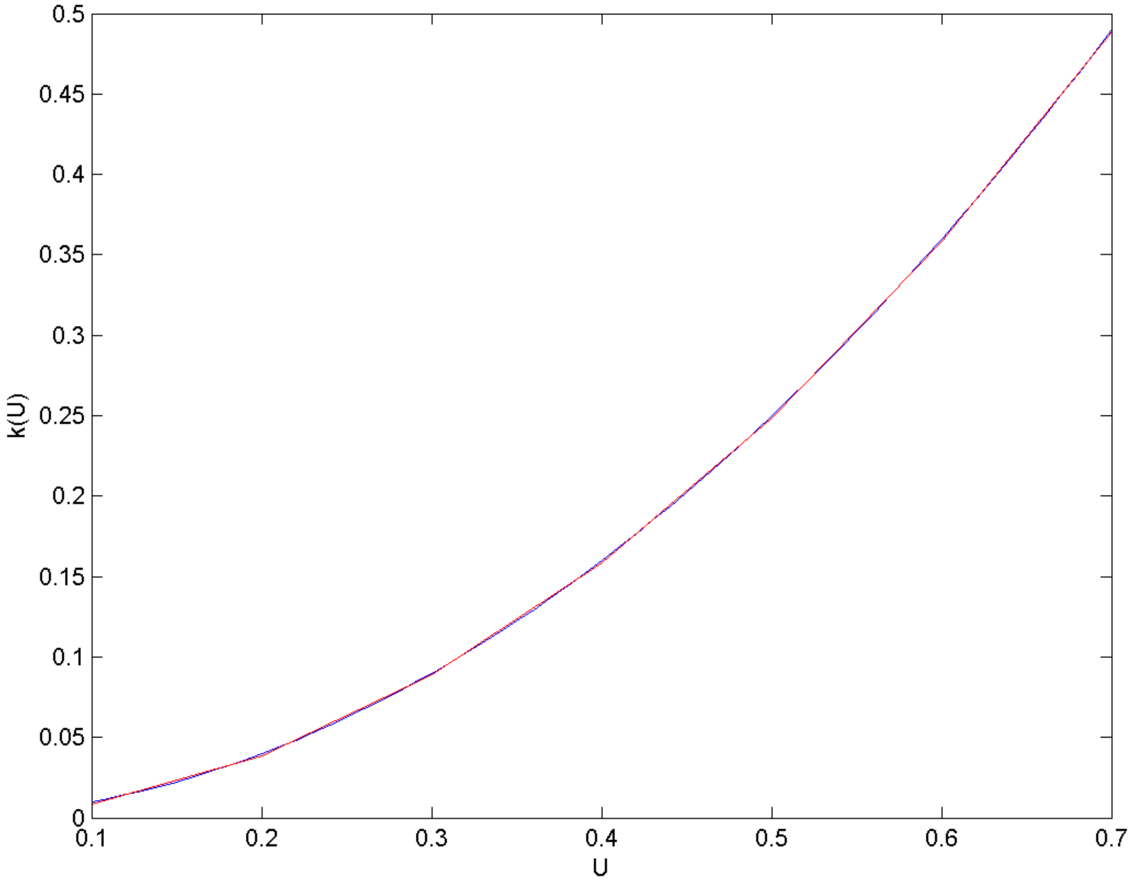

Рис. 2: Результат идентификации k(U)

Заключение

Метод решения коэффициентных обратных задач, основанный на сведении обратной задачи к функционально - интегральному уравнению адаптирован к задаче идентификации коэффициента уравнения аномальной диффузии с лево-и правосторонними дробными производными типа Римана-Лиувилля по пространственной переменной.

Построено функционально-интегральное уравнение относительно коэффициента аномальной диффузии для исследуемого уравнения. Предложен численный алгоритм решения задачи, который реализован программно для некоторых частных случаев функции поля концентрации. С использованием программы были проведены вычислительные эксперименты, подтвердившие работоспособность предложенного алгоритма.

Список литературы

1. Шаталов Ю.С. Функционально-интегральные уравнения теплофизических характеристик. –М.: Наука. Физматлит, 1996.–304 с. –ISBN 5-02-015156-4.

2. Самко С. Г., Килбас А. А., Маричев О. И. Интегралы и производные дробного порядка и некоторые их приложения. - Минск: Наука и техника, 1987-688с.

3. Тихонов А. Н., Арсенин В. Я. Методы решения некорректных задач. — М.: Наука, 1979. — 283 с.

4. Ralf Metzler and Joseph Klafter The random walk‘s quide to anomalous diffusion: a fractional dynamics approach // Phys. Rep. 2000. Vol. 339 P. 1-77

5. Rainer Klages, Gunter Radons, Igor M. Sokolov Anomalous Transport: Foundations and Applications // Willey-VCH. Berlin. 2008.

6. Ralph Metzler, Eli Barkai, Joseph Klafter Anomalous Diffusion and Relaxation Close to Thermal Equilibrium: A Fractional Fokker-Plank Equation Approach // Phys. Rev. Lett. 1999. Vol. 82 Num. 18 P. 3563-3567

7. D. A. Benson, S. W. Wheatcraft, M. M. Meerschaert Application of a fractional advection-dispersion equation // Water Resour. Res. 2000. Vol. 36 Num. 6 P. 1403-1412

8. Eliott W. Montroll, George H. Weiss Random Walks on Lattices II // J. Math. Phys. 1965. Vol. 6 Num. 2 Mon. February P. 167-181

Приложение

Листинг программы

Файл, вычисляющий значение прямой задачи:

```
clear;
init

for n = 2:t_steps
    % A * U = B, U = ?
    A = zeros(x_steps, x_steps);
    U = zeros(x_steps, 1);
    B = zeros(x_steps, 1);

    % fill k
    k = zeros(x_steps, 1);
    for i=1:x_steps
        k(i) = k_coeff(exact_u(n-1,i));
    end

    % prepare right part
    B(1) = 0;
    B(x_steps) = 0;

    for i = 2:x_steps-1
        B(i) = dx * exact_u(n-1, i) / dt;
        B(i) = B(i) - 2 * dx * x_steps_array(i)
            * ( q_b(t_steps_array(n)) - q_b(t_steps_array(n-1)))
            / (b * dt);
        B(i) = B(i) - 2 * dx * (1 - x_steps_array(i) / b)
        * ( q_a(t_steps_array(n)) -
```

```
            q_a(t_steps_array(n - 1)) ) / dt;

        temp = ( q_b(t_steps_array(n)) - q_a(t_steps_array(n)) )
         * x_steps_array(i)^(1-alpha) / (b * gamma(2 - alpha));
        temp = temp + q_a(t_steps_array(n)
            * x_steps_array(i)^(-alpha)) / gamma(1 - alpha);
        temp = temp - ( q_b(t_steps_array(n)) -
             q_a(t_steps_array(n)) ) *
            (b - x_steps_array(i))^(-alpha) *
            ((b - x_steps_array(i+1)) / (1-alpha) - b)
            / (b * gamma(1 - alpha));
        temp = temp - q_a(t_steps_array(n)) *
        (b - x_steps_array(i))^(-alpha) / gamma(1 - alpha);
        B(i) = B(i) + temp * k(i);

        temp = ( q_b(t_steps_array(n)) - q_a(t_steps_array(n)) )
         * x_steps_array(i-1)^(1-alpha)
            / (b * gamma(2 - alpha));
        temp = temp + q_a(t_steps_array(n) *
            x_steps_array(i-1)^(-alpha)) / gamma(1 - alpha);
        temp = temp - ( q_b(t_steps_array(n)) -
            q_a(t_steps_array(n)) ) *
            (b - x_steps_array(i-1))^(-alpha) *
            ((b - x_steps_array(i-1)) / (1-alpha) - b)
                / (b * gamma(1 - alpha));
        temp = temp - q_a(t_steps_array(n)) * (b
            - x_steps_array(i-1))^(-alpha) / gamma(1 - alpha);
        B(i) = B(i) - temp * k(i-1);
    end

    %prepare first and last rows of matrix
    A(1,1) = 1;
```

```
    A(x_steps, x_steps) = 1;
    for i = 2:x_steps-1
        A(1,i) = 0;
        A(x_steps, i) = 0;
    end

    %filling the matrix
    for i = 2:x_steps-1 % row
        A(i,i) = A(i,i) + 2 * dx / dt;

        for m = 0:i - 1%i
            num = i - m;%i + 1 - m;
            A(i,num) = A(i,num) - k(i) * grunvald_coeffs(m+1);
        end
        for m = 0:x_steps - i%x_steps - (i + 1)
            num = i + m;%i + 1 + m;
            %if i ~= x_steps - 1
                A(i,num) = A(i,num) + k(i) * grunvald_coeffs(m+1);
            %end
        end
        for m = 0:i - 2
            num = i - 1 - m;
            if i ~= 2
                A(i,num) = A(i,num) + k(i-1) * grunvald_coeffs(m+1);
            end
        end
        for m = 0:x_steps - (i - 1)
            num = i - 1 + m;
            A(i,num) = A(i,num) - k(i-1) * grunvald_coeffs(m+1);
        end
    end
    U = A\B;
```

```
    for i = 1:x_steps
        exact_u(n,i) = U(i);
    end
end

for i = 1:t_steps
    for j = 1:x_steps
        exact_u(i,j) = exact_u(i,j) + x_steps_array(j)
    * q_b(t_steps_array(i)) / b +
                (1 - x_steps_array(j) / b)
                * q_a(t_steps_array(i));
    end
end

%figure
%surf(exact_u);
```

Правая часть для интегрального уравнения:

```
% right part function, n from 2 to t_steps
function f_res = f_right_part(n)
    global x_steps;
    global exact_u;
    global dx;
    global dt;
    global t_steps_array;

    f_res = 0;
    for i = 1:x_steps
        f_res = f_res + rho(i) * (exact_u(n, i) -
            exact_u(n - 1, i));
    end
```

```

    f_res = f_res * dx / dt;

    f_res = f_res - rho(x_steps) * q_b(t_steps_array(n));
    f_res = f_res + rho(1) * q_a(t_steps_array(n));
end
```

Симметризованная правая часть для интегрального уравнения:

```
% symmetrized right part function, i from 2 to x_steps
function f_right_part_symm_res = f_right_part_symm(i)
    global dx;
    global t_steps;

    f_right_part_symm_res = 0.;
    for j = 2:t_steps
        f_right_part_symm_res = f_right_part_symm_res
         + f_right_part(j) * J_k(j, i);
    end
    f_right_part_symm_res = f_right_part_symm_res * dx * (-1);
end
```

Расчет коэффициентов в разложении Грювальда-Летникова:

```
function coeff = gr_coeff(n, alpha, dx)
    coeff = ones(1, n);
    coeff(1) = 1;
    for i = 2:n
        coeff(i) = coeff(i-1) * (alpha - (i - 1) + 1) / (i - 1);
    end
    for i = 1:n
```

```
            coeff(i) = coeff(i) * (-1)^(i-1) / dx^alpha;
        end
end
```

Инициализация глобальных переменных и сетки:

```
clear;
%fractional order
global alpha; alpha = 0.99;

%space
global a; a = 0;
global b; b = 1;
global dx; dx = 0.01;
global x_steps; x_steps = (b - a) / dx + 1;

global x_steps_array;
x_steps_array = ones(1, x_steps);
for i = 1:x_steps
    x_steps_array(i) = dx * (i - 1);
end

%time
global t0; t0 = 0;
global t1; t1 = 0.25;
global dt; dt = 0.01;
global t_steps; t_steps = (t1 - t0) / dt + 1;

global t_steps_array;
t_steps_array = (t1 - t0) / dt + 1;
for i = 1:t_steps
     t_steps_array(i) = dt * (i - 1);
```

```
end

% starting and border conditions
global exact_u;
exact_u = zeros(t_steps, x_steps);
for i = 1:x_steps
    exact_u(1, i) = start_subs(x_steps_array(i));
end
for i = 1:t_steps
    exact_u(i, 1)       = q_a_subs(t_steps_array(i));
    exact_u(i, x_steps) = q_b_subs(t_steps_array(i));
end

% filling grunvald-letnikov coefficents
global grunvald_coeffs;
grunvald_coeffs = gr_coeff(x_steps, alpha, dx);
```

Ядро функционально-интегрального уравнения:

```
% core function, i from 2 to x_steps, n from 2 to t_steps
function J_k_res = J_k(n, i)
    global exact_u;
    global grunvald_coeffs;
    global x_steps;

    J_k_res = (rho(i) - rho(i-1)) / (exact_u(n, i) -
        exact_u(n, i-1));

    temp_0_i = 0;
    if i ~= 1
        for m = 0:i - 1
            num = i - m;
```

```
                    temp_0_i = temp_0_i + grunvald_coeffs(m+1)
                    * exact_u(n, num);
            end
        end

        temp_i_N = 0;
        if i ~= x_steps
            for m = 0:x_steps - i
                num = i + m;
                temp_i_N = temp_i_N + grunvald_coeffs(m+1) *
                  exact_u(n, num);
            end
        end
        J_k_res = J_k_res * (temp_0_i - temp_i_N);
end
```

Симметризованное ядро функционально-интегрального уравнения:

```
function J_k_symm_res = J_k_symm(i1, i2)
    J_k_symm_res = 0.;
    global t_steps;
    global dt;
    for k = 2:t_steps
        J_k_symm_res = J_k_symm_res + J_k(k, i1)
    * J_k(k, i2);
    end
    J_k_symm_res = J_k_symm_res * dt;
end
```

Функция расчета коэффициента диффузии:

```
function k_res = k(u)
    k_res = u * u;
end
```

Граничное условие слева:

```
function q_a = q_a(t)
    q_a = 0;
end
```

Граничное условие слева для задачи с заменой:

```
function q_a_subs = q_a_subs(t)
    q_a_subs = 0;
end
```

Граничное условие справа:

```
function q_b = q_b(t)
    q_b = 1. / (2 * sqrt(1 - 2 * t));
end
```

Граничное условие справа для задачи с заменой:

```
function q_b_subs = q_b_subs(t)
    q_b_subs = 0;
end
```

Весовая функция:

```
function rho_res = rho(i)
    global a;
    global b;
    global x_steps_array;
    rho_res = (x_steps_array(i) - 0)
    * (1 - x_steps_array(i));
end
```

Решение задачи для $\alpha = 1$:

```
init

U_99 = zeros(t_steps,x_steps);

for i= 1:t_steps
    for j=1:x_steps
        U_99(i,j) = (j-1) * dx
    / (2 * sqrt(1 - 2 * (i-1) * dt));
    end
end

figure
surf(U_99);
```

Решение задачи регуляризации:

```
count_exact
sample

newton_alpha = 1;

```

```
min_U = min(min(U_99));
max_U = max(max(U_99));
dU = 0.001;
U_steps = (max_U - min_U) / dU + 1;


A_REG = zeros(x_steps, x_steps);
K_REG = zeros(x_steps, 1);
B_REG = zeros(x_steps, 1);

for i = 1:x_steps
    B_REG(i) = f_right_part_symm(i);
end

for i = 2:x_steps
    for j = 2:x_steps
        A_REG(i-1,j-1) = A_REG(i-1,j-1)
    + J_k_symm(i,j) * (exact_u(t_steps - 2,j)
            - exact_u(t_steps - 2,j-1));
    end
    A_REG(i-1,i-1) = A_REG(i-1,i-1) + newton_alpha;
end

K_REG = A_REG \ B_REG;

figure
k_space=[min_exact:0.01:max_exact];
y = zeros(length(k_space));
for i = 1:length(k_space)
    y(i) = k_space(i) * k_space(i);
end
plot(k_space, y)
```

Начальное условие:

```
function f = start(x)
    f = x / 2.;
end
```

Начальное условие для задачи с заменой:

```
function f_subs = start_subs(x)
    global b;
    f_subs = start(x) - x / b
    * start(b) - (1 - x / b) * start(0);
end
```