Задание

на курсовую работу по дисциплине

«Методы оптимизации»

Студент: группы ПМ-435 Гребельник Михаил Степанович

Консультант: Лукащук Станислав Юрьевич

1. Тема задания

Оптимизация численной неявной схемы и разработка параллельного алгоритма для исследования уравнения Кортевега-де-Фриза дробного порядка.

2. Основное содержание

2.1. Разработать численный алгоритм на основе неявной конечно-разностной схемы, позволяющий проводить компьютерное моделирование течение жидкости, описываемое начально-краевой задачей для уравнения Кортевега-де-Фриза дробного порядка.

2.2. Написать программу на языке C++, реализующую разработанный алгоритм. Разработать параллельный алгоритм. Оптимизировать программу, используя средства OpenMP.

2.3. С использованием программы произвести серию вычислительных экспериментов.

3. Требования к оформлению материалов работы

Пояснительная записка к заданию должна быть оформлена в текстовом редакторе Microsoft Word в соответствии с требованиями стандартами СТО УГАТУ 016 2007 и содержать:

- Титульный лист
- Задание на учебную практику
- Содержание
- Постановка задачи
- Алгоритм численного моделирования
- Результаты вычислительных экспериментов
- Заключение
- Список литературы

4. Список рекомендуемой литературы

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

4.2 A.A. Kilbas, H.M. Srivastava, J.J. Trujillo Theory and applications of fractional differential equations. North-Holland Mathematics Studies 204. 2006 г. 523с.

4.3 Igor Podlubny. Fractional differential equations. Academic press, 1999. 340 с.

Дата выдачи задания 1 октября 2013 г.

Дата окончания 1 января 2013 г.

Консультант _____ Лукащук С.Ю.

Принял к исполнению _____ Гребельник М. С.

Содержание

1.Постановка задачи

Разработать численный алгоритм, позволяющий проводить компьютерное моделирование волновых процессов, описываемое следующей начально-краевой задачей для уравнения Кортевега- де- Фриза дробного порядка:

$$D_t^{\alpha} U + U \cdot J_t^{1-\alpha}(U_x) = aU_{xxx}, \alpha \epsilon (0,1), a = const; \quad (1)$$

$$J_t^{1-\alpha} U|_{t=0} = f(x), U_x|_{x=0} = U_x|_{x=1} = U_{xx}|_{x=1} = 0, \quad (2)$$

где $U = U(x,t)$- искомая функция.

В уравнении (1) выражения

$$J_t^{1-\alpha} U = \frac{1}{\Gamma(1-\alpha)} \int_0^t \frac{U(x,\tau)}{(t-\tau)^{\alpha}} d\tau, \quad D_t^{\alpha} U = \frac{1}{\Gamma(1-\alpha)} \frac{\partial}{\partial t} \int_0^t \frac{U(x,\tau)}{(t-\tau)^{\alpha}} d\tau, \quad (3)$$

(3) называются дробным интегралом порядка 1-α и дробной производной порядка 1-α , $0<\alpha<1$ соответственно. Дробные производные и дробные интегралы в приведенном виде называют обычно производными и интегралами Римана — Лиувилля.

В (3) через $\Gamma(z)$ выражается гамма функция:

$$\Gamma(z) = \int_0^{+\infty} t^{z-1} e^{-t} dt, \quad \Gamma(z+1) = z\Gamma(z).$$

2.Алгоритм численного моделирования

2.1 Преобразование уравнения

Рассмотрим начальное условие из (2). Справедливо утверждение[2]:

$$\lim_{t\to 0} \; J_t^{1-\alpha} U = \lim_{t\to 0} \frac{U(x,t)\Gamma(\alpha)}{t^{\alpha-1}} = f(x),$$

Рассмотрим дробный интеграл от функции U:

$$J_t^{1-\alpha} U \equiv \frac{1}{\Gamma(1-\alpha)} \int_0^t \frac{U(x,\tau)}{(t-\tau)^{\alpha}} d\tau =$$

$$= \frac{1}{\Gamma(1-\alpha)} \int_0^s \frac{U(x,\tau)}{(t-\tau)^{\alpha}} d\tau + \frac{1}{\Gamma(1-\alpha)} \int_s^t \frac{U(x,\tau)}{(t-\tau)^{\alpha}} d\tau =$$

$$= \frac{1}{\Gamma(1-\alpha)} \int_0^s \frac{U(x,\tau)}{(t-\tau)^{\alpha}} d\tau + \; {}_sJ_t^{1-\alpha} U.$$

Тогда дробная производная от функции $U(x,t)$ может быть записана в виде

$$D_t^{\alpha} U \equiv \frac{1}{\Gamma(1-\alpha)} \frac{\partial}{\partial t} \int_0^s \frac{U(x,\tau)}{(t-\tau)^{\alpha}} d\tau + \; {}_sD_t^{1-\alpha} U.$$

Распишем подробнее первое слагаемое, обозначив его за I, найдя частную производную

$$I = \frac{-\alpha}{\Gamma(1-\alpha)} \int_0^s \frac{U(x,\tau)}{(t-\tau)^{\alpha+1}} d\tau = \frac{1}{\Gamma(-\alpha)} \int_0^s \frac{U(x,\tau)}{(t-\tau)^{\alpha+1}} d\tau.$$

При $0 < t < s$ считаем, что $U(t,x) = \dfrac{f(x)t^{\alpha-1}}{\Gamma(\alpha)}$.

Тогда для I получаем

$$I = \frac{1}{\Gamma(-\alpha)} \int_0^s \frac{f(x)t^{\alpha-1}}{\Gamma(\alpha)(t-\tau)^{\alpha+1}} d\tau = -\frac{\alpha \sin \pi\alpha}{\pi} f(x) \int_0^s \frac{\tau^{\alpha-1}}{(t-\tau)^{\alpha+1}} d\tau,$$

так как $\Gamma(\alpha)\Gamma(-\alpha) = -\dfrac{\pi}{\alpha \sin \pi\alpha}$.

Далее,

$${}_sD_t^{-\alpha} U = \; {}_sD_t^{-\alpha}\big(U - U(s)\big) + \; {}_sD_t^{-\alpha} U(s).$$

Выполним следующую замену зависимой переменной:

$$U - U(s) = V(x,t;s) \Rightarrow U = U(s) + V.$$

Так как $U(s)$ не зависит от t, то справедливо

$${}_sD_t^{\alpha} U(s) = U(s) \cdot \; {}_sD_t^{\alpha}(1) = U(s) \cdot \frac{(t-s)^{-\alpha}}{\Gamma(1-\alpha)},$$

$$D_t^{\alpha} U(s) = I(t;s) + U(s) \cdot \frac{(t-s)^{-\alpha}}{\Gamma(1-\alpha)} + \; {}_sD_t^{\alpha} V.$$

После замены уравнение (1) примет вид

$$ {}_sD_t^\alpha V + I(t;s) + U(s)\frac{(t-s)^{-\alpha}}{\Gamma(1-\alpha)} + \big(V+U(s)\big)\,{}_0J_t^{1-\alpha}(U_x(s)+V_x) = $$
$$ = a(U_{xxx}(s)+V_{xxx}). $$

Распишем подробнее выражение $J_t^{1-\alpha}U_x$:

$$ J_t^{1-\alpha}U_x == \frac{1}{\Gamma(1-\alpha)}\int_0^s \frac{U_x(x,\tau)}{(t-\tau)^\alpha}d\tau + {}_sJ_t^{1-\alpha}(U_x(s)+V_x) = $$
$$ = \frac{1}{\Gamma(1-\alpha)}\frac{f'(x)}{\Gamma(\alpha)}\int_0^s \frac{\tau^{1-\alpha}}{(t-\tau)^\alpha}d\tau + U_x(s)\ {}_sJ_t^{1-\alpha}(1) + {}_sJ_t^{1-\alpha}V_x = $$
$$ = \frac{\sin\pi\alpha}{\pi}f'(x)\frac{s^\alpha}{t^\alpha}\frac{1}{\alpha}\ {}_2F_1\ \left(\alpha,\alpha,\alpha+1;\frac{s}{t}\right) $$
$$ + U_x(s)\frac{(t-s)^{1-\alpha}}{\Gamma(2-\alpha)} + {}_sJ_t^{1-\alpha}V_x, $$

где

$$ \Gamma(\alpha)\cdot\Gamma(1-\alpha) = \frac{\pi}{\sin(\pi\alpha)}, $$

и ${}_2F_1\ \left(\alpha,\alpha,\alpha+1;\frac{s}{t}\right)$ - гипергеометрическая функция.

Перенесем дробную производную в левую часть, а остальные слагаемые – вправо:

$$ {}_sD_t^\alpha V - \alpha\frac{\sin\pi\alpha}{\pi}f(x)\frac{s^\alpha}{t^{1+\alpha}}\frac{1}{\alpha}\frac{t^\alpha}{(t-s)^\alpha} + U(x;s)\frac{(t-s)^{-\alpha}}{\Gamma(1-\alpha)} $$
$$ + \big(V+U(x;s)\big)\left[\frac{\sin\pi\alpha}{\pi\alpha}f'(x)\frac{s^\alpha}{t^\alpha}\ {}_2F_1\ \left(\alpha,\alpha,\alpha+1;\frac{s}{t}\right)\right. $$
$$ \left. + U_x(x;s)\frac{(t-s)^{1-\alpha}}{\Gamma(2-\alpha)} + {}_sJ_t^{1-\alpha}V_x\right] - a(U_{xxx}(x;s)+V_{xxx}) = 0, $$

где

$$ U(x;s) = \frac{f(x)s^{\alpha-1}}{\Gamma(\alpha)}. $$

В итоге, уравнение принимает вид:

$$ {}_sD_t^\alpha V - \frac{\sin\pi\alpha}{\pi}f(x)\frac{s^\alpha}{t}\frac{1}{(t-s)^\alpha} + \frac{s^{\alpha-1}}{(t-s)^{1-\alpha}}\frac{1}{\Gamma(1-\alpha)\Gamma(\alpha)}f(x) $$
$$ + \left(V+\frac{f(x)s^{\alpha-1}}{\Gamma(\alpha)}\right)\left[\frac{\sin\pi\alpha}{\pi\alpha}f'(x)\frac{s^\alpha}{t^\alpha}\ {}_2F_1\ \left(\alpha,\alpha,\alpha+1;\frac{s}{t}\right)\right. $$
$$ \left. + \frac{f'(x)s^{\alpha-1}}{\Gamma(\alpha)} + {}_sJ_t^{1-\alpha}V_x\right] - a\left(\frac{f'''(x)s^{\alpha-1}}{\Gamma(\alpha)} + V_{xxx}\right) = $$
$$ = 0. \qquad (*) $$
$$ J_t^{1-\alpha}U|_{t=0} = f(x), U|_{x=0} = g(t) \qquad (**) $$

2.2 Численная схема

Введем равномерную пространственно - временную сетку:

$$x_i = i\Delta x\,,\ i = \overline{0, I}\,,$$

$$t_1 = s\,, \qquad t_n = s + (n-1)\Delta t, \qquad t_{n+1} = s + n\Delta t\,.$$

Обозначим через искомую сеточную функцию

$$V(x,t) \to V(x_i, t_n) = V_i^n.$$

Аппроксимируем дробную производную по формуле Грюнвальда-Летникова [3]:

$${}_sD_t^\alpha V = \frac{1}{(\Delta t)^\alpha}\sum_{j=0}^{\left[\frac{t-s}{\Delta t}\right]}(-1)^j\binom{\alpha}{j}V(x, t-j\Delta t).$$

Запишем для начально-краевой задачи (*), (**) явную конечно-разностную схему:

$$\begin{aligned}
\left({}_sD_t^\alpha V\right)\Big|_{t=t_{n+1}} &- \frac{\sin\pi\alpha}{\pi}f(x)\frac{s^\alpha}{t_{n+1}}\frac{1}{(t_{n+1}-s)^\alpha} + \frac{s^{\alpha-1}}{(t_{n+1}-s)^\alpha}\,\frac{\sin\pi\alpha}{\pi}f(x) \\
&+ \frac{1}{2}\left\{\left(V|_{t=t_{n+1}} + \frac{f(x)s^{\alpha-1}}{\Gamma(\alpha)}\right)\left[\frac{\sin\pi\alpha}{\pi\alpha}f'(x)\frac{s^\alpha}{t_n^\alpha}\ {}_2F_1\ \left(\alpha,\alpha,\alpha+1;\frac{s}{t_n}\right)\right.\right. \\
&\left.\left. + \frac{f'(x)s^{\alpha-1}}{\Gamma(\alpha)} + \ {}_sJ_{t_n}^{1-\alpha}V_x\right] + \left(V|_{t=t_n} + \frac{f(x)s^{\alpha-1}}{\Gamma(\alpha)}\right)\right. \\
&\left.\times\left[\frac{\sin\pi\alpha}{\pi\alpha}f'(x)\frac{s^\alpha}{t_{n+1}^\alpha}\ {}_2F_1\ \left(\alpha,\alpha,\alpha+1;\frac{s}{t_{n+1}}\right) + \frac{f'(x)s^{\alpha-1}}{\Gamma(\alpha)}\right.\right. \\
&\left.\left. + \ {}_sJ_{t_{n+1}}^{1-\alpha}V_x\right]\right\} - aV_{xxx}|_{t=t_{n+1}} - a\frac{f'''(x)s^{\alpha-1}}{\Gamma(\alpha)} = 0.
\end{aligned}$$

Для $t = t_{n+1}$, ${}_sD_t^\alpha V$ примет вид

$$\begin{aligned}
\left({}_sD_t^\alpha V\right)|_{t=t_{n+1}} &= \frac{1}{\Delta t^\alpha}\sum_{j=0}^{\left[\frac{s+n\cdot\Delta t-s}{\Delta t}\right]}(-1)^j\binom{\alpha}{j}V(x, s+n\Delta t-j\Delta t) = \\
&= \frac{1}{\Delta t^\alpha}V(x, s+n\Delta t) + \frac{1}{\Delta t^\alpha}\sum_{j=0}^{n}(-1)^j\binom{\alpha}{j}V(x, s+(n-j)\Delta t)\,.
\end{aligned}$$

Расчетная формула примет вид

$$V_i^{n+1} + \sum_{j=0}^{n} (-1)^j \binom{\alpha}{j} V_i^{n-j+1} - \frac{\sin \pi\alpha}{\pi} \frac{s^\alpha f_i}{(s+n\Delta t)(n)^\alpha} + \frac{s^{\alpha-1}}{(n)^\alpha} \frac{\sin \pi\alpha}{\pi} f_i$$
$$+ \frac{\Delta t^\alpha}{2} \left(V_i^{n+1} + \frac{f_i s^{\alpha-1}}{\Gamma(\alpha)} \right)$$
$$\times \left[\frac{\sin \pi\alpha}{\pi\alpha} f_i' \frac{s^\alpha}{(s+(n-1)\Delta t)^\alpha} \; {}_2F_1 \left(\alpha, \alpha, \alpha+1; \frac{s}{t_n} \right) + \frac{f_i' s^{\alpha-1}}{\Gamma(\alpha)} \right.$$
$$\left. + \; {}_sJ_{t_n}^{1-\alpha} \left(\frac{V_{i+1}^n - V_i^n}{2\Delta x} \right) \right] + \frac{\Delta t^\alpha}{2} \left(V_i^n + \frac{f_i s^{\alpha-1}}{\Gamma(\alpha)} \right)$$
$$\times \left[\frac{\sin \pi\alpha}{\pi\alpha} f_i' \frac{s^\alpha}{(s+n\Delta t)^\alpha} \; {}_2F_1 \left(\alpha, \alpha, \alpha+1; \frac{s}{t_{n+1}} \right) + \frac{f_i' s^{\alpha-1}}{\Gamma(\alpha)} \right.$$
$$\left. + \; {}_sJ_{t_{n+1}}^{1-\alpha} \left(\frac{V_{i+1}^{n+1} - V_{i-1}^{n+1}}{2\Delta x} \right) \right] - a\Delta t^\alpha \frac{V_{i+2}^n - 3V_{i+1}^n + 3V_i^n - V_{i-1}^n}{(\Delta x)^2}$$
$$- a \frac{f_i''' s^{\alpha-1}}{\Gamma(\alpha)} = 0,$$
$$n = 1,2,3, \dots, i = \overline{1, I-1}. \quad (4)$$

2.3 Аппроксимация на сетке

Начальные условия:

$$V_i^0 = 0, V_i^1 = 0, V_i^{n-2} = 0, V_i^{n-1} = 0, V_i^n = 0, i = \overline{0, I}. \quad (5)$$

Граничное условие:

$$V_0^{n+1} = g(t_{n+1}) - \frac{f(0) s^{\alpha-1}}{\Gamma(\alpha)}, V_I^{n+1} = V_{I-1}^{n+1} = V_{I-2}^{n+1}, V_0^{n+1} = V_1^{n+1}, n = $$
$$= 1,2,3, \dots \quad (6)$$

Функция $f(x)$ в численной схеме будет иметь вид

$$f_i = f(x_i), f_i' = \frac{f_{i+1} - f_{i-1}}{2\Delta x}, f_i'' = \frac{f_{i+1} - 2f_i + f_{i-1}}{\Delta x^2}, f_i''' =$$
$$= \frac{1}{2} \left(\frac{f_{i+2} - 3f_{i+1} + 3f_i - f_{i-1}}{\Delta x^3} + \frac{f_{i+1} - 3f_i + 3f_{i-1} - f_{i-2}}{\Delta x^3} \right).$$

Распишем дробный интеграл на слое t_n:

$${}_sJ_{t_n}^{1-\alpha} V = \frac{1}{\Gamma(1-\alpha)} \int_s^{t_n} \frac{V(x,\tau)}{(t_n-\tau)^\alpha} d\tau = \frac{1}{\Gamma(1-\alpha)} \sum_{j=1}^{n-1} \int_{t_j}^{t_{j+1}} \frac{V(x,\tau)}{(t_n-\tau)^\alpha} d\tau.$$

На интервале $[t_j, t_{j+1}]$ будем использовать линейную интерполяцию:

$$\frac{V(x,\tau) - V(x,t_j)}{V(x,t_{j+1}) - V(x,t_j)} = \frac{\tau - t_j}{\Delta t},$$

Откуда:

$$V(x,\tau) = V(x,t_j) + \frac{\tau - t_j}{\Delta t} [V(x,t_{j+1}) - V(x,t_j)].$$

Получаем:

$$ {}_sJ_{t_n}^{1-\alpha}(V_{i+1}) = \frac{1}{\Gamma(1-\alpha)} \sum_{j=1}^{n-1} \int_{t_j}^{t_{j+1}} \frac{V_{i+1}^{j} + \frac{\tau - t_j}{\Delta t}\left[V_{i+1}^{j+1} - V_{i+1}^{j}\right]}{(t_n - \tau)^{\alpha}} d\tau = $$

$$ = \frac{1}{\Gamma(1-\alpha)} \sum_{j=1}^{n-1} \left(\int_{t_j}^{t_{j+1}} \frac{V_{i+1}^{j} - \frac{t_j}{\Delta t}\left[V_{i+1}^{j+1} - V_{i+1}^{j}\right]}{(t_n - \tau)^{\alpha}} d\tau + \right. $$

$$ \left. + \frac{1}{\Delta t} \int_{t_j}^{t_{j+1}} \frac{\tau\left[V_{i+1}^{j+1} - V_{i+1}^{j}\right]}{(t_n - \tau)^{\alpha}} d\tau \right). $$

Распишем дробный интеграл на слое t_{n+1}:

$$ {}_sJ_{t_{n+1}}^{1-\alpha}(V_{i+1}) = \frac{1}{\Delta t\, \Gamma(1-\alpha)} \sum_{j=1}^{n} \int_{t_j}^{t_{j+1}} \frac{\Delta t V_{i+1}^{j} + (\tau - t_j)\left[V_{i+1}^{j+1} - V_{i+1}^{j}\right]}{(t_{n+1} - \tau)^{\alpha}} d\tau = $$

$$ = \frac{1}{\Delta t\, \Gamma(1-\alpha)} \sum_{j=1}^{n} \int_{t_j}^{t_{j+1}} \left(\frac{(\tau - t_j) V_{i+1}^{j+1}}{(t_{n+1} - \tau)^{\alpha}} + \frac{(\Delta t - \tau + t_j) V_{i+1}^{j}}{(t_{n+1} - \tau)^{\alpha}} \right) d\tau $$

$$ = \frac{1}{\Delta t\, \Gamma(1-\alpha)} \int_{t_n}^{t_{n+1}} \frac{(\tau - t_n) V_{i+1}^{n+1}}{(t_{n+1} - \tau)^{\alpha}} d\tau + \frac{1}{\Delta t\, \Gamma(1-\alpha)} $$

$$ \times \sum_{j=1}^{n-1} \int_{t_j}^{t_{j+1}} \frac{(\tau - t_j) V_{i+1}^{j+1}}{(t_{n+1} - \tau)^{\alpha}} d\tau $$

$$ + \frac{1}{\Delta t\, \Gamma(1-\alpha)} \sum_{j=1}^{n} \int_{t_j}^{t_{j+1}} \frac{(\Delta t - \tau + t_j) V_{i+1}^{j}}{(t_{n+1} - \tau)^{\alpha}} d\tau. $$

2.4Описание метода прогонки

Так как численная схема неявная и матрица системы является пятидиагональной, то для ее решения необходимо использовать метод прогонки.

Система уравнений $Ax = F$ равносильна соотношению

$$ c_0 y_0 - d_0 y_1 + e_0 y_2 = f_0, i = 0, \quad (7) $$

$$ -b_1 y_0 + c_1 y_1 - d_1 y_2 + e_1 y_3 = f_1, i = 1, \quad (8) $$

$$ a_i y_{i-2} - b_i y_{i-1} + c_i y_i - d_i y_{i+1} + e_i y_{i+2} = f_i, 2 \le i \le N-2, \quad (9) $$

$$ a_{N-1} y_{N-3} - b_{N-1} y_{N-2} + c_{N-1} y_{N-1} - d_{N-1} y_N = f_{N-1}, i = N-1, \quad (10) $$

$$ a_N y_{N-2} - b_N y_{N-1} + c_N y_N = f_N, i = N. \quad (11) $$

Алгоритм состоит в следующем:

1) по формулам

$$ \alpha_{i+1} = \frac{1}{\Delta_i}[d_i + \beta_i(a_i \alpha_{i-1} - b_i)], i = 2, 3, \dots, N-1, \quad (12) $$

$$ \alpha_1 = \frac{d_0}{c_0}, \alpha_2 = \frac{1}{\Delta_1}(d_1 - \beta_1 b_1), $$

$$ \gamma_{i+1} = \frac{1}{\Delta_i}[f_i - a_i \gamma_{i-1} - \gamma_i(a_i \alpha_{i-1} - b_i)], i = 2, 3, \dots, N, \quad (13) $$

$$ \gamma_1 = \frac{f_0}{c_0}, \gamma_2 = \frac{1}{\Delta_1}(f_1 - b_1 \gamma_1), $$

$$\beta_{i+1} = \frac{e_i}{\Delta_i}, i = 1, 2, \dots, N-2, \qquad (14)$$
$$\beta_1 = \frac{e_0}{c_0},$$

где

$$\Delta_i = c_i - a_i\beta_{i-1} + \alpha_i(a_i\alpha_{i-1} - b_i), 2 \le i \le N, \qquad (15)$$
$$\Delta_1 = c_1 - b_1\alpha_1,$$

находятся прогоночные коэффициенты α_i, β_i и γ_i;

2) неизвестные y_i находятся последовательно по формулам

$$y_i = \alpha_{i+1}y_{i+1} - \beta_{i+1}y_{i+2} + \gamma_{i+1}, i = N-2, N-3, \dots, 0 \qquad (16)$$
$$y_{N-1} = \alpha_N y_N + \gamma_N, y_N = \gamma_{N+1}$$

Это алгоритм монотонной прогонки.

2.5 Реализация метода прогонки

Положим что

$a_i = \mathrm{V}_{\mathrm{i-2}}^{\mathrm{n+1}}, b_i = \mathrm{V}_{\mathrm{i-1}}^{\mathrm{n+1}}, c_i = \mathrm{V}_{\mathrm{i}}^{\mathrm{n+1}}, d_i = \mathrm{V}_{\mathrm{i+1}}^{\mathrm{n+1}}, e_i = \mathrm{V}_{\mathrm{i+2}}^{\mathrm{n+1}}$ $i = \overline{0, I}$.

За f берутся слагаемые, не попавшие в a, b, c, d, e.

Тогда в уравнении (4) приведем подобные при a, b, c, d, e ,f:

a: $\dfrac{-a\Delta t^{\alpha}}{2(\Delta x)^3}$;

$$b: \frac{-a\Delta t^{\alpha}}{2(\Delta x)^3} + \frac{\Delta t^{\alpha}}{2}\left(V_i^n + \frac{f_i s^{\alpha-1}}{\Gamma(\alpha)}\right)\frac{-1}{2\Delta x\Delta t\,\Gamma(1-\alpha)}\left[\frac{(t-\tau)^{1-\alpha}(\alpha\tau - t - \tau)}{(\alpha-2)(\alpha-1)}\right]_{j\Delta t}^{(j+1)\Delta t} - t_n\left[\frac{(t-\tau)^{1-\alpha}}{\alpha-1}\right]_{j\Delta t}^{(j+1)\Delta t}\Bigg];$$

$$c: 1 + \frac{2a\Delta t^{\alpha}}{(\Delta x)^2} + \frac{\Delta t^{\alpha}}{2} \times \left[\frac{\sin\pi\alpha}{\pi\alpha} f_i' \frac{s^{\alpha}}{(s + (n-1)\Delta t)^{\alpha}}\ {}_2F_1^n\left(\alpha, \alpha, \alpha+1; \frac{s}{t_n}\right) + \frac{f_i' s^{\alpha-1}}{\Gamma(\alpha)} + {}_sJ_{t_n}^{1-\alpha}\left(\frac{V_{i+1}^n - V_i^n}{2\Delta x}\right)\right];$$

$$d: \frac{a\Delta t^{\alpha}}{2(\Delta x)^3} + \frac{\Delta t^{\alpha}}{2}\left(V_i^n + \frac{f_i s^{\alpha-1}}{\Gamma(\alpha)}\right)\frac{1}{2\Delta x\Delta t\,\Gamma(1-\alpha)}\left[\frac{(t-\tau)^{1-\alpha}(\alpha\tau - t - \tau)}{(\alpha-2)(\alpha-1)}\right]_{j\Delta t}^{(j+1)\Delta t} - t_n\left[\frac{(t-\tau)^{1-\alpha}}{\alpha-1}\right]_{j\Delta t}^{(j+1)\Delta t}\Bigg];$$

e: $\dfrac{a\Delta t^{\alpha}}{2(\Delta x)^3}$;

$$-F\colon \sum_{j=0}^{n}(-1)^j\binom{\alpha}{j}V_i^{n-j+1}-\frac{\sin\pi\alpha}{\pi}\frac{s^\alpha f_i}{(s+n\Delta t)(n)^\alpha}+\frac{s^{\alpha-1}}{(n)^\alpha}\frac{\sin\pi\alpha}{\pi}f_i$$
$$+\frac{\Delta t^\alpha}{2}\frac{f_i s^{\alpha-1}}{\Gamma(\alpha)}$$
$$\times\left[\frac{\sin\pi\alpha}{\pi\alpha}f_i'\frac{s^\alpha}{(s+(n-1)\Delta t)^\alpha}\ {}_2F_1^n\left(\alpha,\alpha,\alpha+1;\frac{s}{t_n}\right)+\frac{f_i' s^{\alpha-1}}{\Gamma(\alpha)}\right.$$
$$\left.+\ {}_sJ_{t_n}^{1-\alpha}\left(\frac{V_{i+1}^n-V_i^n}{2\Delta x}\right)\right]+\frac{\Delta t^\alpha}{2}\left(V_i^n+\frac{f_i s^{\alpha-1}}{\Gamma(\alpha)}\right)$$
$$\times\left[\frac{\sin\pi\alpha}{\pi\alpha}f_i'\frac{s^\alpha}{(s+n\Delta t)^\alpha}\ {}_2F_1^{n+1}\left(\alpha,\alpha,\alpha+1;\frac{s}{t_{n+1}}\right)+\frac{f_i' s^{\alpha-1}}{\Gamma(\alpha)}\right.$$
$$-\frac{1}{2\Delta x\Delta t\,\Gamma(1-\alpha)}\left[\sum_{\mathrm{j}=1}^{\mathrm{n}-1}\mathrm{V}_{\mathrm{i}+1}^{\mathrm{j}+1}\left(\frac{(t-\tau)^{1-\alpha}(\alpha\tau-\mathrm{t}-\tau)}{(\alpha-2)(\alpha-1)}^{(j+1)\Delta t}_{j\Delta t}\right.\right.$$
$$\left.-t_j\frac{(t-\tau)^{1-\alpha}{}^{(j+1)\Delta t}}{\alpha-1}_{j\Delta t}\right)$$
$$+\sum_{\mathrm{j}=1}^{\mathrm{n}}\mathrm{V}_{\mathrm{i}+1}^{\mathrm{j}}\left(\frac{(t-\tau)^{1-\alpha}(\alpha\tau-\mathrm{t}-\tau)}{(\alpha-2)(\alpha-1)}^{(j+1)\Delta t}_{j\Delta t}\right.$$
$$\left.-(1+t_j)\frac{(t-\tau)^{1-\alpha}{}^{(j+1)\Delta t}}{\alpha-1}_{j\Delta t}\right)$$
$$-\sum_{\mathrm{j}=1}^{\mathrm{n}-1}\mathrm{V}_{\mathrm{i}-1}^{\mathrm{j}+1}\left(\frac{(t-\tau)^{1-\alpha}(\alpha\tau-\mathrm{t}-\tau)}{(\alpha-2)(\alpha-1)}^{(j+1)\Delta t}_{j\Delta t}-t_j\frac{(t-\tau)^{1-\alpha}{}^{(j+1)\Delta t}}{\alpha-1}_{j\Delta t}\right)$$
$$-\sum_{\mathrm{j}=1}^{\mathrm{n}}\mathrm{V}_{\mathrm{i}-1}^{\mathrm{j}}\left(\frac{(t-\tau)^{1-\alpha}(\alpha\tau-\mathrm{t}-\tau)}{(\alpha-2)(\alpha-1)}^{(j+1)\Delta t}_{j\Delta t}\right.$$
$$\left.\left.\left.-(1+t_j)\frac{(t-\tau)^{1-\alpha}{}^{(j+1)\Delta t}}{\alpha-1}_{j\Delta t}\right)\right]\right]-a\Delta t^\alpha\frac{f_i'''s^{\alpha-1}}{\Gamma(\alpha)}\ .$$

Далее для этих коэффициентов применяется метод прогонки, и получаем вектор решения для сеточной функции.

2.6 Алгоритм проведения вычислительных экспериментов

В задаче первый слой задан начальными условиями. Второй слой высчитывается по явной формуле:

$$V_i^{n+1} = -\sum_{j=0}^{n}(-1)^j\binom{\alpha}{j}V_i^{n-j+1} +$$
$$+\Delta t^\alpha\left\{\frac{\sin\pi\alpha}{\pi}\frac{s^\alpha f_i}{(s+n\Delta t)(n\Delta t)^\alpha} - \frac{s^{\alpha-1}}{(n\Delta t)^\alpha}\frac{\sin\pi\alpha}{\pi}f_i\right.$$
$$-\left(V_i^n + \frac{f_i s^{\alpha-1}}{\Gamma(\alpha)}\right)\left[\frac{\sin\pi\alpha}{\pi\alpha}f_i'\frac{s^\alpha}{(s+(n-1)\Delta t)^\alpha}\ {}_2F_1\ \left(\alpha,\alpha,\alpha+1;\frac{s}{t_n}\right)\right.$$
$$\left.+\frac{f_i' s^{\alpha-1}}{\Gamma(\alpha)} + {}_sJ_{t_n}^{1-\alpha}\left(\frac{V_{i+1}^n - V_{i-1}^n}{2\Delta x}\right)\right]$$
$$\left.+a\Delta t^\alpha\frac{V_{i+2}^n - 3V_{i+1}^n + 3V_i^n - V_{i-1}^n}{(\Delta x)^3} + a\Delta t^\alpha\frac{f_i''' s^{\alpha-1}}{\Gamma(\alpha)}\right\},$$
$$n = 1,2,3,\dots, i = \overline{1, I-1}.$$

Далее, начиная с третьего слоя, используется метод прогонки.

На левой границе определено граничное условие, на правой границе положено условие нулевого градиента.

3. Описание параллельного алгоритма

Был реализован параллельный алгоритм с применением библиотеки OpenMP. Распараллелили подсчет коэффициентов в формуле дробной производной Грюнвальда-Летникова, подсчет начального условия на сетке, подсчет первой и второй производной функции начального условия.

Распараллелили явную схему, в которой подсчитывались начальные значения для старта неявной схемы.

Был распараллелен подсчет коэффициентов A,B,C,F для заполнения трехдиагональной матрицы системы и вектора правой части.

Также была достигнута параллелизация нормировки конечного решения.

Параллельные регионы объявляются директивой `#pragma omp parallel`, в нужных местах проставлена директива `reduction(+:список)`, где `список` – список общих переменных. `Reduction` гарантирует корректную работу с общей переменной.

4.Результаты вычислительных экспериментов

При расчете было положено $\alpha = 0.5$, $\Delta x \in \{0.001\}$,

$\Delta t = \frac{0.1\Delta x^{\frac{3}{\alpha}}}{a}$, где $a = 1, s = 0.1, f(x) = f(x) = x^2(1-x), g(t) = 0, n = 2000$

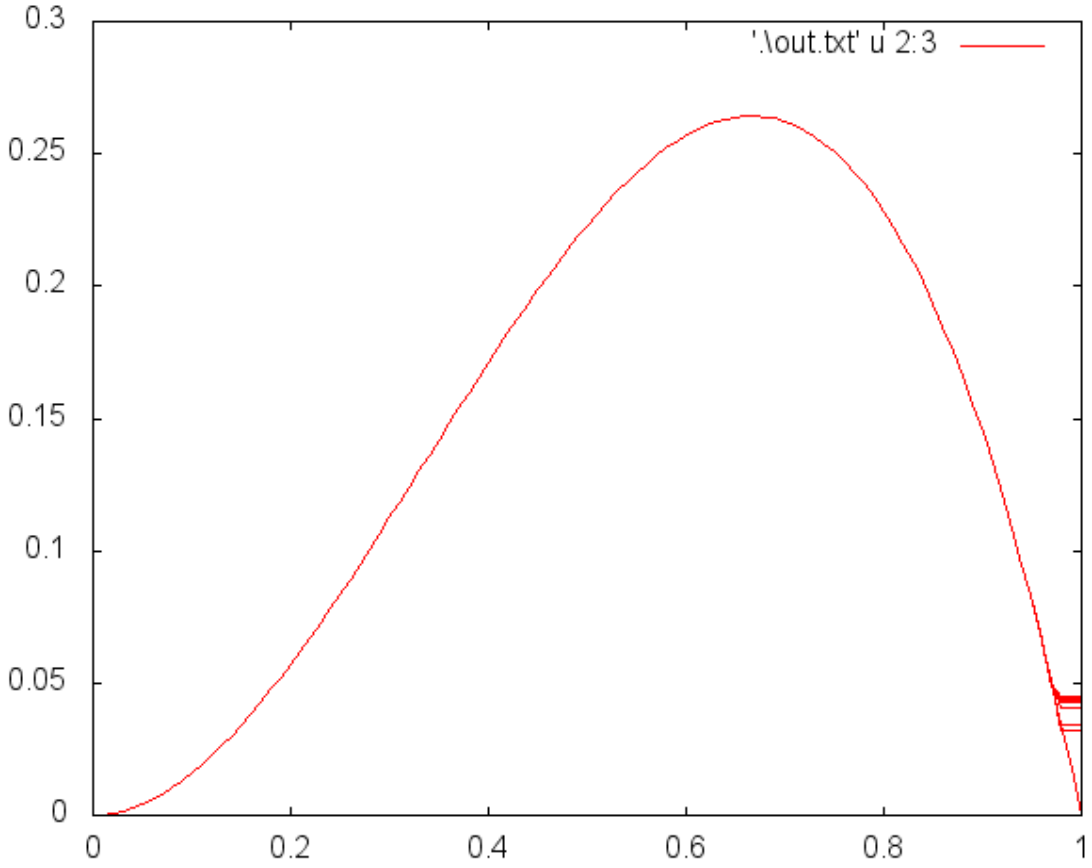

Рис. 1

На рис. 1 ось абсцисс обозначается за x, а ось ординат за значение на сетке. Так как Δx мало, то решения на разных временных слоях имеют малое отклонение.

Также был построен объемный график по которому можно было судить о том, что волна начала подниматься. На рисунке 2 справа ось абцисс, на оси Oz находятся значения решения на сетке, а внизу ось по времени, имеет направление справа налево. Разбиение по времени очень мало.

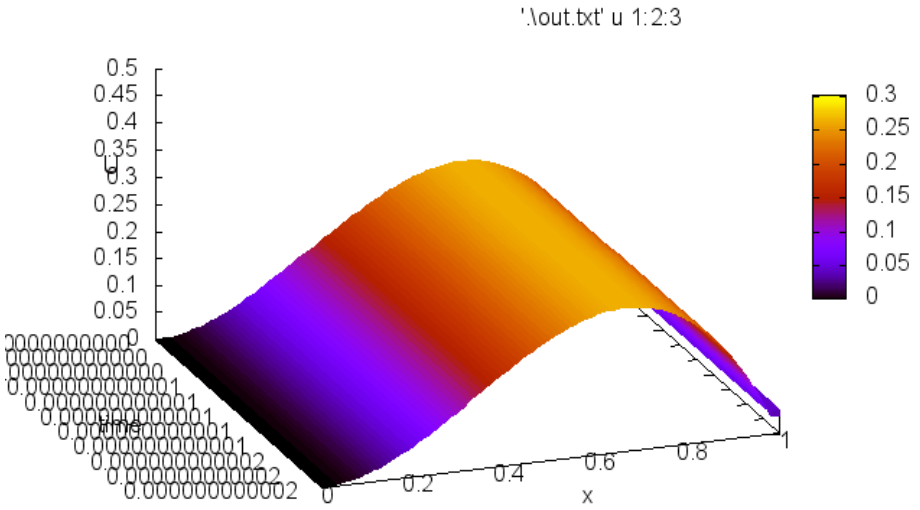

Рис. 2

Теперь исследуем эффективность и ускорение параллельного алгоритма. Как говорилось выше, n = 2000. Обратимся к таблице 1, где P-число ядер, T время затраченное на расчет, S_p – ускорение, а E_p - эффективность.

Место для формулы.

P	T, минут	S_p	E_p
1	29,035	1	1
2	15,695	1,85	0,92
4	8,681	3,34	0,83

Таблица 1

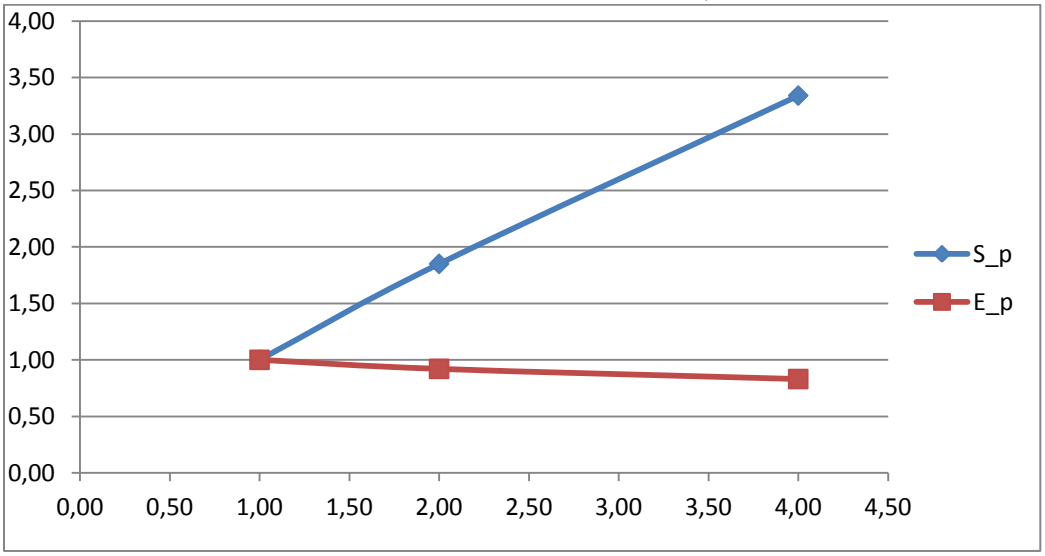

Рис. 3

На рисунке 3 внизу ось обозначает кол-во ядер, слева – значения ускорения и эффективности.

Заключение

Разработан численный алгоритм, позволяющий проводить компьютерное моделирование течения жидкости, описываемое начально-краевой задачей для уравнения Кортевега- де- Фриза дробного порядка. Написана программа на языке C++, реализующая разработанный алгоритм. Программа была оптимизирована средствами OpenMP. С использованием программы произведены серии вычислительных экспериментов на ЭВМ, в ходе которых было обнаружено, что численная схема устойчива при выбранных параметрах α, Δx, s.

Список литературы

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

2. Kilbas A.A., Srivastava H.M., Trujillo J.J. Theory and applications of fractional differential equations. - North-Holland Mathematics Studies 204, 2006. - 523 с.

3. Podlubny I. Fractional differential equations. - Academic press, 1999. - 340 с.

4. Самарский А. А., Николаев Е. С., Методы решения сеточныхуравнений. – Москва: Наука, 1978 -588с.

Приложение А.
(справочное)

Листинг программы для вычисления сеточной функции:

```
#define _USE_MATH_DEFINES
#define _CRT_SECURE_NO_WARNINGS
#include <math.h>
#include <time.h>
#include <iostream>
#include <fstream>
#include <process.h>
#include <conio.h>
#include "alglib\ap.h"
#include "alglib\specialfunctions.h"
using namespace std;
using namespace alglib;

int A=1,                          //коэффициент при правой части
	C=0,                          //левый край отрезка по x
	D=1,                          //правый край отрезка по x
	T1=0,                         //начальное время t
	T2=1,                         //конечное время t
	n=0;                          //номер слоя по времени
double alpha=0.5;           //задайте альфа на полуинтервале (0;1]
double dx=0.01;                   //шаг по x
double dt=0.1*(pow(dx,2/alpha))/(A);//dt вычисляется в зависимости от dx
double s=0.1;                     //самая первая ступенька
int stepAmountX=(D-C)/dx+1;       //кол-во узлов сеточной функции
int stepAmountT=(T2-T1)/dt;       //полные временные слои
int myStep=30;                    //число временных слоев необходимых для вычисления
int dodo=2;                       //заглушка для явной схемы
//массивы для прогонки
double* A1=(double*)calloc(stepAmountX,sizeof(double));//поддиагональ
double* B1=(double*)calloc(stepAmountX,sizeof(double));//диагональ
double* C1=(double*)calloc(stepAmountX,sizeof(double));//наддиагональ
double* D1=(double*)calloc(stepAmountX,sizeof(double));//здесь правая часть системы
double* X=(double*)calloc(stepAmountX,sizeof(double));//метод прогонки записывает решение
double* ksi=(double*)calloc((stepAmountX+1),sizeof(double));//прогоночные коэфициенты
double* eta=(double*)calloc((stepAmountX+1),sizeof(double));//прогоночные коэфициенты
//-----------------------------------------------------------------------------------
double* func = (double*)calloc(stepAmountX, sizeof(double)); //массив значений косинуса
double* funcDer = (double*)calloc(stepAmountX, sizeof(double));//1 производная косинуса
double* funcDer2 = (double*)calloc(stepAmountX, sizeof(double));//2 производная косинуса
double** U=(double**)calloc(myStep, sizeof(double*)); //временный массив для значений V
double** Fin=(double**)calloc(myStep, sizeof(double*));//конечный массив результатов
double* Coeff = (double*)calloc(myStep, sizeof(double));
//метод прогонки
//sloy - номер временного слоя на котором выполняется прогонка
void Shuttle(int sloy){
	int N=stepAmountX;
	for(int j=0;j<N;j++)
	{
		X[j]=0;
	}
	A1[0]=0;
	C1[N-1]=0;
	for(int i=0;i<N;i++)
	{
		B1[i]=-B1[i];
	}
	ksi[0]=eta[0]=0;
	for(int i=0;i<N;i++)
	{
		ksi[i+1]=C1[i]/(B1[i]-A1[i]*ksi[i]);
		eta[i+1]=(A1[i]*eta[i]-D1[i])/(B1[i]-A1[i]*ksi[i]);
	}
	X[N-1]=-eta[N];
```

```
	for(int i=N-2;i>-1;i--)
	{
		X[i]=ksi[i+1]*X[i+1]-eta[i+1];

	}
	for(int i=0;i<N;i++)
	{
		X[i]=-X[i];
	}
	for(int j=0;j<N+1;j++)
	{
		ksi[j]=0;
		eta[j]=0;
	}
	for(int j=0;j<N;j++)
	{
		A1[j]=B1[j]=C1[j]=D1[j]=0;
	}
	for(int j=0;j<stepAmountX;j++)
	{
		U[sloy][j]=X[j];
	}

}
//гамма функция
//input - аргумент гамма функции
double gamma(double input)
{
	return alglib::gammafunction(input);
}
//записывает результаты в файл, с нормировкой или без неё
void toFile(int t,double d)
{
	double shag=0.;
	fstream str;
	str.open("out.txt", ios::out);
	for(int n=0;n<t;n++)
	{
		str<<"#"<<n<<endl;  //для gnuplot
		shag=0;
		for(int i=0;i<stepAmountX;i++)
		{
			if(flag==true){
				str<<shag<<"\t"<<Fin[n][i]/gamma(alpha)<<endl;
				}else{
				str<<shag<<"\t"<<Fin[n][i]<<endl;
			}
			shag=shag+d;
		}
		str<<endl;
	}
	str.close();
}
//гипергеометрическая функция
double hyperGeometric(double a, double b, double c, double z)
{
	if(z==0) return 1.;//подстановка 0

	double res=1., res2=1.;
	int hyperIter=20;		//кол-во слагаемых для гипергеометрической функции
	for(int i=1;i<hyperIter;i++)
	{
		res2=1;
		for(int l=0;l<i-1;l++)
		{
			res2*=(a+l)*(b+l)/((1+l)*(c+l));
		}
		res+=pow(z,i)*res2;
```

```
	}
	return res;
}
//функция - начальное условие
double fCos(double x)
{
	if(flag==false)
	{
		if( abs(cos(M_PI*x/2)) > pow(10.,-15.)) return cos(M_PI*x/2);
		else return 0.;
	}
	else
	{
		double res=0;
		if ( x>=0 && x<0.5 )  res=1;
		if ( x==0.5 )          res=0.5;
		if ( x>0.5 && x<=1 )  res=0;
		return res;
	}
}
//функция - граничное условие
double g(double t)
{
	return 1.;
}
//центральная производная
//sloy - временной слой на котором берется производная
//nextTochka - X(i+1)
//currTochka - X(i-1)
double centrDeriv(int sloy, int nextTochka, int currTochka)
{
	return (U[sloy][nextTochka]-U[sloy][currTochka])/(2*dx);
}
// dt/()
//вычисляет интеграл
//podstanovka - Tn
//up - T(j+1)
//down - Tj
double Integral1(double podstanovka,double up,double down)
{
	double v = (1/(alpha-1))*(pow(s+podstanovka*dt-up*dt,1-alpha)-
pow(s+podstanovka*dt-down*dt,1-alpha));
	cout.precision(15);
	return v;
}
//-------------------------------------------------------------------------------------
--
// t*dt/()
//podstanovka - Tn
//up - T(j+1)
//down - Tj
double Integral2(double podstanovka,double up,double down)
{
	double v=((1.)/((alpha-2)*(alpha-1)))*(pow(s+podstanovka*dt-up*dt,1-
alpha)*(alpha*up*dt-s-podstanovka*dt-up*dt)-pow(s+podstanovka*dt-down*dt,1-
alpha)*(alpha*down*dt-s-podstanovka*dt-down*dt));
	return v;
}
// dt/()
double newIntegral1(double podstanovka, int check)
{
	double ret=0;
	if (podstanovka>check)
	{
		ret=Integral1(podstanovka,check,check-1);
	}
	return ret;
}
```

```
double newIntegral2(double podstanovka, int check)
{
	double ret=0;
	if (podstanovka>check)
	{
		ret=Integral2(podstanovka,check,check-1);

	}
	return ret;
}
//подсчет дробного интеграла на n слое
//down - номер временного слоя
//order -  порядок дробного интеграла
//step - номер шага по Ox
double fracInt(double down, double order, int step)
{
	if(step>=stepAmountX) return 0.;
	double result=0;
	for (int j=1;j<down;j++)
	{
		result+=newIntegral1(n-1,j+1)*(U[j][step]+((s+(j-1)*dt)/dt)*(-
U[j+1][step]+U[j][step])) + 1/(dt)*newIntegral2(n-1,j+1)*(U[j+1][step]-U[j][step]);
	}
	return result/gamma(order);
}
int main()
{
	fstream temp,temp1;
	temp.open("temp.txt", ios::out);
	temp1.open("temp1.txt", ios::out);
	cout<<stepAmountT<<endl;
	double
shag=0.,sum1=0.,sum2=0.,sum3=0.,sum4=0.,sum5=0.,sum6=0.,sum7=0.,sum8=0.,allSum=0.,sumGrou
p=0;
	clock_t start,end,tp;	//переменные для подсчета потраченного времени
	double time=0.;
	//-------------------проиницализируем начальные условия-------------------------
-
	U=(double**)calloc(myStep, sizeof(double*));
	Fin=(double**)calloc(myStep, sizeof(double*));
	double* Us=(double*)calloc(stepAmountX, sizeof(double));//U(s)
	for(int i = 0; i < myStep; i++)
	{
		U[i] = (double*)calloc(stepAmountX, sizeof(double));
		Fin[i] = (double*)calloc(stepAmountX, sizeof(double));
	}
	Coeff[0]=1;
	Coeff[1]=alpha;
	for(int i=2;i<myStep;i++)
	{
		Coeff[i]=Coeff[i-1]*(alpha-i+1)/(i);
	}
	for (int i=0;i<myStep;i++)
	{
		Coeff[i]=pow(-1,i)*Coeff[i];
	}
	for(int i=0;i<stepAmountX;i++)
	{
		func[i]=fCos(i*dx);
	}
	for(int i=1;i<stepAmountX-1;i++)
	{
		funcDer[i]=(func[i+1]-func[i-1])/(2*dx);
		funcDer2[i]=(func[i+1]-2*func[i]+func[i-1])/(dx*dx);
	}
	funcDer[0]=funcDer2[0]=func[0];
	funcDer[stepAmountX-1]=funcDer2[stepAmountX-1]=func[stepAmountX-1];
	for(int i=0;i<stepAmountX;i++)
```

```
	{
		U[0][i]=0;	//начальные слои
		U[1][i]=0;
	}
	for(int i=2;i<myStep;i++)
	{
		U[i][0]=g(s+n*dt)-fCos(0)*pow(s,alpha-1)/gamma(alpha);//левый край
	}
	start=clock();
	for (int i=0;i<stepAmountX;i++)
	{
		Us[i]=func[i]*pow(s,alpha-1)/gamma(alpha);
	}
	//Явная схема
	for(n=1;n<dodo;n++)
	{
		for(int i=1;i<stepAmountX-1;i++)
		{
			allSum=0.;
			sum1=sum2=sum3=sum4=sum5=sum6=0.;
		sum2=sin(M_PI*alpha)*pow(s,alpha)*func[i]/(M_PI*(s+n*dt)*pow(n*dt,alpha))-
pow(s,alpha-1)*sin(M_PI*alpha)*func[i]/(pow(n*dt,alpha)*M_PI);
			sum3=U[n][i]+func[i]*pow(s,alpha-1)/gamma(alpha);
	sum4=sin(M_PI*alpha)/(M_PI*alpha)*funcDer[i]*pow(s,alpha)*hyperGeometric(alpha,alp
ha,alpha+1,s/(s+(n-1)*dt))/pow(s+(n-1)*dt,alpha)+funcDer[i]*pow(s,alpha-1)/gamma(alpha);
			sum5=1/(2*dx)*(fracInt(n,1-alpha,i+1)-fracInt(n,1-alpha,i-1));
			sum6=A*(U[n][i+1]-2*U[n][i]+U[n][i-1])/(dx*dx)  +  A*pow(s,alpha-
1)*funcDer2[i]/gamma(alpha);
			allSum+=-sum1+pow(dt,alpha)*(sum2-sum3*(sum4+sum5)+sum6);
			U[n+1][i]=allSum;
		}
		U[n+1][stepAmountX-1]=U[n+1][stepAmountX-2]+Us[stepAmountX-2]-
Us[stepAmountX-1];
	}
	//Неявная схема
	for(n=dodo+1;n<myStep;n++)
	{
		for(int i=1;i<stepAmountX-1;i++)
		{
			sum1=0;
			for(int j=1;j<=n;j++)	{sum1+=Coeff[j]*U[n-j][i];}
			//i-1
			A1[i]=(-A*pow(dt,alpha))/pow(dx,2.) +
pow(dt,alpha)/2.*(U[n][i]+func[i]*pow(s,alpha-1)/gamma(alpha))*(1./(2.*dx*dt*gamma(1-
alpha)))*(newIntegral2(n+1,n)-(s+(n-1)*dt)*newIntegral1(n+1,n));
			//i+1
			C1[i-1]=(-A*pow(dt,alpha))/(pow(dx,2)) +
(pow(dt,alpha))/2.*(U[n][i]+func[i]*pow(s,alpha-1)/gamma(alpha))*((-1)/(2.*dx*dt*gamma(1-
alpha)))*(newIntegral2(n+1,n)-(s+(n-1)*dt)*newIntegral1(n+1,n));
			//i
			B1[i-1]=1 +
((pow(dt,alpha))/2.)*((sin(M_PI*alpha)*pow(s,alpha)*funcDer[i]*hyperGeometric(alpha,alpha
,alpha+1,s/(s+(n-1)*dt)))/(M_PI*alpha*pow(s+(n-1)*dt,alpha))+(funcDer[i]*pow(s,alpha-
1)/(gamma(alpha))) + (1./(2*dx))*(fracInt(n,1-alpha,i+1)-fracInt(n,1-alpha,i-1))) +
((2*A*pow(dt,alpha))/pow(dx,2.));
			sumGroup=0;
			//слагаемые - остатки от n+1 fracInt'a
			for (int j=1;j<=n-1;j++)
			{
				sumGroup+=(U[j+1][i+1]-U[j+1][i-1])*(newIntegral2(n,j+1)-
(s+(j-1)*dt)*newIntegral1(n,j+1));
			}
			for (int j=1;j<=n;j++)
			{
				sumGroup+=(U[j][i+1]-U[j][i-1])*(newIntegral2(n,j+1)-(s+(j-
1)*dt)*newIntegral1(n,j+1));
			}
```

```
                        D1[i-1]=-(sum1  -
((sin(M_PI*alpha)*pow(s,alpha)*func[i])/(M_PI*pow(n,alpha)*(s+n*dt))) + ((pow(s,alpha-
1)*func[i]*sin(M_PI*alpha))/(M_PI*pow(n,alpha)))  +  ((pow(dt,alpha)*func[i]*pow(s,alpha-
1))/gamma(alpha))*((sin(M_PI*alpha)*pow(s,alpha)*funcDer[i]*hyperGeometric(alpha,alpha,al
pha+1,s/(s+(n-1)*dt)))/(M_PI*alpha*pow(s+(n-1)*dt,alpha))+(funcDer[i]*pow(s,alpha-
1)/gamma(alpha))+((1./(2*dx))*(fracInt(n,1-alpha,i+1)-fracInt(n,1-alpha,i-1)))  +
((pow(dt,alpha))/2.)*(U[n][i]+func[i]*pow(s,alpha-
1)/gamma(alpha))*(sin(M_PI*alpha)*pow(s,alpha)*funcDer[i]*hyperGeometric(alpha,alpha,alph
a+1,s/(s+n*dt))/(M_PI*alpha*pow(s+n*dt,alpha))+(funcDer[i]*pow(s,alpha-1)/gamma(alpha))-
(1./(2*dx*dt*gamma(1-alpha)))*sumGroup)  -  ((A*pow(s,alpha-
1)*funcDer2[i]*pow(dt,alpha))/(gamma(alpha))));
                }
                //(ищем A10, C9, B9, B10, D9, D10, при 11 узлах) формула универсальна для
всех не найденных A B C D
                for(int i=stepAmountX-1;i<stepAmountX;i++)
                {
                        for(int j=1;j<=n;j++)      {sum1+=Coeff[j]*U[n-j][i];}
                        //i-1
                        A1[i]=(-A*pow(dt,alpha))/pow(dx,2.) +
pow(dt,alpha)/2.*(U[n][i]+func[i]*pow(s,alpha-1)/gamma(alpha))*(1./(2.*dx*dt*gamma(1-
alpha)))*(newIntegral2(n+1,n)-(s+(n-1)*dt)*newIntegral1(n+1,n));
                        //i+1
                        C1[i-1]=(-A*pow(dt,alpha))/(pow(dx,2)) +
(pow(dt,alpha))/2.*(U[n][i]+func[i]*pow(s,alpha-1)/gamma(alpha))*((-1)/(2.*dx*dt*gamma(1-
alpha)))*(newIntegral2(n+1,n)-(s+(n-1)*dt)*newIntegral1(n+1,n));
                        //i
B1[i-1]=1 +
((pow(dt,alpha))/2.)*((sin(M_PI*alpha)*pow(s,alpha)*funcDer[i]*hyperGeometric(alpha,alpha
,alpha+1,s/(s+(n-1)*dt)))/(M_PI*alpha*pow(s+(n-1)*dt,alpha))+(funcDer[i]*pow(s,alpha-
1)/(gamma(alpha))) + (1./(2*dx))*(fracInt(n,1-alpha,i+1)-fracInt(n,1-alpha,i-1))) +
((2*A*pow(dt,alpha))/pow(dx,2.));
                        //последняя B (B10)
                        B1[stepAmountX-1]=1 +
((pow(dt,alpha))/2.)*((sin(M_PI*alpha)*pow(s,alpha)*0*hyperGeometric(alpha,alpha,alpha+1,
s/(s+(n-1)*dt)))/(M_PI*alpha*pow(s+(n-1)*dt,alpha))+(funcDer[i]*pow(s,alpha-
1)/(gamma(alpha))) + (1./(2*dx))*(fracInt(n,1-alpha,stepAmountX+1)-fracInt(n,1-
alpha,stepAmountX-1))) + ((2*A*pow(dt,alpha))/pow(dx,2.));
                        sumGroup=0;
                        //слагаемые - остатки от n+1 fracInt'a
                        for (int j=1;j<=n-1;j++)
                        {
                                sumGroup+=(0-U[j+1][i-1])*(newIntegral2(n,j+1)-(s+(j-
1)*dt)*newIntegral1(n,j+1));
                        }
                        for (int j=1;j<=n;j++)
                        {
                                sumGroup+=(0-U[j][i-1])*(newIntegral2(n,j+1)-(s+(j-
1)*dt)*newIntegral1(n,j+1));
                        }
                        D1[i-1]=-(sum1  -
((sin(M_PI*alpha)*pow(s,alpha)*func[i])/(M_PI*pow(n,alpha)*(s+n*dt))) + ((pow(s,alpha-
1)*func[i]*sin(M_PI*alpha))/(M_PI*pow(n,alpha))) + ((pow(dt,alpha)*func[i]*pow(s,alpha-
1))/gamma(alpha))*((sin(M_PI*alpha)*pow(s,alpha)*funcDer[i]*hyperGeometric(alpha,alpha,al
pha+1,s/(s+(n-1)*dt)))/(M_PI*alpha*pow(s+(n-1)*dt,alpha))+(funcDer[i]*pow(s,alpha-
1)/gamma(alpha))+((1./(2*dx))*(fracInt(n,1-alpha,i+1)-fracInt(n,1-alpha,i-1)))  +
((pow(dt,alpha))/2.)*(U[n][i]+func[i]*pow(s,alpha-
1)/gamma(alpha))*(sin(M_PI*alpha)*pow(s,alpha)*funcDer[i]*hyperGeometric(alpha,alpha,alph
a+1,s/(s+n*dt))/(M_PI*alpha*pow(s+n*dt,alpha))+(funcDer[i]*pow(s,alpha-1)/gamma(alpha))-
(1./(2*dx*dt*gamma(1-alpha)))*sumGroup)  -  ((A*pow(s,alpha-
1)*funcDer2[i]*pow(dt,alpha))/(gamma(alpha))));
                        //последняя D  (D10)
                        sumGroup=0;
                        for (int j=1;j<=n-1;j++)
                        {
                                sumGroup+=(0-U[j+1][stepAmountX-1])*(newIntegral2(n,j+1)-
(s+(j-1)*dt)*newIntegral1(n,j+1));
                        }
                        for (int j=1;j<=n;j++)
```

```
                    {
                          sumGroup+=(0-U[j][stepAmountX-1])*(newIntegral2(n,j+1)-(s+(j-
1)*dt)*newIntegral1(n,j+1));
                    }
                    D1[stepAmountX-1]=-(sum1  -
((sin(M_PI*alpha)*pow(s,alpha)*func[i])/(M_PI*pow(n,alpha)*(s+n*dt))) + ((pow(s,alpha-
1)*0*sin(M_PI*alpha))/(M_PI*pow(n,alpha)))  +  ((pow(dt,alpha)*0*pow(s,alpha-
1))/gamma(alpha))*((sin(M_PI*alpha)*pow(s,alpha)*0*hyperGeometric(alpha,alpha,alpha+1,s/(
s+(n-1)*dt)))/(M_PI*alpha*pow(s+(n-1)*dt,alpha))+(0*pow(s,alpha-
1)/gamma(alpha))+((1.)/(2*dx))*(fracInt(n,1-alpha,stepAmountX+1)-fracInt(n,1-
alpha,stepAmountX-1)))  +  ((pow(dt,alpha))/2.)*(0+0*pow(s,alpha-
1)/gamma(alpha))*(sin(M_PI*alpha)*pow(s,alpha)*0*hyperGeometric(alpha,alpha,alpha+1,s/(s+
n*dt))/(M_PI*alpha*pow(s+n*dt,alpha))+(0*pow(s,alpha-1)/gamma(alpha))-
(1./(2*dx*dt*gamma(1-alpha)))*sumGroup)  -  ((A*pow(s,alpha-
1)*0*pow(dt,alpha))/(gamma(alpha))));
                    }
              Shuttle(n);
       }
       for(int l=0;l<myStep;l++)
       {
              for(int p=0;p<stepAmountX;p++)
              {
                    Fin[l][p]=U[l][p]+Us[p];
              }
       }
       end=clock();
       time=(double)(end-start)/CLOCKS_PER_SEC;
       cout<<"time="<<time<<" second\n";
       toFile(myStep,dx);
       system("plot.plt");
       temp1.close();
       //_getch();
       return 0;
}
```