

Numerical calculation of the Kolmogorov-Fisher reaction-diffusion equation

Sayfullayeva Maftuha Zafrullaevna

National University of Uzbekistan, E-mail: maftuha87@mail.ru

Annotation: The equation of the parabolic type of the Kolmogorov-Fisher diffusion reaction was considered, which was solved using the Picard iteration method, a special method and the Newton method.

Key words:

1. Introduction

Everyone knows that today all processes and phenomena are subject to one or another law. In turn these laws can be written down in the form of mathematical formulas. Many of them are non-linear equations and their study is of great importance.

Equation of parabolic type, Kolmogorov-Fisher's reaction-diffusion from this number.

We are given the following equation:

$$\Omega = \{(t, x) : 0 < t < T, 0 < x < X\} \quad -div(u(t), u)$$
$$Lu \equiv -u_t + \nabla(D \nabla u) + Ku(1 - u^\beta) = 0 \quad (1)$$

Here $D = D(x, t)$ and $K = K(x, t)$ are the reaction-diffusion coefficients.

Suppose that the following conditions are met initially:

$$u(0, x) = \psi(x) \geq 0, \quad 0 \leq x \leq X, \quad (2)$$

but on the edges:

$$\begin{cases} u(t, 0) = \phi_1(t) > 0 \\ u(t, X) = \phi_2(t) = 0 \end{cases}, \quad t \in [0, T], \quad (3)$$

Now we're approximating the equation:

$$\frac{y_i^{j+1} - y_i^j}{\tau} = \frac{D}{h^2} (y_{i+1}^{j+1} - 2y_i^{j+1} + y_{i-1}^{j+1}) + K \cdot y_i^{j+1} - K \cdot (y_i^{j+1})^{\beta+1}$$

Then we will calculate approximate solutions using the method of testing, but since on the right side the last term has non-linearity it makes our task a little bit more complicated.

We use the iteration method for numerical calculations. To do this, let's take y_{j+1} unknown persons located on the $j+1$ layer with the help of $s+1$ iterations, and the last member will be considered separately. Let's write down

$$b_i(y^{j+1}) = (y_i^{j+1})^{\beta+1}$$

We linearize our recording using the following three methods:

1. For the Picard method

$$b_i(y^{j+1}) = \binom{s}{y_i}^{\beta+1}$$

2. For a special method

$$b_i(y^{j+1}) = y_i^{s+1} \binom{s}{y_i}^{\beta}$$

3. For Newton's method

$$b_i(y^{j+1}) = \binom{s}{y_i}^{\beta+1} + \beta \binom{s}{y_i}^{\beta} (y_i^{s+1} - y_i)$$

It should be noted that all three methods will return the same results, but the Newton method uses fewer iterations.

Let's calculate A_i , B_i , C_i , F_i coefficients for the three methods:

1. For the Picard method

$$A_i = \frac{\tau D}{h^2}, \quad B_i = \frac{\tau D}{h^2}$$

$$C_i = A_i + B_{i+1} - K\tau, \quad F_i = y_i^s - K \cdot \tau \cdot \binom{s}{y_i^{j+1}}^{\beta+1}, \quad s = 0, 1, 2, \dots, \quad i = 1, 2, \dots, n$$

2. For special method

$$A_i = \frac{\tau D}{h^2}, \quad B_i = \frac{\tau D}{h^2}$$

$$C_i = A_i + B_{i+1} - K \cdot \tau + K \cdot \tau \cdot \binom{s}{y_i^{j+1}}^{\beta}, \quad F_i = y_i^j, \quad s = 0, 1, 2, \dots, \quad i = 1, 2, \dots, n$$

3. For Newton method

$$A_i = \frac{\tau D}{h^2}, \quad B_i = \frac{\tau D}{h^2}$$

$$C_i = A_i + B_{i+1} - K \cdot \tau + \beta \cdot K \cdot \tau \cdot \binom{s}{y_i^{j+1}}^{\beta}, \quad F_i = y_i^j - K \cdot \tau \cdot \binom{s}{y_i^{j+1}}^{\beta+1} + \beta \cdot K \cdot \tau \cdot \binom{s}{y_i^{j+1}}^{\beta+1}, \quad s = 0, 1, 2, \dots, \quad i = 1, 2, \dots, n$$

The iterations will continue until the following condition is met.

$$\max_{\substack{0 \leq i \leq n \\ 0 \leq j \leq m}} \left| \bar{y}_{i,j}^{s+1} - \bar{y}_{i,j}^s \right| < \varepsilon$$

The initial iteration (s=0) decides from the previous layer: $y_{ij} = y_{ij-1}$

From here we can see that in all three cases, our scheme is stable in all cases $h = \frac{X}{n}$ and $\tau = \frac{T}{m}$. Approximation inaccuracy $O(h^2 + \tau)$.

The equation can be solved without using iteration, we will need the predictor-corrector method. Let's take j+0.5 between j and j+1 layers and calculate $y^{j+0.5}$ with the help of the runtime method, then repeat this action for the interval from j to j+0.5 and calculate y^{j+1} .

Let's write down the approximation,

$$\begin{cases} \frac{y_i^{j+1/2} - y_i^j}{0.5 \cdot \tau} = \frac{y_{i+1}^{j+1/2} - 2y_i^{j+1/2} + y_{i-1}^{j+1/2}}{h^2} + K \cdot y_i^j \left(1 - (y_i^j)^\beta\right) \\ \frac{y_i^{j+1} - y_i^j}{\tau} = \frac{1}{2} \left(\frac{y_{i+1}^{j+1/2} - 2y_i^{j+1/2} + y_{i-1}^{j+1/2}}{h^2} + \frac{y_{i+1}^{j+1} - 2y_i^{j+1} + y_{i-1}^{j+1}}{h^2} \right) + K \cdot y_i^{j+1/2} \left(1 - (y_i^{j+1/2})^\beta\right) \end{cases}$$

It seems to be easier to find a solution than in the previous scheme, but there are problems with this solution. For the scheme to be sustainable, the following condition must be met

$$\frac{\tau}{h^2} \leq \frac{1}{2}$$

Suppose you want to see the changes in the process over a long period of time t. Some tasks are observed during months, years. Then, to meet the condition, you need to raise M, but it will complicate the calculation process. Even the newest computers cannot do it. Therefore, the first obtained scheme is taken as normal.

Now let's consider (1) -(3) equations, where $D = D(x, t, u)$. In this case, the equation will be called quasi-linear. Its approximation

$$\frac{y_i^{j+1} - y_i^j}{\tau} = \frac{1}{h^2} \left[a_{i+1}(y^{j+1})(y_{i+1}^{j+1} - y_i^{j+1}) - a_i(y^{j+1})(y_i^{j+1} - y_{i-1}^{j+1}) \right] + K \cdot (y_i^{j+1}) - K \cdot b_i(y^{j+1})$$

It can be written down as

$$a) \quad a_i(y^{j+1}) = D \cdot \left(\frac{y_{i+1}^{j+1} + y_i^{j+1}}{2} \right)^\sigma$$

$$b) \quad a_i(y^{j+1}) = D \cdot \left(\frac{(y_{i+1}^{j+1})^\sigma + (y_i^{j+1})^\sigma}{2} \right)$$

The equation is linearized in the same way as the above scheme, $a_i(y^{j+1})$ is taken by s iterations.

Now let's consider a two-dimensional case. The two-dimensional equation will be in the following form:

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x_1} \left(D \cdot \frac{\partial u}{\partial x_1} \right) + \frac{\partial}{\partial x_2} \left(D \cdot \frac{\partial u}{\partial x_2} \right) + K \cdot u \cdot (1 - u^\beta)$$

$$u|_{t=0} = u_0(x_1, x_2) \geq 0,$$

$$u|_{x=0} = \phi_0(t, x_2), \quad u|_{x=X} = \phi_1(t, x_2)$$

$$u|_{y=0} = \varphi_0(t, x_1), \quad u|_{y=Y} = \varphi_1(t, x_1)$$

here $u = u(x, y, t)$, and $D = D(x, t, u)$, $K = K(x, t)$, consider as diffusion coefficients. Let's calculate the equation with the help of Pisman-Rekford scheme. Let's take $j+0.5$ between j and $j+1$ layers and calculate $y_{j+0.5}$ using the method of running, then repeat this action for the interval from j to $j+0.5$ and calculate y_{j+1} . The opposite is also true.

The approximation looks like it:

$$\left\{ \begin{array}{l} \frac{y_{i,j}^{k+1/2} - y_{i,j}^k}{0.5 \cdot \tau} = \frac{1}{h_1^2} \left[a_{i+1,j} \left(y^{k+1/2} \right) \left(y_{i+1,j}^{k+1/2} - y_{i,j}^{k+1/2} \right) - a_{i,j} \left(y^{k+1/2} \right) \left(y_{i,j}^{k+1/2} - y_{i-1,j}^{k+1/2} \right) \right] + \\ \frac{1}{h_2^2} \left[\bar{a}_{i,j+1} \left(y^k \right) \left(y_{i,j+1}^k - y_{i,j}^k \right) - \bar{a}_{i,j} \left(y^k \right) \left(y_{i,j}^k - y_{i,j-1}^k \right) \right] + K \cdot y_{i,j}^{k+1/2} \cdot \left(1 - \left(y_{i,j}^{k+1/2} \right)^\beta \right) \\ \frac{y_{i,j}^{k+1} - y_{i,j}^{k+1/2}}{0.5 \cdot \tau} = \frac{1}{h_1^2} \left[a_{i+1,j} \left(y^{k+1/2} \right) \left(y_{i+1,j}^{k+1/2} - y_{i,j}^{k+1/2} \right) - a_{i,j} \left(y^{k+1/2} \right) \left(y_{i,j}^{k+1/2} - y_{i-1,j}^{k+1/2} \right) \right] + \\ \frac{1}{h_2^2} \left[\bar{a}_{i,j+1} \left(y^{k+1} \right) \left(y_{i,j+1}^{k+1} - y_{i,j}^{k+1} \right) - \bar{a}_{i,j} \left(y^{k+1} \right) \left(y_{i,j}^{k+1} - y_{i,j-1}^{k+1} \right) \right] + K \cdot y_{i,j}^{k+1} \cdot \left(1 - \left(y_{i,j}^{k+1} \right)^\beta \right) \end{array} \right.$$

A_i, B_i, C_i, F_i ratios,

$$A_i = \frac{0.5 \cdot \tau}{h_1^2} \cdot a_{i-1,j} \left(y^{k+1/2} \right), \quad B_i = \frac{0.5 \cdot \tau}{h_1^2} \cdot a_{i,j} \left(y^{k+1/2} \right), \quad C_i = A_i + B_i + 1 - 0.5 \cdot \tau \cdot K$$

$$F_i^s = y_{i,j}^k + \frac{0.5 \cdot \tau}{h_2^2} \cdot \left[\bar{a}_{i,j+1}(y^k)(y_{i,j+1}^k - y_{i,j}^k) - \bar{a}_{i,j}(y^k)(y_{i,j}^k - y_{i,j-1}^k) \right] - 0.5 \cdot \tau \cdot K \cdot \left(y_{i,j}^k \right)^{\frac{s}{k+1}}$$

A_j, B_j, C_j, F_j ratios,

$$A_j^s = \frac{0.5 \cdot \tau}{h_2^2} \cdot \bar{a}_{i,j-1} \left(y^{k+1} \right), \quad B_j^s = \frac{0.5 \cdot \tau}{h_2^2} \cdot \bar{a}_{i,j} \left(y^{k+1} \right), \quad C_j^s = A_j^s + B_j^s + 1 - 0.5 \cdot \tau \cdot K$$

$$F_j^s = y_{i,j}^{k+\frac{1}{2}} + \frac{0.5 \cdot \tau}{h_1^2} \cdot \left[a_{i+1,j} \left(y^{k+\frac{1}{2}} \right) \left(y_{i+1,j}^{k+\frac{1}{2}} - y_{i,j}^{k+\frac{1}{2}} \right) - a_{i,j} \left(y^{k+\frac{1}{2}} \right) \left(y_{i,j}^{k+\frac{1}{2}} - y_{i-1,j}^{k+\frac{1}{2}} \right) \right] - 0.5 \cdot \tau \cdot K \cdot \left(y_{i,j}^{k+\frac{1}{2}} \right)^{\frac{s}{k+1}}$$

In general, the method of direction variables is used to solve multidimensional schemes ($x \in R^n$).

Calculation results.

Let's calculate (1)-(3) equation with the following data:

$$D=1; K=1; \beta = 1; N=20; M=10; T=1; X=1; \varepsilon = 0.001;$$

The initial condition: $u_0(x) = \frac{1}{2} e^{-\frac{x^2}{4}}$, and there'll be edge-to-edge:

$$\phi_0(t) = \frac{e^t}{1+e^t} \cdot \frac{1}{\sqrt{1+t}}; \quad \phi_1(t) = \frac{e^t}{1+e^t} \cdot \frac{1}{\sqrt{1+t}} \cdot e^{-\frac{1}{4(1+t)}}$$

The results of the calculations are shown in the table below:

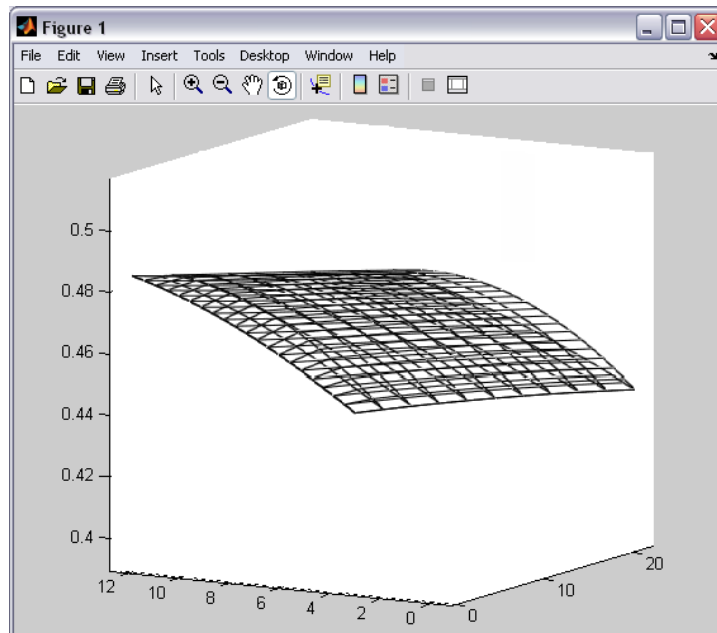
TABLE 1

t	The exact bottom solution	Approximate solution	Uncertainty	Number of iterations		
				Picard's method	Special method	Newton's method
j=1	0,50668	0,47290	0,03378	6	6	1
j=2	0,51345	0,47645	0,037	4	3	1
j=3	0,51989	0,48017	0,03972	4	3	1
j=4	0,52606	0,48389	0,04217	4	3	1

j=5	0,53189	0,48749	0,0444	4	3	1
j=6	0,53730	0,49088	0,04642	4	3	1
j=7	0,54172	0,49397	0,04775	3	3	1
j=8	0,54612	0,49672	0,0494	3	3	1
j=9	0,55005	0,49908	0,05097	3	3	1
j=10	0,55345	0,50103	0,05242	3	2	1

The lower three lines show the iteration calculations by the Picard method, the special Newton method. As mentioned above, Newton's method gave the best solutions.

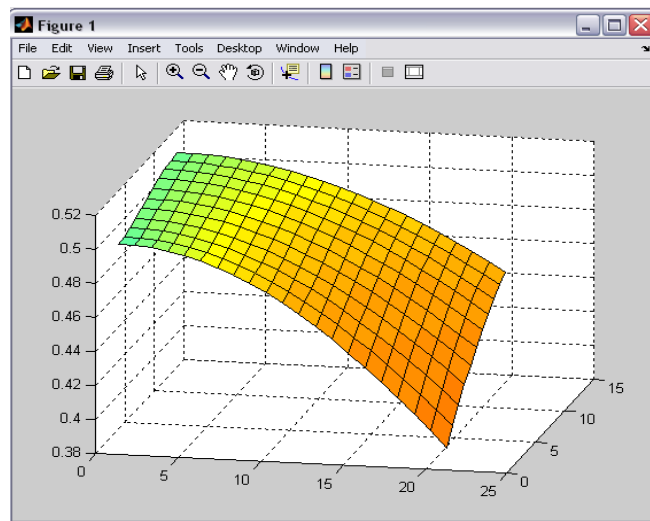
Let's draw a three-dimensional graph:



The chart shows that the function chart will approach the OX-axis when the x-axis grows. Let's move to the two-dimensional equation. The solution will be written in the form of a table and a three-dimensional chart:

TABLE 2

t	The exact bottom solution	Approximate solution	Inaccuracy	Number of iterations
k=1	0,61988	0,61844	0,00144	2
k=2	0,61956	0,61691	0,00265	2
k=3	0,61904	0,61536	0,00368	2
k=4	0,61830	0,61374	0,00456	2
k=5	0,61734	0,61205	0,00529	2
k=6	0,61617	0,61025	0,00529	2
k=7	0,61478	0,60833	0,00645	2
k=8	0,61316	0,60629	0,00687	2
k=9	0,61131	0,60411	0,0072	2
k=10	0,60922	0,60178	0,00744	2



To solve the above problem, we used C# programming language and Matlab mathematical package.

References.

1. Samarskiy A.A., Galaktionov V.A., Kurdyumov S.P., Mikhailov A.P. Modes with aggravation in problems for quasi-linear parabolic equations. // Science -1987
2. Samarskiy, A.A.; Gulin, A.V. Numerical methods (in Russian) // Science 1989
3. Kalitkin N.N. Numerical methods // Science 1989