https://doi.org/10.26160/2474-5901-2021-27-46-49

SOLVING NONLINEAR DIFFERENTIAL EQUATIONS IN MATLAB

Semenova M.N., Popova U.V.

Keywords: math modeling, differential equation, MatLab, Simulink, error.

Abstract. Mathematical modeling of differential equations as applied to problems of physics in the MatLab/Simulink environment is considered. The MatLab software package is designed to provide analytical and numerical solutions for various mathematical problems and simulate complex technical objects and systems. The Simulink app is one of the tools within the MatLab package. It has major structured, object-oriented, and visual programming capabilities. When solving the second-order non-linear differential equation, we observed small result deviations due to the method selected and the integration step.

РЕШЕНИЕ НЕЛИНЕЙНЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ В СРЕДЕ MATLAB

Семёнова М.Н., Попова У.В.

Ключевые слова: математическое моделирование, дифференциальное уравнение, MatLab, Simulink, погрешность.

Аннотация. Рассмотрено математическое моделирование дифференциальных уравнений применительно к задачам физики в среде MatLab/Simulink. Пакет программ MatLab предназначен для аналитического и численного решения различных математических задач, а также для моделирования сложных технических объектов и систем. Одним из инструментальных средств пакета MatLab является приложение Simulink, которое оснащено мощными возможностями структурного, объектно-ориентированного и визуального программирования. При решении нелинейного дифференциального уравнения второго порядка получили небольшие отклонения результатов, связанные с выбором метода и шагом интегрирования.

Almost all problems of contemporary science can be reduced to solving differential or integral equations [1]. Differential equations are widely used in various branches of mathematics and related sciences like mechanics, physics, engineering, chemistry, biology, economics, etc. [2]. This is due to the fact that such equations can be used to solve many problems in the world around us if they are related to some parameters and their numerical changes due to changes in time, coordinates, and other parameters [3].

The purpose of this paper is mathematical modeling of differential equations as applied to problems of physics in the MatLab/Simulink environment.

Review the simulation of a free and forced oscillation problem for a mathematical pendulum. In the general case, the forced oscillation equation looks like $\frac{d^2y}{dx^2} = p(x)\frac{dy}{dx} + q(x)y + f(x)$. If the oscillation system is impacted by the external momentum and the force of viscous friction, the differential equation describing the oscillations of the pendulum will be expressed as follows:

$$f(t) = \frac{d^2\varphi}{dt^2} + 2\beta\frac{d\varphi}{dt} + \omega_0^2\theta_0 Sin(\omega t), \qquad (1)$$

where φ is the angle of pendulum deflection from the equilibrium position, t is the time, β is the damping coefficient, ω_0 is the natural frequency of the undamped pendulum oscillations, and ω and θ_0 are the frequency and the torque angular amplitude that affects the pendulum respectively.

Due to the non-linear nature of differential equation (1), analyzing it stipulates some mathematical difficulties. Thus, it is feasible to apply the Runge-Kutta numerical method for solving this equation. If we consider the phase trajectory of damping (ω=0) and forced (ω=1) oscillations with $\omega_0=3.13$, $\theta=12^0$, $\beta=0.4$, $\omega=1$, $\varphi(0)=40^0$ and $\varphi'(0)=0$, we can study the pendulum oscillations. The structure chart for the model is shown in Figure 1 (a) and the solution results are presented in Figure 1 (b).

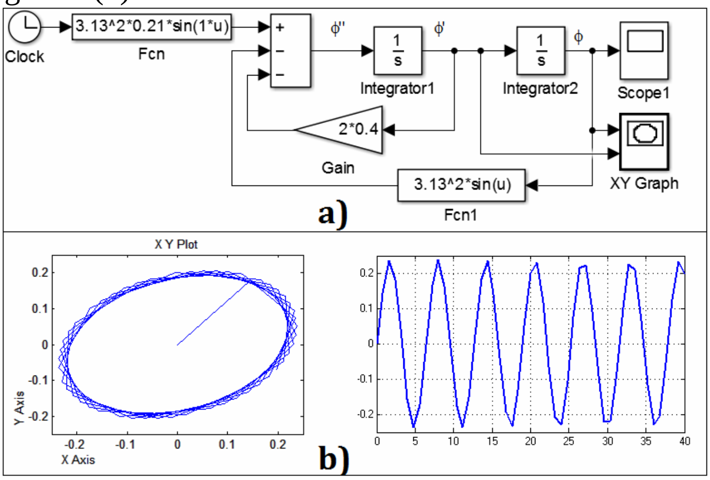
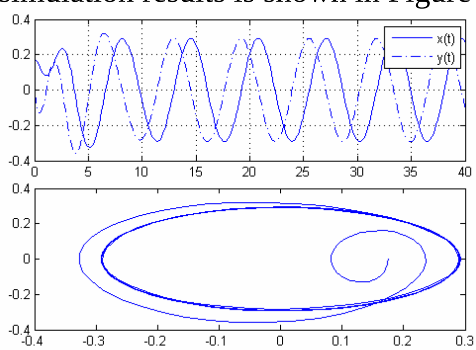


Fig. 1. Structure chart (a) and solution result (b) for the mathematical pendulum differential equation

For reference, we provide the solution to this problem in MatLab. Having denoted $x(1)=\varphi(t)$, $x(2)=\varphi'(t)$, we get the vector function of the right side of the equation system $\{x(2), -2\beta x(2) - \omega_0^2 Sin(x(1)) + \omega_0^2\theta_0 Sin(\omega t)\}$. Make a file function where we specify the vector of the system right side and a script file for the program code. The program code with the simulation results is shown in Figure 2.

```
function F=fun(t,x)
global omega0 theta0 beta omega
F=[x(2); -2*beta*x(2)-omega0*sin(x(1))+omega0*theta0*sin(omega*t)];
end
```

```
clc; clear
global omega0 theta0 beta omega
omega0=3.13; theta0=12*pi/180; beta=0.4; omega=1;
y0 = [10*pi/180 0]; % initial condition vector
[t, y]=ode45(@fun,[0 40],y0); % system solution
subplot(2,1,1); plot(t, y(:,1), t, y(:,2), 'b-.', 'LineWidth', 1)
grid on; legend('x(t)','y(t)'); subplot(2,1,2)
plot(y(:,1),y(:,2),'b','LineWidth',1) % phase trajectory plot
```



Published with MATLAB® R2014a

Fig. 2. The program code for the solution of mathematical pendulum differential equation in MatLab

Consider now a second-order non-linear differential equation $y''(t) = \frac{y'}{x} + \frac{(y')^2}{y} + 4yCos(x)$ with initial conditions $y(1)=1$ and $y'(1)=1$. The model structure chart will look as shown in Figure 3 (a). In the Math Function block (Math Operations library), we get the squared value of signal $y'(x)$. We divide the signal obtained in the Divide block by the value of signal $y(x)$ and send the result to the Sum block. In the Divide1 block, we divide the derivative signal of the function by the value of x and send the result to the Sum block. From the continuous linear Ramp signal in the Fcn block, we get signal $4\cos(x)$, multiply it by signal $y(x)$ in the Product block, and send the result to the Sum block. As a result, we get the required differential equation at the input of the first integrator block.

Launch the model and present the solution results. Figure 3 (b) shows the phase trajectory of the function and Figure 3 (c) shows the function charts for $y(x)$ and $y'(x)$ depending on the time.

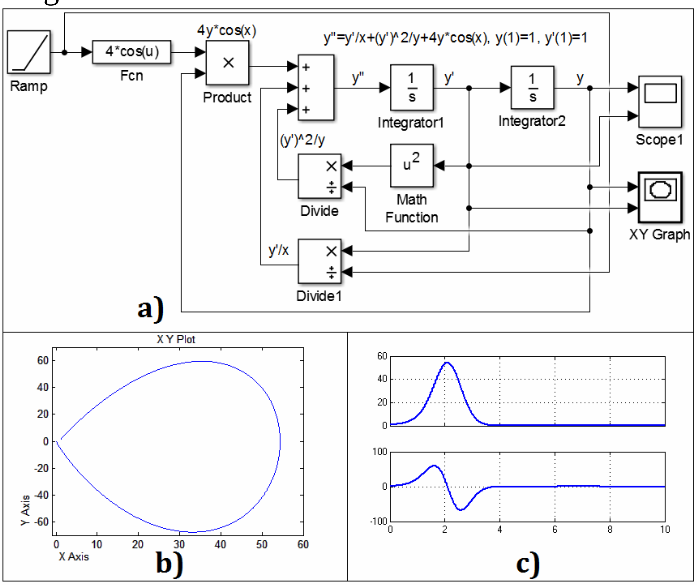


Fig. 3. The solution of the non-linear differential equation: the model structure (a) and simulation results (b,c)

As mentioned above, the selection of the solving method for differential equations impacts the accuracy of modeling results. In this research work, we used one-step explicit Runge-Kutta methods of the 2nd, 3rd, 4th, and 5th orders. Modeling time and step also have a significant impact on the result accuracy. In Solver options, there are two integration step parameters: Max step size and Min step size. In both cases, it is set to auto by default. In this case, the value of the Max step size is (Stop time – Start time)/50. This value is often too big, the modeling result has nothing to do with the real process. This can explain small discrepancies between the modeling results obtained via Simulink (Figure 1 (b)) and the MatLab programming tools (Figure 2). Even though the parameters used in these two options were identical.

In this work, the method of mathematical modeling was used to study differential equations as applied to problems of physics in the MatLab/Simulink environment. We reviewed the character- and number-driven solutions for these problems. We described the blocks from different Simulink libraries. We provided a description of the procedures and results of simulating non-linear differential equations of second order.

Список литературы / References

1. Murzaev R.T., Semenov A.S., Potekaev A.I., Starostenkov M.D., Kulagina V.V., Dmitriev S.V. Spatially Localized Oscillations in Low-Stability States of Metal System // Russian Physics Journal. 2021. Vol. 64. Is. 2. P. 293-301.
2. Babicheva R.I., Semenov A.S., Soboleva E.G., Kudreyko A.A., Zhou K., Dmitriev S.V. Discrete breathers in a triangular β -Fermi-Pasta-Ulam-Tsingou lattice // Physical Review E. 2021. Vol. 103. Is. 5. No. 052202.
3. Semenov A.S., Semenova M.N., Bebikhov Y.V., Yakushev I.A. Mathematical Modeling of Physical Processes in Metals and Ordered Alloys // Smart Innovation, Systems and Technologies. 2022. Vol. 247. P. 437-449.

Семёнова Мария Николаевна – кандидат физико-математических наук, старший преподаватель, mariya_semyonova86@mail.ru	**Semenova Maria Nikolaevna** – candidate of physical and mathematical sciences, senior lecturer, mariya_semyonova86@mail.ru
Попова Ульяна Витальевна – студент, ulyanapopovavit@mail.ru	**Popova Ulyana Vitalievna** – student, ulyanapopovavit@mail.ru
Северо-Восточный федеральный университет им. М.К. Аммосова, Якутск, Россия	North-Eastern Federal University n.a. M.K. Ammosov, Yakutsk, Russia

Received 30.11.2021