Dynamic Simulation of Theoretical Mechanics Experiments using Maple

Yanxiang Gong*, Juan Guo, Feng Li

College of Physics and Electronic Engineering, Taishan University, Taian 271000, China.

Abstract. Some experimental phenomena in theoretical mechanics are not suited for realization in an ordinary laboratory. Virtual experiments in theoretical mechanics can be realized using Maple, a scientific computing software. Although most previous studies have only drawn static images of the trajectories of objects, this thesis investigates the use of Maple's animation capabilities to achieve dynamic simulations of theoretical mechanics experiments. The benefits of dynamic simulation are that the trajectory of the object over time can be seen and the motion characteristics of the object, such as the direction of approach of the incoming track, can be fully understood. The dynamic simulation of theoretical mechanics experiments will fully mobilize students' interest in learning, transform abstract physical models into visual physical images and deepen their understanding of physical phenomena. In addition, virtual experiments enable students to train their programming skills and improve their scientific research literacy.

Keywords: Maple, theoretical mechanics, educational technology, animation simulation.

1. Introduction

Physics experiments are an important part of physics teaching and one of the three main research tools in physics (the other two are theoretical analysis and numerical simulation), but some experiments are not suitable for implementation in ordinary laboratories, such as the Foucault pendulum experiment in theoretical mechanics and the perihelion inlet phenomenon of planets. As a result, the understanding of abstract physical problems becomes a bottleneck for learning, which affects the learning effect, due that students often only know the analytical derivation process for some mechanical phenomena, but not the physical images. Describing the physical image of a theoretical mechanics problem vividly with virtual experiments is the best way to learn theoretical mechanics. Rather, it inspires our interest in learning and deepens our understanding of physical theory, while allowing us to learn about simple programming, drawing and data processing [1, 2]. Lots of previous studies introducing virtual experiments made static images for mechanics experiments, however. It leads to students knowing the trajectory of the object, but not being able to see dynamically and vividly the trajectory of the object over time. Dynamic images will be able to give more information, for example, the incoming direction of the object's trajectory, its instantaneous position, etc.

It is also one of the issues to be considered in depth when choosing the right programming language for a virtual experiment. It is difficult to master traditional scientific computing languages such as C and Fortran, which involve a lot of repetitive and complex programming work and are therefore not suitable for undergraduate students. More advanced and powerful computational languages such as Maple, Matlab, Mathematica, and Mathcad are common choices [3, 4]. Compared to other software, Maple selected has the following advantages: First, the programming is easy to learn and simple, which is incomparable to the usual computing languages. C and Fortran can only be used for numerical operations, and Matlab is not as specialized as Maple, although it can do symbolic operations, for example, symbolic variables or symbolic expressions must be defined in advance. Matlab's symbolic operations start with the Maple kernel, whose function names are obvious and easy to remember. It can compute huge integers exactly. Secondly, there is a high degree of integration of functions. It combines image processing, symbolic operations, and numerical calculations in one. It is something that is not available in ordinary high-level languages. Theoretical mechanical problems such as the perihelion motion of Mercury and the scattering of alpha particles can be easily simulated with Maple, which offers powerful drawing capabilities, including 1D, 2D and 3D image drawing. Particularly, dynamics simulation of mechanical problems can be easily achieved by Maple's animation and various useful drawing packages. In this thesis, two typical theoretical mechanics problems are used as examples to study how to...
realize "dynamic" simulations of theoretical mechanics experiments using the Maple language.

2. Dynamic Simulation of Mercury's Perihelion Approach

One of the important experimental verifications of Mercury's perihelion progression for general relativity. In case only the Newtonian gravitational interaction between the Sun and Mercury is considered (both as masses), Mercury's orbit is a closed ellipse. But progression will occur in its orbit if relativistic corrections are considered. In the polar coordinate system, the differential equation for Mercury's orbit is [5].

\[
\frac{d^2 u}{d\theta^2} + u = \frac{Gm_0}{h^2} + \frac{3Gm_0}{c^2} u^2
\]

(1)

Where \(u = 1/r\), \(G\) is the universal gravitational constant, \(m_0\) is the mass of the Sun, \(h\) is twice the speed of Mercury's grazing surface, and \(c\) the speed of light in a vacuum. After parameters: \(a = \frac{Gm_0}{h^2}, b = \frac{3Gm_0}{c^2}\) are introduced, the equation can be rewritten as

\[
\frac{d^2 u}{d\theta^2} = -u + a + bu^2
\]

(2)

Let \(x = u, y = du/d\theta\), rewrite the above second-order differential equation as a system of first-order differential equations, i.e.

\[
\begin{align*}
\frac{dx}{d\theta} &= y \\
\frac{dy}{d\theta} &= -x + a + bx^2
\end{align*}
\]

(3)

The parameters need to be chosen appropriately to exaggerate the orbital procession due to the small actual correction term when solving numerically. A result is shown in Figure 1.

\[\text{Figure 1. Dynamic Trajectories of the Perihelion Approach of the Mercury, with the cross marks varying with angle (0 as a function of time)}\]

The procedure, implemented in Maple, is as follows:

```maple
restart;
with(plots);
a := 1;
b := 0.01;
eq := diff(u1(theta), theta) - u2(theta) = 0,
diff(u2(theta), theta) + u1(theta) - a - b*u1(theta)^2 = 0;
sol := dsolve(\{eq, u1(0) = 0.1, u2(0) = 0\}, \\
\{u1(theta), u2(theta)\}, numeric);
Mercury := proc(z)
local U1, X, Y, note;
global theta, sol;
U1 := eval(u1(theta), sol(z));
X := 1.*cos(z)/U1;
Y := 1.*sin(z)/U1;
plot([\[X, Y\]], style = point, axes = boxed, color = blue);
end proc;
animate(Mercury, \[theta\], theta = 0..8*Pi, frames = 500, 
trace = 500);
```

3. Dynamic Simulation of Foucault Pendulum Motion

The Foucault pendulum, a device to show the rotation of the Earth, was first made in Paris in 1851 by the French physicist Foucault. The special trajectory of this pendulum is due to the rotation of the Earth and the influence of the pendulum by the Coriolis force. Its physical model is as follows [6]. The coordinate system is shown in Figure 2.

\[
\begin{align*}
\ddot{x} &= 2m\omega y\sin\lambda - p^2x \\
\ddot{y} &= -2m\omega x\sin\lambda - p^2y
\end{align*}
\]

(4)

(5)

where \(m\) is the mass of the Foucault pendulum, \(x, y\) are the velocity components in the Cartesian coordinate system, \(x^\prime, y^\prime\) are the acceleration components, \(\omega\) is the angular velocity of the Earth's rotation, and \(\lambda\) is the dimensional angle of the sphere where the Foucault pendulum is located. \(p^2 = g/l\), is the acceleration of gravity and is the pendulum length of the Foucault pendulum. A simulation result is shown in Figure 3.

\[\text{Figure 2. Schematic Diagram of the Earth's Coordinate System [6]}\]
Figure 3. Dynamic Trajectories of the Foucault Pendulum, with the cross markers changing dynamically with time

The procedure, implemented in Maple, is as follows:

```maple
restart;
with(plots);
lambda := (60*Pi)/180;
g := 9.8;
l := 67;
omega := 0.03;
sys := diff(x(t), t $ 2) - 2*omega*diff(y(t), t)*sin(lambda) + g*x(t)/l = 0,
        diff(y(t), t $ 2) + 2*omega*diff(x(t), t)*sin(lambda) + g*y(t)/l = 0;
sol := dsolve({sys, x(0) = 4, y(0) = 0, D(x)(0) = 0, D(y)(0) = 4}, {x(t), y(t)}, numeric);
drawFuke := proc(z)
    local X1, Y1;
    global x, y, sol;
    X1 := eval(x(t), sol(z));
    Y1 := eval(y(t), sol(z));
    plot([X1, Y1], style = point, scaling = constrained, color = red);
    end proc;
animate (drawFuke, [t], t = 0 .. 100, frames = 500, trace = 500);
```

4. Summary

The analysis in this thesis shows that dynamic experiments in theoretical mechanics can be easily produced using Maple's animation capabilities and the rich package of numerical calculation functions. It is specifically summarized as follows. First, the kinematic characteristics of the Foucault pendulum and Mercury's perihelion inlet can be dynamically displayed using Maple's solved differential equations and animation functions, making it easier for students to understand their physical images. Second, information such as the direction of the orbital approach and the instantaneous position of the object can be understood through dynamic maps; the influence of parameters such as initial conditions and dimensions on dynamic trajectory maps can be studied to gain a deeper understanding of physical phenomena. Third, Maple software is very suitable for experimental simulation of theoretical mechanics. Thus, it enables university students to cultivate an interest in learning theoretical physics, acquire programming knowledge, conduct early research training, etc.

References