Use of a computer in advanced mechanics – Principle of least action

Slavomir Tuleja*
Gymnazium arm. gen. L. Svobodu, Humenne, Slovakia

Edwin F. Taylor†
Massachusetts Institute of Technology, MA, USA

Jozef Hanc‡
Technical University, Kosice, Slovakia

(Dated: February 20, 2006)

The *ActionClockTicks* application is a simple interactive *Java* program that enables students to hunt for the worldlines of stationary action in several different scenarios ranging from a simple projectile motion in uniform gravitational field to a Moon shot. It is meant only as a proto-program for a general-purpose interactive software by means of which the students could create new scenarios of their own and apply one of the simplest and most powerful expressions of the classical mechanics—Principle of least action. We describe the numerical method for finding the worldline that has stationary action.

I. INTRODUCTION

Principle of least action is a very useful theoretical tool that except explaining Newton's second law[1] and Lagrange's equations,[2] links conservation laws and symmetry[3]. In addition it offers a way for a seamless transition from quantum mechanics to classical mechanics[4].

Unfortunately it is introduced only late in the curriculum. One of the obstacles for its earlier introduction is the mathematical complexity of variational calculus involved in its formulation. By contrast, the principle itself is conceptually simple.

In what follows we provide the mathematical formulation of the variational problem and show a possible strategy of how to avoid the mathematical complexity exhibited by its traditional solution.

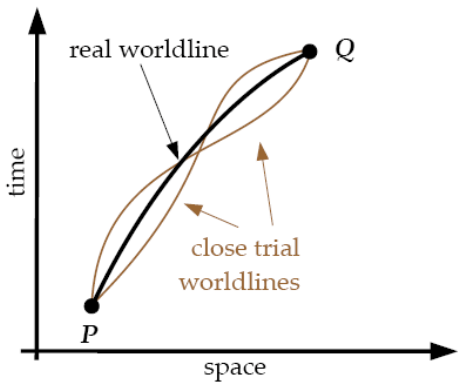

FIG. 1: Worldline in spacetime connecting the initial and final events P and Q. The black bold curve represents the real worldline taken by the particle. The two brown curves represent two close worldlines with almost the same action.

II. USING ACTION TO PREDICT MOTION

A. Mathematical formulation of the problem

Suppose the motion of a single particle is restricted to a plane. To find the worldline that takes the particle from an initial event P to a final event Q using the principle of least action, one has to compute a number called *action* for each possible worldline connecting events P and Q. Action is defined as the functional:

$$S = \int_{t_P}^{t_Q} L\left[x(t), y(t), v_x(t), v_y(t), t\right] dt,$$

where

$$L\left[x(t), y(t), v_x(t), v_y(t), t\right] = K\left[v_x(t), v_y(t)\right] - U\left[x(t), y(t)\right]$$

is the *Lagrangian.*

From the infinite number of possibilities the particle chooses the worldline connecting the starting and final events in spacetime for which the action S is *stationary.* It means that the action along any *close* trial worldline must be *approximately the same* as shown in Fig. 1.

The problem of finding the stationary-action worldline is usually addressed using rather advanced methods of variational calculus[5]. We aim to show an alternative approach to finding such worldline. We use computer as an aid.

B. Visualization of worldline

Before we start we need to solve a problem of computer visualization. How can we draw a *worldline in two-dimensional space?* Notice that one trajectory can represent many worldlines, as shown in Fig. 2! Trajectory tells us nothing about particle's speed. It tells us

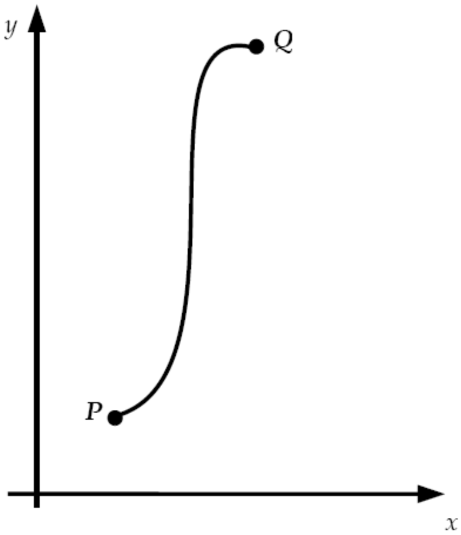

FIG. 2: Particle trajectory in space. There are infinitely many worldlines corresponding to this trajectory. For example the particle could travel with constant speed along the entire trajectory, or it could change its speed continually—being slow in curved portions of the trajectory and fast in the almost straight portion. Therefore trajectory tells us nothing about particle speed.

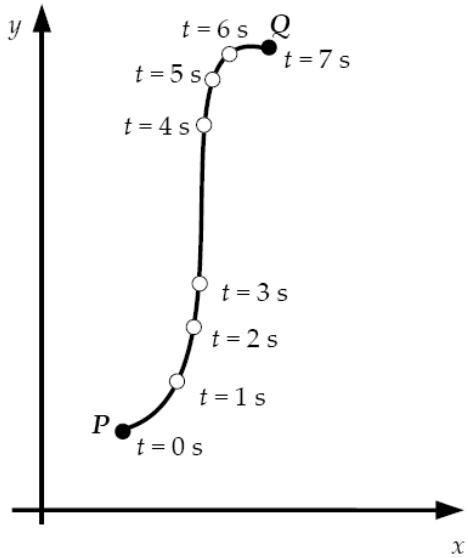

FIG. 3: Trajectory with imaginary stopwatch clock-ticks. The clock-ticks help us in seeing what the speed of the particle was in different portions of its trajectory. They provide timing for the particle. Numbered dots $0, 1, ..., 7$ represent clock-tick events.

only which points in space it was in, but doesn't tell us when.

One possible solution is to provide timing for the particle. Imagine the particle carries a clock making short clock ticks *spaced equally in time*. Each of the clock ticks can be regarded as an event. Draw the two-dimensional *trajectory in space* along with the dots representing the clock-ticks as shown in Fig. 3. The clock-tick dots tell us indirectly what the particle velocity was in different portions of its trajectory. The smaller the clock-tick interval, the better.

C. Finding of the stationary-action worldline on a computer

To represent such worldlines on a computer we have to make a small sacrifice. We approximate *worldline* segments between adjacent clock ticks by *straight lines*. The particle will have a definite velocity along each straight segment of worldline (and trajectory). The resulting trajectory is shown in Fig. 4.

We can compute action for the trajectory as the sum of actions along its segments (numbered 1, 2, ..., n):

$$S = \sum_{i=1}^{n} \Delta S_i.$$

Denote

$$\bar{x}_i = \frac{x_{i-1} + x_i}{2}, \quad \bar{y}_i = \frac{y_{i-1} + y_i}{2},$$

$$\bar{v}_{xi} = \frac{x_i - x_{i-1}}{\Delta t}, \quad \bar{v}_{yi} = \frac{y_i - y_{i-1}}{\Delta t},$$

and

$$\bar{t}_i = \Delta t \left(i - \frac{1}{2} \right)$$

the average x and y coordinates on the segments and the velocity components on the segments. Resulting expression for action can then be written as

$$S = \sum_{i=1}^{n} L\,(\bar{x}_i, \bar{y}_i, \bar{v}_{xi}, \bar{v}_{yi}, \bar{t}_i)\Delta t.$$

Thus action S becomes a function of x and y positions of the intermediate clock-tick events

$$S = S(x_1, x_2, ..., x_{n-1}, y_1, y_2, ..., y_{n-1}).$$

How can we now find a trajectory of stationary action? To find it, means to find such a set of x and y positions of intermediate events, that making a small change in value of any of them results approximately in no change in action. This is equivalent to demanding the following $2n - 2$ conditions being satisfied:

$$\frac{\partial S}{\partial x_i} = 0, \quad i = 1, 2, ..., n-1$$

$$\frac{\partial S}{\partial y_i} = 0, \quad i = 1, 2, ..., n-1$$

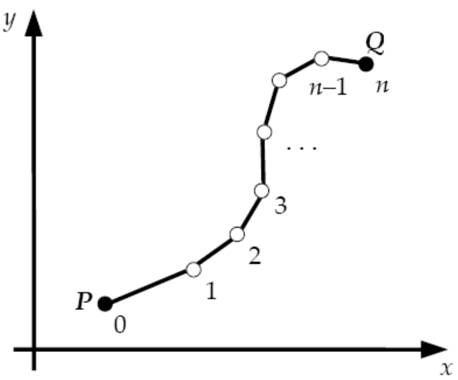

FIG. 4: Worldline approximated by connected straight segments as represented in the computer. Computer restricts itself only to this class of trajectories.

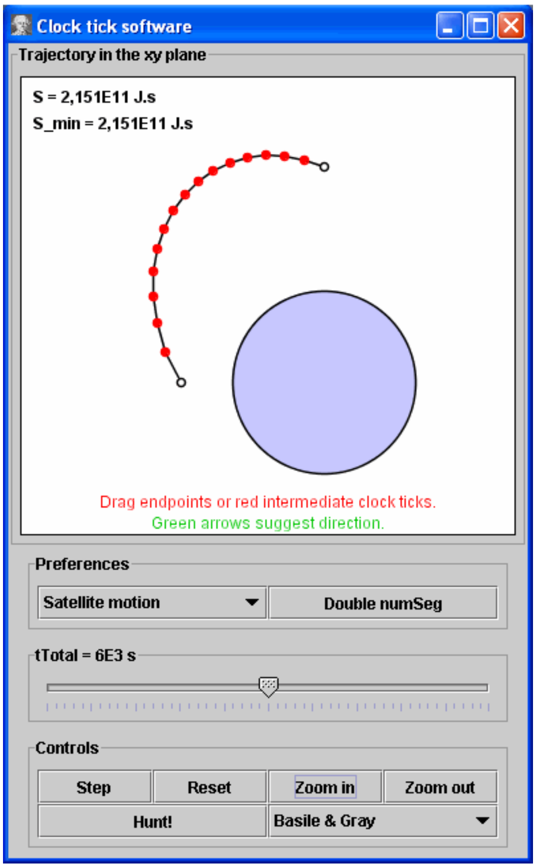

FIG. 5: A screen shot of the Java application *ActionClockTicks* we programmed to make searching for the stationary-action worldline interactive. The scenario shown on this screenshot is "Satellite motion". To make action stationary a user can *either* click-and-drag the intermediate clock-ticks of the trajectory *or* press the HUNT button. The later starts automatic search for stationary action.

The conditions have the form of the simultaneous system of $2n-2$ equations with $2n-2$ unknowns. In general

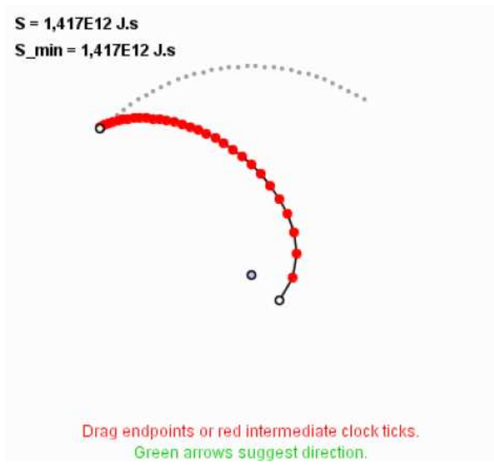

FIG. 6: This screen shot depicts the "Moon shot" motion scenario. The heavy black line with red dots is the spaceship trajectory with clock-ticks. The upper series of gray dots represents the Moon in different times. The blue dot in the center is the Earth. User first chooses the total time for motion using the slider at the bottom control panel of the applet. Then (s)he chooses the initial and final points on the trajectory. Finally (s)he hunts for the stationary-action worldline taking the spaceship from Earth to Moon.

this system is *nonlinear* and can be solved by *Newton's iteration method* generalized for functions of more variables. This is what is actually done in our interactive Java program described in the next subsection.

D. Interactive software

Fig. 5 shows a screen shot of the Java application[6] we created for students. The program allows them to find the worldline of stationary action either manually or automatically as they explore different motion scenarios. The scenarios are (1) the projectile motion in the uniform gravitational field near Earth, (2) the satellite motion near Earth, (3) the motion of a particle in potential $U(x, y) = ky^2$, and (4) the Moon shot (see Fig. 6).

III. CONCLUSIONS

We have created a simple software that enables students to predict two-dimensional motion of a particle. The student can make such predictions using an alternative to Newton's laws of motion, namely the Principle of least action. The principle is little known because mathematical competence required from the student if he wants to apply it to predict motion (in a traditional way) is rather complex.

However, today the computer offers a new approach to predicting particle motion using the principle of least action. Such approach was outlined in this paper. We were successful in replacing the mathematical complexity of variational calculus (1) by its reduction to simple calculus of functions with many variables and (2) by hiding even this simpler mathematical approach "under the hood" of the computer. Thus everything the student needs is the conceptual understanding of the Principle of least action and a sound ability to work with a computer.

The Principle of least action has an important role in later theoretical courses, one of them being quantum mechanics. The approach outlined in this paper can be used early[7] in physics instruction to introduce students to the concept of action.

Acknowledgments

The work was partly supported from the grant KEGA, Project No. 3/3005/05 "Teaching physics by programming models of physical phenomena using interactive software".

* Electronic address: stuleja@gmail.com
† Electronic address: eftaylor@mit.edu
‡ Electronic address: jozefhanc@yahoo.com

1 J. Hanc, S. Tuleja, M. Hancova, "Simple derivation of Newtonian mechanics from the principle of least action," Am. J. Phys. **71** (4), April 2003, pages 386 - 391.

2 J. Hanc, E. F. Taylor, S. Tuleja, "Deriving Lagrange's equations using elementary calculus," Am. J. Phys. **72** (4), April 2004, pages 510-513

3 J. Hanc, S. Tuleja, M. Hancova, "Symmetries and Conservation Laws: Consequences of Noether's Theorem," Am. J. Phys. **72** (4), April 2004, pages 428-435

4 J. Ogborn, E. F. Taylor, "Quantum physics explains Newton's laws of motion," Physics Education **40**(1), January 2005, pages 26-34.

5 Herbert Goldstein, Charles Poole, and John Safko, *Classical Mechanics*, 3rd ed. (Addison-Wesley, New York, 2002)

6 The program can be downloaded from `http://www.eftaylor.com/software/ActionClockTicks/`.

7 E. F. Taylor, "A Call to Action," Am. J. Phys. **71** (5), May 2003, pages 423-425.