CREATIVE MATH.
12 (2003), 95 - 101

A simulation of the action of terrestrial gravitation over a comet

Cristian Eduard Rusu

Abstract. This paper is a simple illustration of how a low-degree-difficulty abstract formula, concerning a physical phenomenon, can be implemented in an interactive and graphically suggestive way, with the use of a computer. A short historical view on the subject is included. Input data sets and graphic results are presented.

1. Introduction

The purpose of the simulation is to obtain an approximate representation of a comet's trajectory in the Earth's gravitational field, by using a simple algorithm that can be understood by any person familiarized with the mathematical expression of the universal law of gravitation and the laws concerning mechanical movement.

The simulation can generally be used to illustrate the motion of any body in the vicinity of Earth, such as asteroid, meteorite or artificial satellite, thus aiming towards a response to a problem frequently under debate.

The program, elaborated in the Borland Pascal 7.0 environment, draws the trajectory graphically based on input data and computes certain related mechanical elements.

2. A short history concerning the subject

The universal law of gravity was discovered and stated in 1687 by Newton (1642-1727) and has stood as a fundament of classical Physics. In Celestial Mechanics, its applications are overwhelming, due to the fact that it managed to explain the laws of planetary movement stated by Kepler (1571-1630), thus completely systemizing Solar System Mechanics.

Of course, Newton is not the first of the great thinkers that have laid the basis of both Physics and our knowledge of our planet and the near Universe. The first philosopher known to us, whose ideas are worthy of notice, was Pythagoras (approx. 580-500 BC). He is known to have been the first to claim the spherical shape of Earth. But the first who has sustained this idea with what we now call scientific facts was Aristotle (384-322 BC), around 340 BC. Once the Earth's sphericity was proved, research was begun,

Received: 15.03.2003; In revised form: 21.11.2003
Key words and phrases. *Astronomy, comet, gravity, computer science, physics.*

concerning its dimension. The result of Eratosthenes (approx. 275-195 BC) is worthy to consider. He proposed a value of 12800 Km for Earth's diameter, very close to the one accepted today.

After the Earth, as a whole, was "recognized" as a physical entity (matter), which was no sooner than the Renaissance, its physical properties – the most important of which is mass – were studied. Henry Cavendish (1731-1810), the scientist who determined the value of the gravitational constant, also computed the mass of the Earth. He published a value of 6×10^{24} kg, remarkably correct for his time.

One great progress of Astronomy, after Renaissance, was the explanation of some phenomena that have both troubled and fascinated human imagination even before antiquity. One of such phenomena is the apparition of comets, considered since time immemorial as calamity forecasters. Aristotle and Galileo are remembered to have rationally interpreted this phenomenon. Unfortunately, they considered it to be just a burst of atmospheric gas. The one who managed to discover the true nature of a comet and also applied to it the laws of mechanics was Edmund Halley (1656-1742), known for the comet bearing his name.

Astronomy, and especially Celestial Mechanics, has known a crescendo evolution, both through the development of observational devices (terrestrial and space telescopes), and through the power increase of computer systems. More and more advanced computers and precision application algorithms have allowed virtually absolute precision trajectory calculus, both for celestial bodies inner and outer the Solar System. Today, the progress of the oldest science is closely linked to the one of computers.

3. Mathematical fundaments and algorithm design

In the gravitational field of a central mass, a body describes a plane trajectory. This can easily be represented on a flat computer screen. 2D - graphics is thus sufficient, with the advantage of small system requirements.

The basic mathematical formula implemented by the algorithm is the law of gravity, in the form:

$$\begin{cases} a_x = \dfrac{d^2x}{dt^2} = \gamma\dfrac{M}{x^2} \\ a_y = \dfrac{d^2y}{dt^2} = \gamma\dfrac{M}{y^2} \end{cases}$$

where

a_x, a_y	acceleration on the axes,
x, y	distance between bodies, on the axes,
t	time,
γ	gravitational constant,
M	mass of the central body (Earth).

The concise version of the algorithm, very intuitive, is as follows.

The distance between the two bodies (introduced in pixels) is converted into kilometers and then the actual acceleration of the comet in the Earth's gravitational field is computed. Then, using the law of accelerated movement, the new position of the comet is determined, after a time interval of constant length t. The same procedure is used until one of the exit conditions is met.

Theoretically, as the time between the calculus of two successive positions of the comet – specified by t – decreases, the trajectory should gain precision. However, the field of the screen is composed of pixels, whose finite number and dimension obviously cannot be overlooked. This is why the parabolic or elliptic trajectory will be considered and represented as a succession of broken lines – independently of the used algorithm. After running the program a certain number of times, the value of 500 seconds was attributed to t. The use of another value t can have some "odd" consequences. For instance, if the value of t is too high, the broken lines that assemble the trajectory become obvious, and if it is too low, the trajectory becomes a square. Not only the trajectory, but also numerical output data are closely linked with the value of t. An example of "wrong" results is that running the program for both a large t and a large comet velocity in the Earth's direction, the comet may pass it without the occurrence of any collision.

The algorithm leads to the following results:

- *Case 1:* The comet leaves the gravitational field (whose space extension is comparable to the screen resolution). The escape speed and the corresponding time are computed.
- *Case 2:* The comet strikes the earth. The collision speed, the corresponding time and the energy released are computed.
- *Case 3:* The comet enters a stable orbit (Earth's satellite). The corresponding message is printed. This is the case if, after computing the trajectory for a finite number of positions, the precedent cases do not apply.

There are 4 situations in which the input data is incompatible with the execution:

- There is a syntax error in the input data.
- The coordinates of the comet or the Earth outrun the screen resolution.
- The comet overlaps the Earth
- The comet speed is too high.

4. Implementation

The algorithm implemented in Borland Pascal 7.0 yields a relatively small dimension program which needs small memory and system resources:

	System requirements	**Graphic mode resolution**
Minimal resources	100 Mhz	319×199
Optimal resources	266 Mhz	639×479

The differences between program running on computers endowed with minimal and optimal resources are negligible.

The running time extends from some tens of a second to a few seconds, depending on the input data, and so, the trajectory is visible as the computer draws it. For instance, the consequences of the second Kepler theorem are visible, if the proper conditions are met.

The astronomical and physical constants declared in the constant section (Pascal application) can be easily modified. Of course, the comet and the Earth are celestial bodies arbitrarily considered, in order to achieve a visible illustration of the effect of gravity. They can be replaced with other bodies by changing the characteristic astronomical constants involved.

For simplicity, the Earth is represented as immobile with respect to the screen. The graphical coordinate system is also fixed, being centered in the upper-left corner of the screen (standard Turbo Pascal graphic mode setting).

The yielded results (the output data) do not pretend to be scientifically rigorous, but approximate and graphically suggestive.

5. User interface

The program consists of two components connected by an interface: the data file (gravitat.txt) and the algorithm implementation file (gravitat.exe).

The input data, as required in the data file, are represented as a numerical set, which consists of:

- the coordinates of the Earth on the screen,
- initial comet coordinates,
- the two components of the velocity vector,
- mass of the comet.

The output data consist of the graphical representation of the trajectory and certain related mechanical elements printed in the text file.

6. Application results

Figures 1 and 2 present some application results.
Input data format:

- Coordinates of the Earth on the screen - 2 integers within the screen resolution
- Initial coordinates of the comet - 2 integers; 1 pixel means 200 km
- The components (on the axes) of the comet velocity vector - 2 reals; a unit means 1 km/s; values within the interval of [-5,5] are recommended

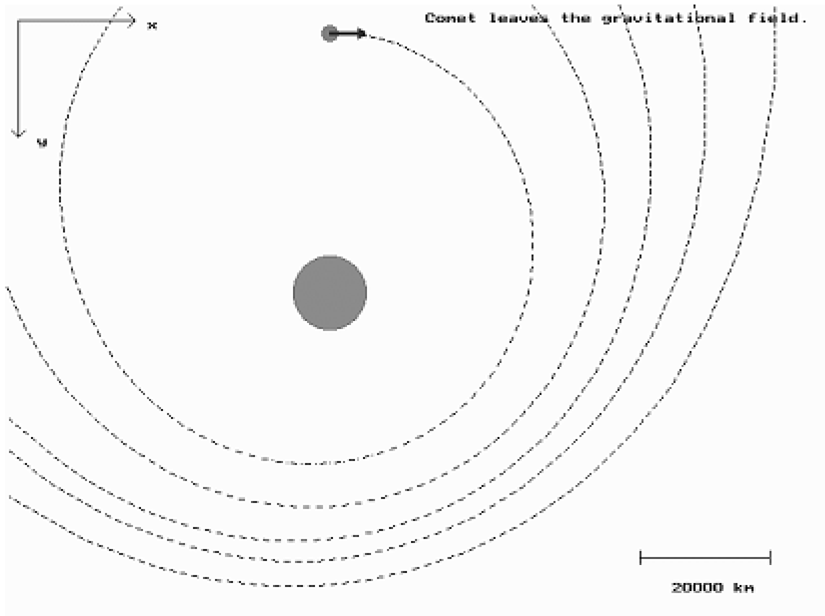

FIGURE 1. The comet leaves the gravitational field.

- Mass of the comet in tones.

Application 1.

Input data:

250 225
250 25
2.7 0
1000000

Results:

See figure 1.
Comet leaves the gravitational field after 271.1 hours, with the velocity of 1.6 km/s.

Application 2.

Input data:

350 225
100 150
0 1
1000000

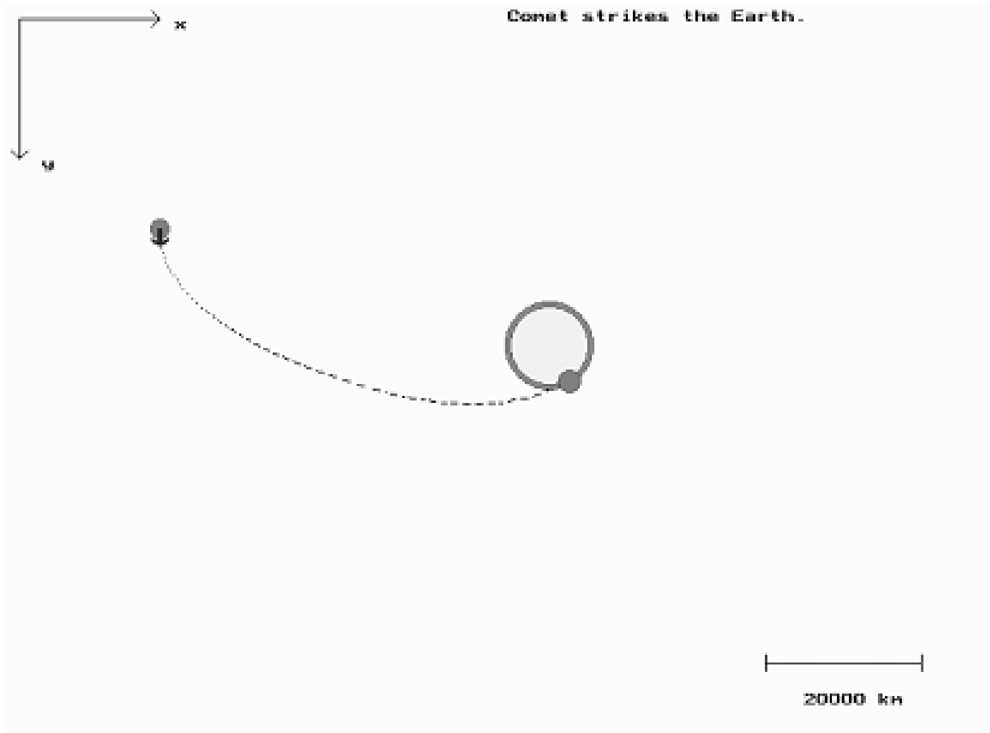

Figure 2. The comet strikes the earth.

Results:
See figure 2.
Comet strikes Earth after 5.8 hours, with the velocity of 12.1 km/s. The released energy is 1231332.2 kJ.

7. Conclusions

The purpose of the program presented in this paper is to offer an intuitive view on the simulated physical phenomenon. It actually yields an interdisciplinary link between Physics and Computer Science.

Due to the simplicity of the algorithm, the program may easily be used for didactical purposes.

References

[1] Asimov I., *Despre Pământ şi cer*, Elis Publishing House, 1996

[2] Chiriac D., *Mic dicţionar de personalităţi ale ştiinţei*, Editura Ştiinţifică şi Enciclopedică, Bucureşti, 1977

[3] Crişan C., *Ghid de utilizare Turbo Pascal 5.0-5.5*, Romanian Software Comp., 1991

[4] Rusu C. E., *Proiect pentru atestatul de informatică*, Gh. Şincai National College, Baia Mare, Romania, 2003

[5] Sass I.H.A., *Astronomy and Astrophysics - Sciences of the Univers* (in Romanian), Risoprint, Cluj-Napoca, 2002

MEXT Scholarship Grantee
Osaka University, Japan
E-mail address: eduardrusu@yahoo.com