Contents

COMP1730/6730 2023 Semester 1, Project-2 (Physics)

The assignment is an exercise in elementary physics which has a very surprising connection with number theory. This project will be suitable to COMP6730 students, or for Honours students studying Science and Mathematics. Background knowledge (or additional research) regarding the mechanics of one-dimensional two-body collisions (when the motion happens on straight line) and conservation laws (energy and momentum) is required. The estimated size (including the code comments and docstrings) is approximately 250–300 lines of code. The time effort is about 15 hours.

Caveat: choose this project if you posses reasonable knowledge of mechanics (as covered by the ANY first semester Physics Course).

The problem

The number π is undoubtedly the most famous in mathematics. There are multiple algorithms to compute it. With the advent of computers mathematicians have been competing to compute more and more digits in the expansion of π. They believe this is an important quest, and we shall trust them. But the use of supercomputers in the pursuit of higher precision is not the only way to explore the nature of this constant. Sometimes, it's not the efficiency of algorithm which computes π, but the very nature of it which allows a new discovery. One such algorithm you will be implementing in this assignment.

When the algorithm inventor, Grigorii Galperin, started to discuss it at various seminars in the US 20 years ago, nobody believed him at first. However, the listeners were soon fully convinced and impressed because of sheer elementary nature of it. "Elementary" doesn't mean simple or trivial — the algorithm is truly ingenious — it means that only basic, *elementary* knowledge is required to understand and verify it.

The Galperin construction and the claim are described as follows:

- Consider two objects (Galperin used the name "balls" because he has been studying billiards for his entire academic career, but one can also use the name *blocks*, or *sliding blocks*) as material points (that is, their shape and size is irrelevant). They move (slide) along a semi-axis bounded by an infinitely massive wall (located at $x = 0$), collide with each other and the "inner" object also collides with the wall. The Fig. 1 gives the entire description of possible combinations of object velocities (this Fig. is borrowed from ref. **3**).

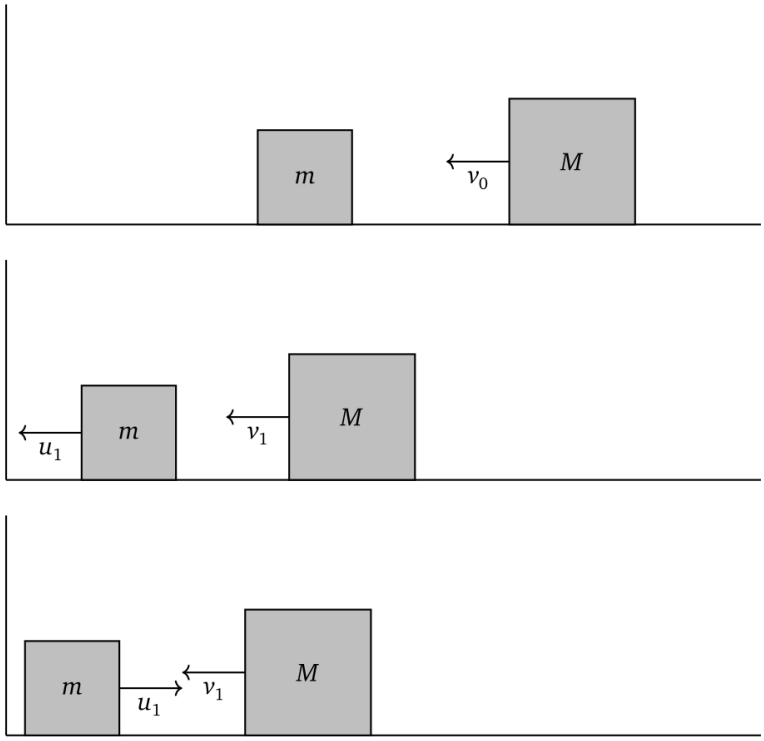

Figure 1: Sliding blocks on a semi-axis allow to compute π.

- The block on the right is heavier than the block on the left (between the wall and the first block), $m_1 > m_2$, the mass ratio is $\frac{m_1}{m_2} = 100^N$, where N is a positive integer.
- The blocks collisions are *absolutely elastic*, meaning that the entire kinetic energy (the sum of the blocks kinetic energies) is preserved. During each block-to-block collision the momentum is also preserved. When the inner block collides with the wall, these collisions are also elastic, such that the block velocity changes on its negative as the result. Because the objects are considered point-like, the time of every collision is zero, that is the change of velocities occurs instantaneously.
- The process starts with the heavier blocks given a boost, making it move towards the lighter block which is initially at rest. The blocks collide, the heavier block transfers part of its momentum to the lighter block, which starts moving towards the wall, bounces off it, moves towards the heavier block (which at first still moves leftwards albeit with a smaller velocity); the blocks collide again, the lighter one starts moving towards the wall again, this time a bit faster, and the entire sequence of collisions repeats again, and again... until the heavier ball changes the direction of its motion on the opposite, and slowly at first starts moving away from the wall. The lighter block moves faster, it catches up if the heavy blocks, passes some of its momentum to it, as the result the speed (absolute velocity) of the heavy blocks gets higher, while the lighter one slows down. It may take a number back-and-forth collisions with the heavy block and the wall before the speed of the lighter block becomes smaller than the speed of the heavy block, at which point the balls have collided for the last time (from now on, the lighter block will never catch up with the heavy one).
- What Galperin proved is quite unexpected — the total number of collisions (both block-to-block and block-to-wall) which occur in the process, if written in the decimal format when the leftmost digit is treated as the integer part, and the remaining digits are treated as digits of the fractional part, is *exactly the number* π written with the precision to N digits.

It is recommended that you watch the video in Ref. **1** to observe the process. The video also contains an elementary proof of this statement, which is useful although not necessary to follow through.

The programming tasks

1. You will write a program which will compute the number of collisions in the described system as a function of the number N (the choice of the initial speed which is given to the heavy ball is not important, it will affect the total time between the first and the last collisions, but not their number).

2. You may define as many necessary functions as you need, for example:

    - to determine the time needed for two moving bodies to collide given their initial positions and velocities;

- the outcome of an elastic collision between two bodies given their mass and momenta
- the outcome of an elastic collision of a body and an infinitely massive wall (if your Physics knowledge allows you to conclude the result of such collision without making computation, then your choice of this project is justified).

3. A particular care is needed to count collisions in the "medium" part of the process, because they happen at a high frequency rate (for large values of N). Overall, the problem will be computationally demanding when we choose large value of N (to achieve high precision in computing π). At small values of N, the program execution should not present computational difficulties.

4. Create an animation of the process which resembles the video in Ref **1**. Use the animation module `matplotlib.animation` in the Matplotlib library.

References

1. The "3Blue1Brown" Youtube channel video Why do colliding blocks compute pi?

2. An article by the original discoverer of this construction: G. A. Galperin. 2003, *Regular and Chaotic Dynamics*, **8**, p. 375

3. An expository article (by two authors closer to home): M. Z. Rafat and D. Dobie, Throwing π at a wall. The pictures used in this paper are taken from that article.

4. Examples in which a Matplotlib animation is produced, for example a double pendulum.