

Programming Problem for the 2000 Comprehensive Exam

Out: Monday 1/24/00 at 9:00 am

Due: Friday 1/28/00 at 5:00 pm

**Department of Computer Science
Brown University
Providence, RI 02912**

1.0 The Task

This year you are going to simulate a toy called “romp” (Randomly oscillating magnetic pendulum). Romp consists of a five inch square metal base with an pendulum support starting at one corner, going up eight inches with a perpendicular support that hangs over the center of the base. From this support hangs a pendulum that is free to move in any direction. The pendulum has a magnet at its end which lies approximately 0.75 inches above the metal base. In addition, the toy comes with nine round magnets, each about a half inch in diameter and an eighth of an inch high. (The magnet at the end of the pendulum is identical to these.) The base is marked out with a half inch black dot at the center, concentric circles with diameters of 1, 2, 3, 4 and 5 inches, and black dots on the outermost circle at the diagonals. A picture of the base and a side view showing the base and the pendulum are shown in Figure 1.

To play with the toy, one places some or all of the nine magnets on the base, positions the pendulum somewhere over the base, and lets it go. The magnets can be positioned either up (attracting the pendulum) or down (repelling the pendulum). They can even be stacked (although you can check and make this an error condition if you wish). The resultant motion of the pendulum is generally chaotic, with the pendulum attempting to reach the center due to gravity and the various magnets on the base pushing or pulling it at the same time.

Your job is to write a simulator of romp. The simulator should allow the user to place magnets on the base and then set an initial position and velocity for the pendulum. The simulation should then show the user the motion of the pendulum until either the user stops it or it stops moving.

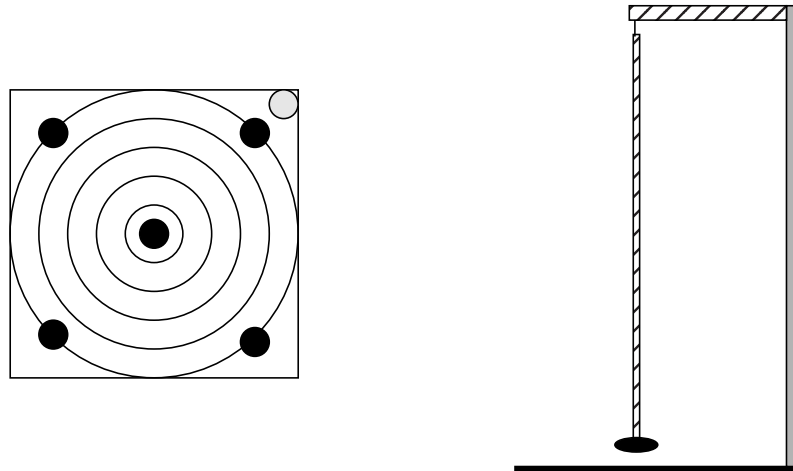


FIGURE 1. The base and a side view of ROMP.

2.0 Physics

There are three basic forces that act on the pendulum. The first is gravity. The pendulum has some mass and wants to be at the bottom. At any point, the force on the pendulum from gravity is pulling it toward the center of the base and is roughly proportional to the distance the pendulum is from the base. The second force is magnetism. Each magnet on the base will attract or repel the pendulum depending on the orientation of the magnet and the distance the pendulum is from the magnet. The final force is friction.

The force of gravity on a pendulum at an angle of Θ is:

$$F = -m(g \sin \Theta)$$

where m is the mass of the pendulum, g is the gravitational constant, and l is the length of the pendulum. (Note that for the angles typically used in pendulums, this is approximated using Θ directly rather than $\sin \Theta$. This approximation is valid, assuming Θ is given in radians, to about three decimal places for angles of less than 10 degrees. Your simulation can use either approach.) Note also that the force is toward an angle of zero (the pendulum being straight down). The period of the pendulum is then approximately $2\pi\sqrt{l/g}$. This is independent of its mass or its current position.

The attractive or repulsive force of two magnets is given as:

$$F = K \left(\frac{q_1 q_2}{r^2} \right)$$

where K is a constant, the q_i 's represent the strength of the magnetic fields of the two magnets, and r is the distance between the two magnets. The force is on the line between the two magnets. All the magnets in romp, including that in the pendulum, have equal strength fields.

While romp tries to minimize frictional forces, they still exist. You can assume that this is a damping force. Such a force is proportional to the velocity at any given point, i.e $F = -ev$, where e is the (positive) damping coefficient and v is the current velocity. Note that the direction of the force is opposite to that of the current velocity.

While you have some of the constants, you don't have all of them (i.e. K, q, g, m, e are all unknown). The exact values of these don't really matter as long as the relative values are right. (For example, you can assume $K = 1$ if you can adjust q accordingly.) To get the relative values correct, you should note two things. If you put a magnet at the center, its repulsive force will balance against the gravitational force when the pendulum is deflected by ten degrees. Moreover, the frequency of the pendulum in the absence of magnets is about 1.25 ticks a second, i.e. it will swing from one side to the other and back 1.25 times a second. Finally, the pendulum will dampen from a swing of 5 inches to a swing of 1 inch in 30 seconds.

One final note. The support pillar for the pendulum sits in one corner of the board. Because the motion is chaotic, it is possible for the pendulum to collide with this pillar for almost any possible start. You can assume either that such collisions are elastic (i.e. the pendulum bounces off the pillar with an equal but opposite velocity) or inelastic (i.e. the pendulum loses all velocity in the direction of the pillar).

3.0 Simulation

To do a simulation you need to compute the motion of the pendulum (it's the only thing that changes) from one instance of time to the next. This can be done by computing the various forces on the pendulum and then using Newton's formulas for motion ($F = ma$ where F is force, m is mass, and a is acceleration; acceleration is the change in velocity over time; velocity is the change in position over time.)

You should keep track of the position (x and y are sufficient, since the z position is directly dependent on this), and the velocity of the pendulum (again x and y are sufficient). You break time into lots of small increments. For each

increment, you use the current velocity to update the position and the current acceleration to update the velocity. The acceleration here is computed by summing up the gravitational force, the force of each magnet on the base, and the damping force. All of these have to be computed as vectors.

To ensure that your simulation is accurate, you should use a variable time step. The time step to be used at each point in the simulation should be chosen so that the distance that the pendulum moves is less than some fixed amount. This means that the time step should be inversely proportional to the current velocity (with some attention paid to the current forces). You should choose the distance constant here small enough to keep things realistic, but not too small so that the simulation is unable to run in approximately real time. You will have to experiment to find a correct value. You should also limit the size of the time step to something reasonable. You should also experiment to ensure that the pendulum does the right thing (i.e. that its rate of damping and oscillation are approximately correct and that the equilibrium between magnetism and gravity is about what was specified).

The method of updating the position and velocity at each interval can be as simple as using the current velocity to compute the next position and the current forces to compute the next velocity. Note that such a calculation is going to be somewhat inaccurate since the average velocity over the time step is really somewhere between the original and final velocity. Thus, using this approximation will necessitate smaller time steps. You can also use a better way of doing the calculation and thus use larger time steps. (You might want to investigate Runge-Kutta methods for example.)

4.0 User Interface

You have two choices of interface on this assignment. You can either do an all textual interface or you can do a graphical interface. Both have advantages and disadvantages.

The advantage of a textual interface is that the coding should be relatively easy. You need to input the position of the magnets and the initial position of the pendulum and then output the position of the pendulum at discrete intervals of time to show what is going on. The disadvantage of this approach is that it will be very difficult for you to verify that your system is working correctly and you will spend lots of time reading and graphing the output manually to see what is going on.

The graphical interface will be a bit more difficult to program. (You should keep it relatively simple — i.e. just showing the position of the magnets and then the motion of the pendulum as a 2D display.) However, it will make it much easier for you to see if and how well your program is working and to adjust the various parameters accordingly.

In either case, your program must be able to read an initial setup from an input file. This will consist of:

```
<# of magnets>  
<x position> <y position> <up/down> for each magnet  
<x position> <y position> of the pendulum
```

Here the positions are given in inches and denote the location of the center of the corresponding magnet (or for the pendulum, the x,y position directly underneath the magnet). It is assumed that the center of the board is the position (0,0) and that the pillar is in the upper right quadrant (+x,+y). The up-down value for the magnet should be +1 if the magnet is attractive and -1 if the magnet is repulsive. Note that the initial velocity of the pendulum is assumed to be zero in this case. You can add an optional line to the input file with an initial velocity, but you will have to determine the appropriate units.

For both graphical and textual outputs, you should report the position of the pendulum in fixed time intervals (e.g. every tenth of a second).

You might also want to allow the user to play with the various constants dynamically (i.e. the strength of the magnets, the weight of the pendulum, and the damping force). Not only will this make your program more fun to play with, but it will also make it easier for you to find appropriate default values that make the simulation more realistic.

5.0 Mechanics

You should hand in your program, sample runs, and whatever output you have to demonstrate your program works. You should also include a brief write-up that describes the algorithms used for each routine. Handins should be done electronically if at all possible. They should be placed in the directory /map/auxlfred/comps/handins/### where ### is the random number you assign yourself. You should have permissions to create this directory and should protect it accordingly.

All work that you hand in is expected to be your own. You should not discuss your work or ideas with anyone else. You should not share code or ask others for advice on your code. You should not read others code (either those here or code obtained elsewhere) that duplicates in any way what you are to write. You may ask Dr. Reiss questions either in person or via email. Any answers or information that should be known to all will be posted in the comps news group which is your responsibility to read. An elementary physics text will be available (and appropriate sections will be copied) for those who want it.