

MA198/CS199 VPython Project Cover

Circle: Draft / **Final**

Attach this to your paper documentation.

- Your Name and netid: Colleen Williams 661681394
 - Date of this document: 13 December 2016
 - Project title: Water Wheels in VPython
 - Name of RTICA: FlourMill.py, Triphammer5.py, Hierapolis3.py
 - Version Number of VPython: 2.7
-

Do not write below this line, but leave on sheet.

- Printed documentation
- Hand annotated RTICA code:
- Webpage in repo <netid>/public_html/index.html:
- Project documentation in repository has
 - Table of contents.
 - VPython code. (Linking this to webpage is optional)
 - Instruction how to demo the RTICA.
 - LaTeX source .tex with figures, if any.
 - LaTeX .pdf of your documentation.
- Evaluation of Public Presentation:

Contents

1	Abstract	3
2	Histories and Uses of the Mills	3
2.1	How These Wheels Work	3
2.2	Hierapolis Mill	4
2.3	Triphammer	4
2.4	Flour Mill	5
3	The Code Behind the Wheel	5
3.1	Movement of the Water	6
3.2	Flour Mill	6
3.2.1	Placing Pegs for the Gears	6
3.3	Trip Hammer	7
3.4	Hierapolis	8
4	Sources	9

Water Wheels in VPython

Colleen Williams

December 13, 2016

1 Abstract

Water wheels have had a variety of uses over the centuries, despite their fundamental similarity: make moving water do some other sort of work by turning a wheel attached to other functional parts. The purpose of modeling a water wheel is to gain a fundamental understanding of how many modern machines work. My functions of choice were a My examples of three different mills that use water wheels- flour, Hierapolis, and triphammer- will demonstrate the forms of work that a water wheel can perform and how they are carried out. I plan also to show how gear ratios within the mills affect the revolutions per minute and torque of the output machinery at the end of a mill.

2 Histories and Uses of the Mills

Water wheels are a way to change the kinetic energy from a flowing body of water to do some specific type of work. I have three examples of structures, each of which performs a slightly different task. Specifically related to coding with VPython, each part of the wheel can be placed in a frame related to the main wheel and should be able to rotate at some related speed, either the same as the wheel or according to a ratio of gears at some position of the mill. The water wheel is similar to an engine in how it can do work; the motion of the Hierapolis mill is especially comparable to an engine and piston, which changes rotational motion to linear. An engine's energy just comes from a combustable fuel.

2.1 How These Wheels Work

The equation $Pow \left[\frac{Nm}{s} \right] = \frac{W}{s} = \omega \left[\frac{rad}{s} \right] T [Nm]$ (where Pow=power in Joules/second, W=work, s=time in units of seconds, T=torque in units of newton meters, and ω is how far the wheel has turned in $\left[\frac{radians}{second} \right]$) describes the water pushing on the wheel to make it turn. A fixed amount of work is put into the wheel in the form of moving water. Therefore, if the wheel is to make a smaller round object, such as millstones, move very quickly to exert forces onto small objects like grain, the goal is to get torque down and ω up. The millstone can be made to turn quickly because a gear that connects to the stone in a given ratio.

A ratio between two gears can determine how many revolutions one gear makes compared to another, allowing the stone to turn many times when the water wheel turns only once, depending on what the ratio is. The trip hammer, as it only hammers once per rotation of the water wheel, has a small angular velocity but high torque; these hammers in real life are heavy and a lot of torque is needed to swing the hammer. This is comparable to riding a bicycle; it is easy to pedal on a low gear but takes many pedals to travel a short distance, and difficult to pedal on a high gear but fewer pedals for the same distance (1).

2.2 Hierapolis Mill

The Hierapolis water mill is a specific mill used by the Romans in what is now Turkey. Fig. 1 shows a small part of the mill, which was used to cut stone and use it as building material (4). It was the most challenging mill to build because the saw must move back-and-forth linearly in a sinusoidal pattern, and a linkage connects the saw to the main wheel structure.

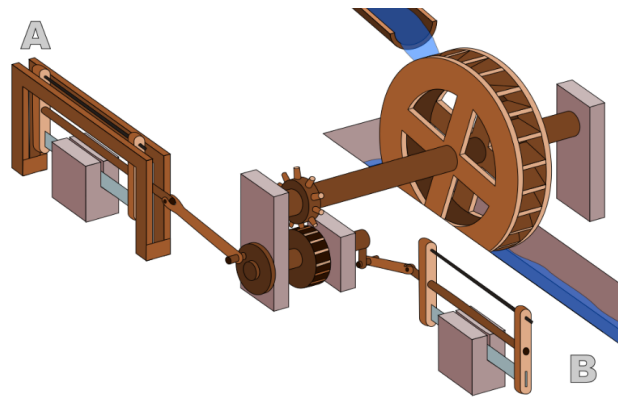


Figure 1: The Hierapolis mill and others like it created linear motion from the flow of water. The mills would have contributed

2.3 Triphammer

The triphammer, as seen in Fig. 2 is a tool used by blacksmiths to pound onto hot metal, thus changing the metals' shape (2). It makes sense to use a water wheel to do this work because, if the flow of water is steady, the wheel gives a predictable, cyclic pattern of motion for the hammer, which is demonstrated in my animation.

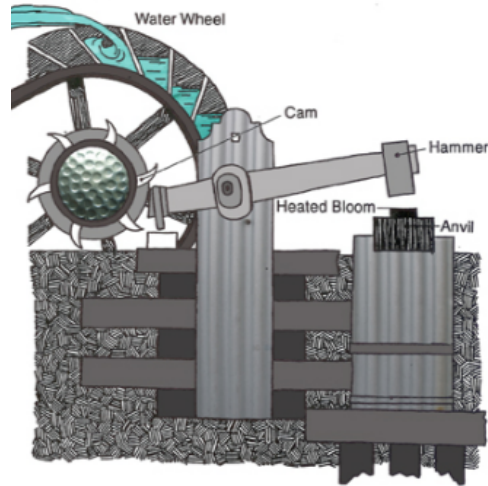


Figure 2: This trip hammer has several cams pushing it up so it can be used for pounding metal.

2.4 Flour Mill

Fig. 1 shows a flour mill used for grinding grain into flour. Grain is poured in between two millstones, which spin and grind it(6). A larger crown turns the smaller pinion, which completes more rotations than the crown in a given amount of time and makes the millstone turn very quickly. This mill consists of many turning parts and not much linear motion.

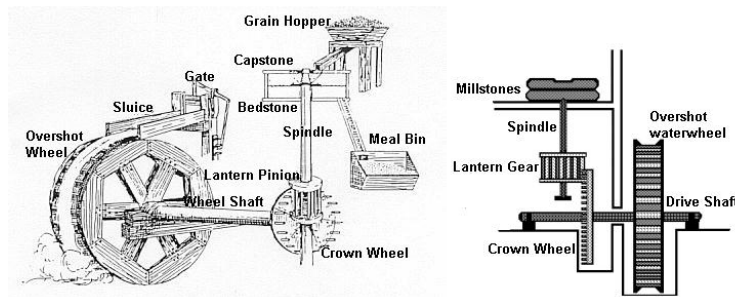


Figure 3: A flour mill with crown and pinion gears shown, as well as where grain is poured in between two grinding stones.

3 The Code Behind the Wheel

Each wheel was created in a separate file and runs with its own code. It should be noted that several coding ideas are used in all three of the wheels:

- Reference frames are points in the world of VPython that is the reference point for the position and movement of objects that are part of that frame. The wheel is used as a first reference frame and named "fwheel" in all three programs. Other independent

frames are relative to the original world frame. For example, someone can define a frame by typing "fframe=frame(frame=world, pos=(x,y,z)), "pos" meaning "position."

- angular velocities are labelled "rot" to make sense of relations among moving parts, for example, the linear velocity of water and angular velocity of the wheel.
- While loops allow code to repeat over again a given number of times, which is perfect for the cyclic motion of a wheel.
- A label on the flour mill and trip hammer RTICAs indicate to click to start the animation. The Hierapolis starts automatically.

3.1 Movement of the Water

The wheel needs something to turn it, in this case water. It was discovered that water flowing from above the wheel is the most effective way to turn it (cite), so it made sense to have a drop of water fall onto the wheel. I have it set above the wheel at a position pos=(x,y,z). With each frame, the water drop has to move down a certain amount and push the wheel, preferably not overlapping with the wheel. The velocity of an object is described as $v = dx/dt$ (v is velocity, x is position, t is time), or $v = \frac{(x_2 - x_1)}{\Delta t}$. Therefore, $x_2 - x_1 = v\Delta t$

In VPython, this is expressed as

```
water.pos = water.pos + water.velocity*deltat, with  
water.velocity = vector(0,-2*25*rot,0)
```

This formula was taken from the VPython website, which offered the most information on how to set objects in motion, along with most other important parts of this project(1). Ideally, the water should be accelerating down instead of moving with a constant velocity. Ideally, a single water molecule also does not fill a whole water wheel paddle and drive the motion of large wooden structures. With this inaccuracy in mind, having zero acceleration does not take away from the understandability of the model.

3.2 Flour Mill

The Flour Mill features two useful frames, fwheel and fpinion, which rotates about its y-axis. I have the radius of the crown equal to a length r and the radius of the pinion to a length $\frac{r}{4}$, allowing a gear ratio of 1:4 and an angular velocity of the same ratio. By the equation, $Pow = \omega T$, torque exerted by the pinion is also four times less than if the gear ratio were 1:1.

3.2.1 Placing Pegs for the Gears

The crown contains pegs that turns with the water wheel and pushes the pinion. These pegs are created and placed with code, such as the following:

```
peg=cylinder(frame=fcrown,pos=(10,8*cos(math.pi/8),8*sin(math.pi/8)),axis=(6,0,0),radius=.5)
```

The x position of each peg is in relation to the frame of the crown (10 units farther in the x -axis). The formula $x = r\cos(\theta)$ assigns y and z values to place the pegs onto the circular crown. A total of sixteen pegs are placed at angles of multiples of π , $\frac{\pi}{8}$, $\frac{\pi}{4}$, and $\frac{\pi}{2}$, as seen in Fig 4.



Figure 4: The cyan pegs on the crown and pinion are a visual aid to see that the pinion rotates four times for every single time the crown rotates.

3.3 Trip Hammer

The trip hammer consists of the same wheel, shaft, and crown structure as part of fwheel, but now a cam takes the place of pegs to move a part, in this case a hammer. The cam in real life should be smooth so as not to bump violently into the hammer each time it turns around. The VPython Help page includes a list of shapes I used to figure out how to make the cam. The letter "A" ended up being most useful because it could be set at an angle using "axis=(a,b,c)," although this command seems to have limitations for reasons I do not yet understand. I ended up needing to rotate fwheel before animation to set the cam next to the hammer.

Building the trip hammer was not as simple as the flour mill because it does not include completely uniform motion. Although fwheel rotates consistently, the hammer does not because the cam pushes the hammer for some frames, but not all. The hammer also must change direction after the cam passes it. I first attempted to rotate fwheel one frame at a time and subsequently rotated the hammer, ran the program, and zoomed into where the hammer and cam might intersect, readjusting the hammer's angle as needed. I thought I might be able to create a formula to describe the motion of the hammer using a scatter plot made on Excel (Fig. 3.3) and make the hammer's code more concise; this plot did not show a kind of function familiar to me, however, so I kept the individual frames to make the hammer rotate.

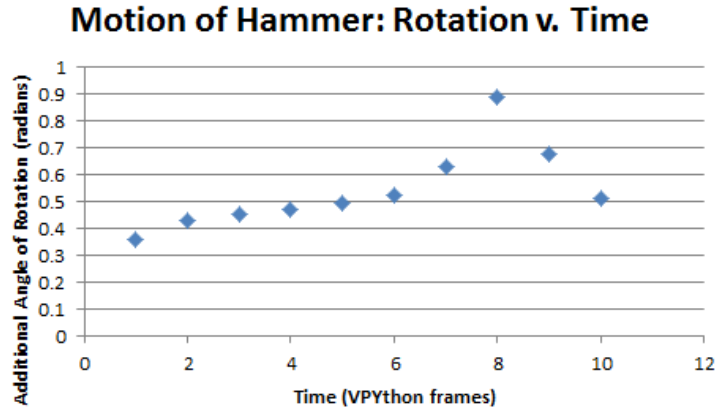


Figure 5: Each point represents how far the hammer must rotate after a given frame to appear to be pushed by the rotating cam.

In order to fully describe the motion of the wheel’s rotation as well as the hammer’s and the water drop, a nested while loop is useful. First, a variable "var" is created and set equal to 0. Then the loop begins:

```
while (var<12) :
    rate(100)
    water.pos = water.pos + water.velocity*deltat
    t = t + deltat
    var=var+1
    fwheel.rotate(axis=(20,0,0), angle=rot)
    fhammer.rotate(axis=(1,0,0), angle=-.36*rot)
    sleep(.05)#1
```

A total of twelve frames are a part of this section of the while loop, each frame adding a value of 1 to the variable "var." An angle "rot" is defined earlier in this wheel’s code as $\frac{\pi}{40}$ radians; because one full rotation contains 2π radians, each full rotation contains $2\pi \text{ radians} * \frac{40 \text{ frames}}{\pi \text{ radians}} = 80$ frames. Why, then, does the first while loop section have twelve frames? This is how many frames it takes to rotate the hammer to its farthest extent before the cam passes the cam and lets the hammer fall again. The next part of the while loop applies to the hammer falling back down to its original position, using "while (var<22)." The final sections of the loop allow the wheel to continue to rotate without the hammer, reset the water’s position so it can represent a constant flow onto the wheel, and reset the variable equal to 0 so the while loop can start over at the "while(var<12)" section. The wheel then keeps on turning.

3.4 Hierapolis

the Hierapolis mill is the most challenging mill because it involves the rotation and translation of a stick connecting the saw and wheel in ways I do not yet know. Rotating the angle of the

wheel in increments and finding the new position of each part was more difficult than doing so for the trip hammer. Eventually, I found code for simple harmonic motion of a ball and applied it to the saw of the mill. Describing the motion of the linkage between the wheel and saw, however, remains a challenge I hope to overcome in the future.

4 Sources

(1) Bluejay, Michael. (n.d.). How to Use Bicycle Gears. Retrieved from <http://bicycleuniverse.info/eqp/geom>

(2) Howell, Courtney. (2011). *Diagram of a forge trip hammer*. [digital image]. Retrieved from <http://explorepahistory.com/displayimage.php?imgId=1-2-FB9>

(3) No title. [digital image]. Retrieved from https://en.wikipedia.org/wiki/Hierapolis_sawmill (underscore before "sawmill" instead of space")

(4) Römisch-Germanisches Zentralmuseum Mainz. (2015, September 30). Retrieved from <http://www3.rgzm.de/ephesos/>

(5) VPython. Retrieved from <http://www.vpython.org/contents/docs/index.html>

(6) Zbornik, Barry. (n.d.). Lima Iowa: Old Mill site and mill information. Retrieved from <http://www.angelfire.com/ia/z/limamill.htm>