Katherine Aho
NSF GK-12 Vibes and Waves in Action
Honors Physics
Lesson 16: Spring Constants of bungee cords

Summary of Lesson

Imagine you are the owner of a bungee jumping business. You have three different bungees available with three different spring constants. Your job is to determine which bungee to give to a customer depending on their weight, that will be give them the most thrilling ride, yet still be safe. The students were asked to write an R program on this scenario where the user inputs their weight. This program incorporates their physics topic of spring motion. They were allowed to use a previous program where they solved a quadratic equation as a basis for this program.

Honors Physics Lesson Plan

Text: Conceptual Physics, Paul G. Hewitt
Chapter: Ch 25- Vibrations and Waves
Objectives: Write a program in R to determine the best bungee cord to use
Essential Question: How does the spring constant affect the stretch of the spring?
Frameworks: Waves- 4.1; SIS1, SIS2, SIS3, SIS4

L-Side Activities: Teacher	R-Side Notes: Students
At the Bell: Brainstorm about the logic to determine the correct bungee **Agenda:** 1. Explain the objective of the program 2. Write the program 3. Answer the follow up questions **Working It Out:** 1. How does the spring constant affect the stretch of the spring? **Class Activity:** You are the design engineer for a bungee jumping business. It is your job to make sure every client gets the most exciting, yet safest, ride. You can supply three different bungee cords, each one has its own stiffness (or spring constant) but they all have a free length (L) of 10 m (natural length when no force is applied to stretch it). The clients will jump from a structure over the water from height of 30 m. For the most thrilling experience, you must provide the bungee cord that allows the client to fall as far as 3 meters above the water, but no closer (for safety reasons). **Homework:** None	The distance the bungees will stretch (s) is given by the following quadratic equation: $\frac{1}{2} ks^2 - mgs - mgL = 0$

R Coding Activity 7

Objective
Using R, determine which bungee cord can be safely used by a client (given their weight).

Background
You are the design engineer for a bungee jumping business. It is your job to make sure every client gets the most exciting, yet safest, ride. You can supply three different bungee cords, each one has its own stiffness (or spring constant) but they all have a free length (L) of 10 m (natural length when no force is applied to stretch it). The spring constants and corresponding colors are as follows:

Stiffness	**Color**	**Spring Constant (k) in N/m**
Soft	Green	160
Medium	Orange	200
Stiff	Red	240

The clients will jump from a structure over the water from height of 30 m. For the most thrilling experience, you must provide the bungee cord that allows the client to fall as far as 3 meters above the water, but no closer (for safety reasons).

The distance the bungees will stretch (s) is given by the following quadratic equation:

$$\frac{1}{2} ks^2 - mgs - mgL = 0$$

Requirements
- Use a minimum of five helpful comments.
- Use helpful and appropriate variable names (or comment about them).
- The spring constants are constants within your program.
- Set $g = 9.8\ m/s^2$
- The program must prompt the user for their weight (in pounds) and convert it to Newtons for use in the equation.
- The output should provide information about the results for each bungee including how far it will stretch and how close to water the client will come.
- Round all numerical output to one decimal place.
- EXTRA CREDIT: The program should make a final statement about which bungee (color) should be used and its final height above the water.

```
# SOLUTION TO BUNGEE JUMPING ASSIGMENT
# FOR HONORS PHYSICS
# WRITTEN BY: KATHERINE AHO
# NSF GK-12 FELLOWSHIP
##########################################



# set constants for bungees
green = 160  # N/m
orange = 200  # N/m
red = 240   # N/m

# unstretched bungee length
L = 10  # m

#stretched bungee length
Ls = 30  #m

# prompt user to enter weight in lbs
lbs<-readline(prompt="Enter your weight in pounds: ")
lbs<-as.numeric(lbs)

# convert weight to newtons
mg = lbs*4.45

####################################
# CASE 1: GREEN BUNGEE
# setting up coefficients for quadratic equation
k1 = green
A1 = 0.5*k1
B1 = -mg
C1 = -mg*L
D1 = (B1^2)-(4*A1*C1)   # determinant

# solving the quadratic equation
x1_green = (-B1 + sqrt(D1))/(2*A1)
x2_green = (-B1 - sqrt(D1))/(2*A1)

# determining which solution to use
# can not use negative solution
# will work if weight is always positive - need more options for "negative weight"
if(x1_green>=0){
  s_green = x1_green
} else {
  s_green = x2_green
}

# calculating height above water
```

```
height_green = Ls - (L+s_green)
cat("The green bungee will stretch",s_green,"meters and will put the rider",
    height_green,"meters above the water.","\n")

# test to see if bungee is safe- puts user at least 3m above water
if(height_green<3){
  cat("The green bungee is not safe. Use a stiffer bungee.","\n")
  height_green = 1e-22  # set to a low number so it can not be used when determining which bungee to
use
}else{
  #cat("\n","The green bungee will put the rider",height_green,"meters above the water.")
}
#######################################
# CASE 2: ORANGE BUNGEE
# setting up coefficients for quadratic equation
k1 = orange
A1 = 0.5*k1
B1 = -mg
C1 = -mg*L
D1 = (B1^2)-(4*A1*C1)   # determinant

# solving the quadratic equation
x1_orange = (-B1 + sqrt(D1))/(2*A1)
x2_orange = (-B1 - sqrt(D1))/(2*A1)

# determining which solution to use
# can not use negative solution
# will work if weight is always positive - need more options for "negative weight"
if(x1_orange>=0){
  s_orange = x1_orange
} else {
  s_orange = x2_orange
}

# calculating height above water
height_orange = Ls - (L+s_orange)
cat("The orange bungee will stretch",s_orange,"meters and will put the rider",
    height_orange,"meters above the water.","\n")

# test to see if bungee is safe- puts user at least 3m above water
if(height_orange<3){
  cat("\n","The orange bungee is not safe. Use a stiffer bungee.")
  height_organe = 1e-21
}else{
  #cat("\n","The orange bungee will put the rider",height_orange,"meters above the water.")
}

#####################################
# CASE 3: RED BUNGEE
```

```
# setting up coefficients for quadratic equation
k1 = red
A1 = 0.5*k1
B1 = -mg
C1 = -mg*L
D1 = (B1^2)-(4*A1*C1)   # determinant

# solving the quadratic equation
x1_red = (-B1 + sqrt(D1))/(2*A1)
x2_red = (-B1 - sqrt(D1))/(2*A1)

# determining which solution to use
# can not use negative solution
# will work if weight is always positive - need more options for "negative weight"
if(x1_red>=0){
  s_red = x1_red
} else {
  s_red = x2_red
}

# calculating height above water
height_red = Ls - (L+s_red)
cat("The red bungee will stretch",s_red,"meters and will put the rider",
    height_red,"meters above the water.","\n")

# test to see if bungee is safe- puts user at least 3m above water
if(height_red<3){
  cat("\n","The red bungee is not safe. Use a stiffer bungee.")
  height_red = 1e-20
}else{
  #cat("\n","The red bungee will put the rider",height_red,"meters above the water.")
}
```