Projectile Motion

Carmen Wright

October 29, 2017

3.3 Homework problem: When throwing an object, the distance achieved depends on its initial velocity, v_0, and the angle above the horizontal at which the object is thrown, θ. The distance, d, in feet, that describes the range covered is given by the equation below, where v_0 is measured in feet per second.

$$d = \frac{(v_0)^2}{16} \sin\theta \cos\theta$$

Use an identity to express the formula so that it contains the sine function only. Recall the double-angle formula: $\sin 2\theta = 2 \sin\theta \cos\theta$. Now divide both sides by two and substitute:

$$d = \frac{(v_0)^2}{16} \sin\theta \cos\theta = \frac{(v_0)^2}{16} \frac{\sin 2\theta}{2} = \frac{(v_0)^2}{32} \sin 2\theta$$

Why would this ever be useful?

1. It makes it easier to determine what angles give us certain distances.

In other words, given a certain distance d, we could find an angle θ that gives us that distance. Solving for θ is easier to do in Formula 2 versus Formula 1.

First, what is the maximum distance? Since the range of the sine function is [-1,1], we know that $|\sin 2\theta| = 1$. That means that the maximum distance D_{max} is

$$D_{max} = \left| \frac{(v_0)^2}{32} \sin 2\theta \right| = \left| \frac{(v_0)^2}{32} \right| |\sin 2\theta| = \left| \frac{(v_0)^2}{32} \right| = \frac{(v_0)^2}{32}$$

What is the smallest angle, θ that produces a fraction, a, of the maximum distance, d, for a given initial speed, v_0? Let d^* be the desired distance, $d^* = aD_{max}$. In other words, if we wanted half of the maximum distance, $d^* = \frac{1}{2}D_{max}$. We do not even need to know the maximum distance in order to find the smallest angle. We just set our formula equal to our desired distance and solve for θ.

$$\frac{(v_0)^2}{32} \sin 2\theta = d^*$$

$$\frac{(v_0)^2}{32} \sin 2\theta = aD_{max}$$

$$\frac{(v_0)^2}{32} \sin 2\theta = a\frac{(v_0)^2}{32}$$

$$\sin 2\theta = a$$

$$2\theta = \arcsin(a)$$

$$\theta = \frac{\arcsin(a)}{2}$$

Remember that a is the fraction of the distance.

Let's say we want to know the smallest angle that gives us half of the maximum distance. Then

$$\theta = \frac{\arcsin(1/2)}{2}$$

```r
a <- 1/2
theta <- asin(a) / 2
theta
```

```
## [1] 0.2617994
```

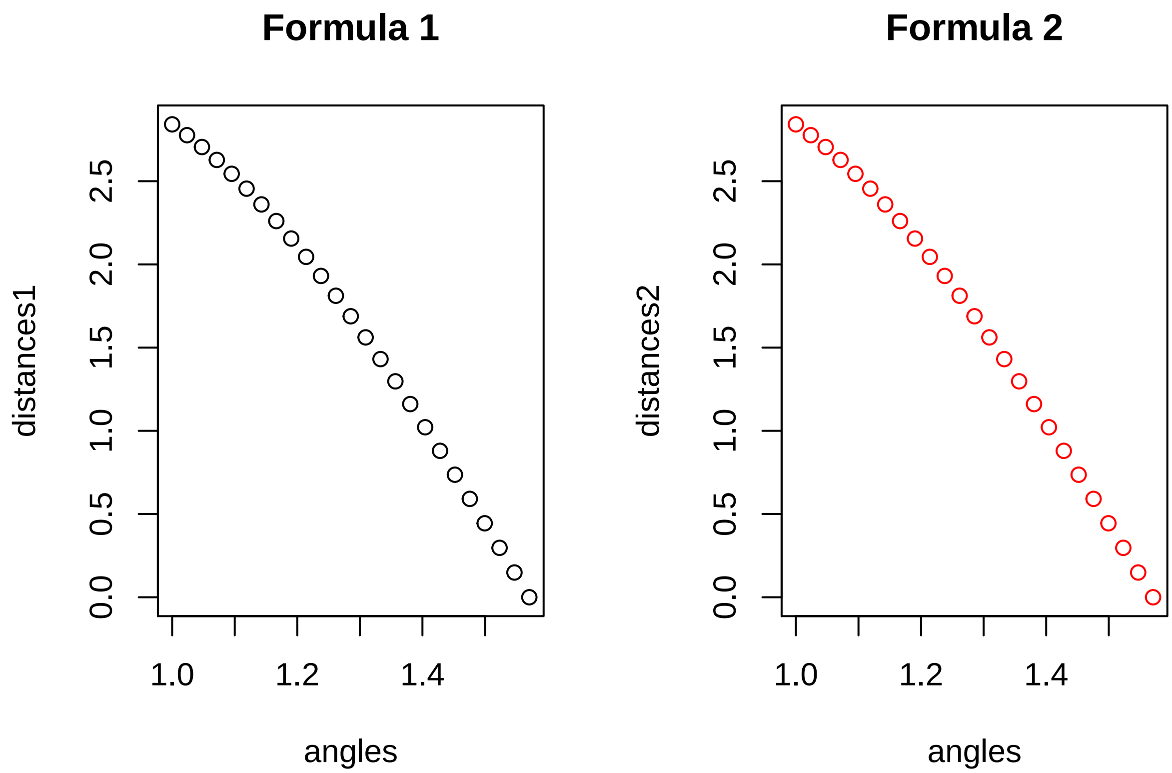
2. It could speed up computations.

Let's define the two formulas as functions that receive the initial velocity and angle as inputs:

```r
projectile1 <- function(v0, theta)
{
  d <- (v0^2 / 16) * sin(theta) * cos(theta)
  return(d)
}

projectile2 <- function(v0, theta)
{
  d <- (v0^2 / 32) * sin(2*theta)
  return(d)
}
```

Suppose our initial velocity is 10 ft/sec, and we want to calculate the ranges for 25 acute angles between 1 and $\frac{\pi}{2} \approx 1.57$. Notice that from the plots that both formulas have the same output.

```r
init_vel <- 10
angles <- seq(1,pi/2,length.out = 25)

distances1 <- projectile1(init_vel, angles)
distances2 <- projectile2(init_vel, angles)

par(mfrow=c(1,2))
plot(angles,distances1,col=1,main="Formula 1")
plot(angles,distances2,col=2,main="Formula 2")
```



That was no problem. And it doesn't seem to matter which formula we use. But we only ran it for 25 different inputs. What if we had much more?

```
nvalues <- 100
velocities <- 1:nvalues
angles <- seq(1,pi/2,length.out = nvalues)
inputs <- expand.grid(velocities,angles)
N <- nrow(inputs)
print(paste("Number of calculations:",N))
```

```
## [1] "Number of calculations: 10000"
```

```
dist1 <- numeric(N)
dist2 <- numeric(N)
```

Let's time the first formula:

```
library(tictoc)

tic("Formula 1")
for (i in 1:N)
{
  dist1[i] <- projectile1(inputs[i,1],inputs[i,2])
}
toc()
```

```
## Formula 1: 0.27 sec elapsed
```

Now the second formula:

```r
tic("Formula 2")
for (i in 1:N)
{
  dist2[i] <- projectile2(inputs[i,1],inputs[i,2])
}
toc()
```

```
## Formula 2: 0.29 sec elapsed
```

10000 may be a lot of calculations, but not enough to make a difference. Let's try it with more calculations.

```r
nvalues <- 500
velocities <- 1:nvalues
angles <- seq(1,pi/2,length.out = nvalues)
inputs <- expand.grid(velocities,angles)
N <- nrow(inputs)
print(paste("Number of calculations:",N))
```

```
## [1] "Number of calculations: 250000"
```

```r
dist1 <- numeric(N)
dist2 <- numeric(N)
```

Let's time the first formula:

```r
library(tictoc)

tic("Formula 1")
for (i in 1:N)
{
  dist1[i] <- projectile1(inputs[i,1],inputs[i,2])
}
toc()
```

```
## Formula 1: 6 sec elapsed
```

Now the second formula:

```r
tic("Formula 2")
for (i in 1:N)
{
  dist2[i] <- projectile2(inputs[i,1],inputs[i,2])
}
toc()
```

```
## Formula 2: 5.88 sec elapsed
```