Ballistic Trajectory with Drag

Ken Harmon

2020 May 28

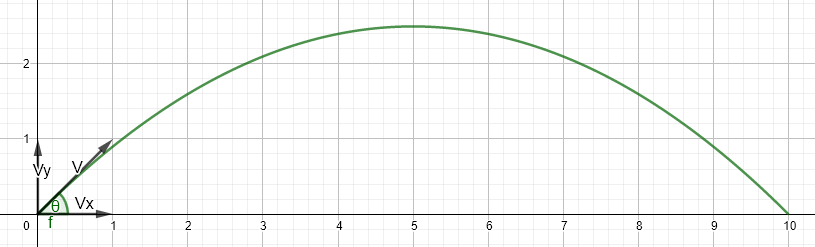
# 

Ballistic Trajectory with Drag

*No Drag*

Firing Artillery is an interesting proposition. Hitting the target with as few rounds as possible is very important. Each round fired comes at a cost. Besides the monetary cost, each round draws unwanted attention from the enemy. Therefore, each round needs to be carefully considered. Each target needs to provide enough payoff to outweigh the cost and the calculations for each round needs to be meticulous and timely in order to have as little adjustment as possible and requiring subsequent rounds.

The initial physics behind firing artillery seems pretty straight forward. Initial velocity at some angle: the initial velocity vector V



We analyze the components of that initial vector separately Vx and Vy.

If there were no drag on the round, Vx would remain constant and Vy would be affected only by gravity. For this paper,

So, for a given initial velocity, in order to hit the target, we need x to equal the range (r) to the target at the same time y equals the height of the target (zero in this case).

So, for a target at a given range and a round fired at a set velocity, we know the angle that must be shot.

We also know the maximum height (H) of the round happens at half the range and half the time.

, then

For example, to hit a target at 15km with an initial velocity of 671 m/s, the initial angle needs to be .1664 radians, 9.54 degrees, or 169.7 mils. Time of flight is 22.668. Max height is 629.87 m at 11.3339 seconds. Impact angle is -169.7 mils. Impact velocity is 671 m/s.

This is all very predictable, but there is a great deal of drag so things change.

*Drag*

If there is drag on the round, there are many variables that affect the velocity of the round. Velocity in the x direction is no longer constant. I get these formulas from Peter Chudinov from the journal of Physics[[1]](#footnote-22).

The drag constant

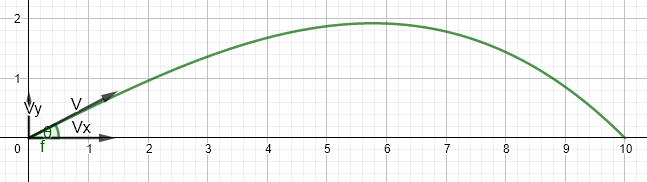
;

ρ*a* is the air density, *cd* is the drag factor for a sphere, *S* is the cross-section area of the object, and *Vt* is the terminal velocity.

I will treat k as a constant even though ρ changes with altitude which may be significant with artillery shots; in this case k could become a function. All variables are functions of θ. Theta is the angle of trajectory that varies throughout the flight.

Launch angle:

Impact angle:



Here are the functions I am going to use to predict artillery shots given and initial velocity and launch angle.

These first two equations I can compute directly from θ.

The following formulas I cannot compute directly but I can get a numerical solution for each θ.

The velocity in the x and y directions is now based off of instantaneous velocity and trajectory angle; not just initial velocity and launch angle.

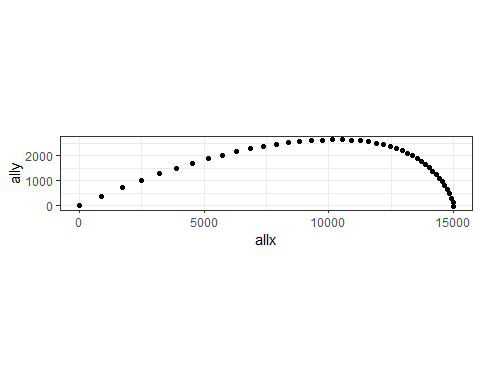
Now convert all these so they are in terms of θ:

Once I have and instantaneous altitude, I can alter k for appropriate air pressure.

*Attempt at prediction*

Actual shot data that I am trying to match: Initial velocity 682 m/s at an initial angle of 0.4174391238457 radians (425.2 mils). My prediction needs to hit a level target at 15000 m in 42.1 seconds while reaching a maximum altitude of 2295 m. Also, impact angle of - 0.6695519342963 radians (-682 mils) with and impact velocity of 310 m/s.

V0 <- 682 # initial velocity in m/s for M795 with M232A1 4H  
am0 <- 425.2 # QE in mils for a level 15000 m shot  
th0 <- am0 \* pi / 3200 # initial angle in radians  
x0 <- 0 #Initial x  
y0 <- 0 # initial y  
t0 <- 0 # initial time  
g <- 9.80665 # gravitational force in m/s/s  
# All functions are in terms of angle of the trajectory  
press <- data.frame(cbind(  
 alt = c(0,200,500,1000,1500,2000,2500,3000,3500,  
 4000,4500,5000,6000,7000,8000,9000),  
 rho = c(1.2250,1.2133,1.1844,1.1392,1.0846,1.0320,  
 .9569,.8632,.7768,.6971,.5895,.4664,.3612,.2655,.1937,.1413)))  
c.press <- glm(rho~poly(alt,6,raw=TRUE), data = press)  
rho <- as.numeric(predict(c.press, data.frame(alt = y0), type = "response"))  
pk <- .00000156  
k <- rho\*pk # is the drag constant at 0 alt  
  
#Lets build a table along the trajectory  
ths <- seq(th0, -1, by = -.02)  
  
f0 = sin(th0)/(cos(th0))^2 + log(tan(th0/2+pi/4))  
allth <- th0  
allf <- f0  
allk <- k  
allV <- V0  
allt <- t0  
allx <- x0  
ally <- y0  
allrho <- rho  
  
  
for (th in ths) {  
 allth <- c(allth,th)  
 allk <- c(allk,k)  
 f = sin(th)/(cos(th))^2 + log(tan(th/2+pi/4))  
 allf <- c(allf,f)  
 V = V0\*cos(th0)/(cos(th)\*sqrt(1+k\*(V0\*cos(th0))^2\*(f0-f)))  
 allV <- c(allV,V)  
 tint <- integrate(function(j) {V0\*cos(th0)/((cos(j))^2\*sqrt(1+k\*(V0\*cos(th0))^2\*(f0-f)))},th0,th)  
 t <- t0 - 1/g\*as.numeric(tint[1])  
 allt <- c(allt,t)  
 xint <- integrate(function(j) {(V0\*cos(th0)/(cos(j)\*sqrt(1+k\*(V0\*cos(th0))^2\*(f0-f))))^2},th0,th)  
 x <- x0 - 1/g\*as.numeric(xint[1])  
 allx <- c(allx,x)  
 yint <- integrate(function(j) {tan(j)\*(V0\*cos(th0)/(cos(j)\*sqrt(1+k\*(V0\*cos(th0))^2\*(f0-f))))^2},th0,th)  
 y <- y0 - 1/g\*as.numeric(yint[1])  
 ally <- c(ally,y)  
 rho <- as.numeric(predict(c.press, data.frame(alt = y), type = "response"))  
 allrho <- c(allrho,rho)  
 k <- pk \* rho  
}  
traj <- data.frame(cbind(allth,allk,allf,allV,allt,allx,ally,allrho))  
trajp <- traj[2:nrow(traj),]%>%filter(ally>=-200)  
trajp %>% ggplot(aes(allx,ally))+geom\_point()+  
 coord\_fixed(ratio = 1)



colnames(trajp) <- c("theta","k","f","V m/s","t s","x m","y m","pressure")  
pander(trajp)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| theta | k | f | V m/s | t s | x m | y m | pressure |
| 0.4174 | 1.919e-06 | 0.9153 | 682 | 0 | 0 | 0 | 1.23 |
| 0.3974 | 1.919e-06 | 0.8636 | 663.5 | 1.48 | 905.5 | 390.8 | 1.19 |
| 0.3774 | 1.857e-06 | 0.8132 | 647.2 | 2.887 | 1737 | 729.5 | 1.161 |
| 0.3574 | 1.812e-06 | 0.764 | 632.6 | 4.232 | 2508 | 1025 | 1.137 |
| 0.3374 | 1.773e-06 | 0.7159 | 619.5 | 5.523 | 3228 | 1282 | 1.113 |
| 0.3174 | 1.736e-06 | 0.6687 | 607.6 | 6.767 | 3906 | 1508 | 1.09 |
| 0.2974 | 1.701e-06 | 0.6225 | 596.9 | 7.971 | 4549 | 1706 | 1.068 |
| 0.2774 | 1.666e-06 | 0.5772 | 587.1 | 9.139 | 5161 | 1879 | 1.047 |
| 0.2574 | 1.633e-06 | 0.5326 | 578.3 | 10.28 | 5747 | 2031 | 1.027 |
| 0.2374 | 1.601e-06 | 0.4887 | 570.2 | 11.39 | 6311 | 2163 | 1.008 |
| 0.2174 | 1.573e-06 | 0.4454 | 562.8 | 12.47 | 6853 | 2277 | 0.9914 |
| 0.1974 | 1.547e-06 | 0.4027 | 555.9 | 13.53 | 7377 | 2374 | 0.9766 |
| 0.1774 | 1.523e-06 | 0.3606 | 549.5 | 14.57 | 7882 | 2454 | 0.9638 |
| 0.1574 | 1.504e-06 | 0.3188 | 543.6 | 15.59 | 8369 | 2520 | 0.9532 |
| 0.1374 | 1.487e-06 | 0.2775 | 538 | 16.58 | 8838 | 2571 | 0.9449 |
| 0.1174 | 1.474e-06 | 0.2365 | 532.7 | 17.56 | 9289 | 2608 | 0.9387 |
| 0.09744 | 1.464e-06 | 0.1958 | 527.6 | 18.51 | 9722 | 2631 | 0.9349 |
| 0.07744 | 1.458e-06 | 0.1553 | 522.8 | 19.45 | 10136 | 2641 | 0.9332 |
| 0.05744 | 1.456e-06 | 0.1151 | 518.1 | 20.36 | 10531 | 2638 | 0.9337 |
| 0.03744 | 1.457e-06 | 0.07493 | 513.6 | 21.25 | 10905 | 2622 | 0.9363 |
| 0.01744 | 1.461e-06 | 0.03488 | 509.1 | 22.12 | 11259 | 2595 | 0.9409 |
| -0.002561 | 1.468e-06 | -0.005122 | 504.8 | 22.96 | 11592 | 2556 | 0.9474 |
| -0.02256 | 1.478e-06 | -0.04513 | 500.6 | 23.79 | 11904 | 2506 | 0.9556 |
| -0.04256 | 1.491e-06 | -0.0852 | 496.5 | 24.59 | 12196 | 2445 | 0.9654 |
| -0.06256 | 1.506e-06 | -0.1254 | 492.4 | 25.37 | 12467 | 2374 | 0.9765 |
| -0.08256 | 1.523e-06 | -0.1657 | 488.5 | 26.12 | 12718 | 2294 | 0.9888 |
| -0.1026 | 1.542e-06 | -0.2062 | 484.6 | 26.86 | 12950 | 2205 | 1.002 |
| -0.1226 | 1.563e-06 | -0.247 | 480.9 | 27.58 | 13164 | 2108 | 1.016 |
| -0.1426 | 1.585e-06 | -0.2881 | 477.3 | 28.28 | 13362 | 2004 | 1.03 |
| -0.1626 | 1.607e-06 | -0.3295 | 473.9 | 28.97 | 13546 | 1893 | 1.045 |
| -0.1826 | 1.63e-06 | -0.3713 | 470.6 | 29.64 | 13715 | 1775 | 1.06 |
| -0.2026 | 1.653e-06 | -0.4136 | 467.5 | 30.3 | 13873 | 1652 | 1.074 |
| -0.2226 | 1.676e-06 | -0.4565 | 464.5 | 30.95 | 14021 | 1523 | 1.089 |
| -0.2426 | 1.698e-06 | -0.4999 | 461.8 | 31.58 | 14159 | 1388 | 1.103 |
| -0.2626 | 1.72e-06 | -0.5439 | 459.3 | 32.22 | 14290 | 1248 | 1.116 |
| -0.2826 | 1.741e-06 | -0.5887 | 457 | 32.84 | 14413 | 1104 | 1.13 |
| -0.3026 | 1.762e-06 | -0.6343 | 454.9 | 33.46 | 14530 | 954.3 | 1.143 |
| -0.3226 | 1.782e-06 | -0.6807 | 453 | 34.07 | 14639 | 799.8 | 1.156 |
| -0.3426 | 1.803e-06 | -0.7281 | 451.3 | 34.68 | 14740 | 640.3 | 1.169 |
| -0.3626 | 1.823e-06 | -0.7765 | 449.7 | 35.28 | 14831 | 475.9 | 1.183 |
| -0.3826 | 1.845e-06 | -0.826 | 448.1 | 35.86 | 14909 | 306.5 | 1.198 |
| -0.4026 | 1.869e-06 | -0.8767 | 446.6 | 36.43 | 14970 | 132.4 | 1.215 |
| -0.4226 | 1.896e-06 | -0.9287 | 445.1 | 36.97 | 15007 | -46.09 | 1.236 |

1. <https://iopscience.iop.org/article/10.1088/1742-6596/1287/1/012032> [↑](#footnote-ref-22)