

How can you predict a projectile’s path?  
When you launch a ping pong ball, its **trajectory** (path) moves in two directions –  
   
 **vertical** (how *high* the ball goes)   
 **horizontal** (how *far* the ball goes)

**►**

This shows the **velocity** (speed) of the **ball** (v) at three points   
in its **trajectory**, separated into **x** and **y** directions (vx andvy.)

**►**







The highest point of the ball’s trajectory is the **vertical distance** (dy).





**Measure The Time**

*Use a* ***stop watch*** *or photogate to time how long the ball is in the air.*



**Calculate Y-Axis Velocity**

**vf** = average (Y-Axis) velocity

**t** = time

**d** = distance (height)

**vi** = initial velocity



If you have the other variables,  
calculating the velocity is easy!

**Kinematic Equation**



How far the bar is launched is the **horizontal distance** (dy).





**vf** = average (X-Axis) velocity (speed)

**t** = time

**d** = distance

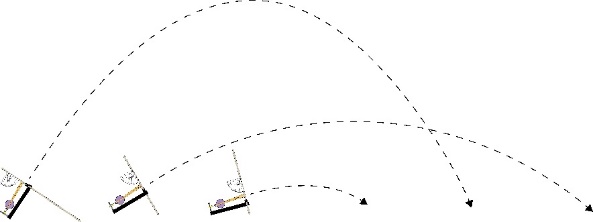


**Calculate X-Axis Velocity**

**Kinematic Equation**









*Take your best shot!*

**Adjust** your launcher’s **angle** size three times,  
taking three shots for each angle.



**Small**

**Medium**

**Large**

< 30˚

30˚-60˚

> 60˚

|  |  |  |
| --- | --- | --- |
| **Variables** | **Y-Axis** | **X-Axis** |
| **Distance** | dy | dx |
| **Velocity** | vy | vx |
| **Acceleration** | 9.8 m/s2 | 0 m/s2 |
| **Time** | t | t |

**Collect** data for the **y-axis** and **x-axis**. **Graph** the distance each shot landed -  
*a new graph for each launch angle.*

A **photogate**   
provides very accurate time measurements.





**Angle˚**

**X-Axis** **=** **Distance in Meters**

1 2 3 4 5 6 7 8 9 10



**Curve of Best Fit!**

**Misfire** (Outlier)

**Draw** a **curve** or **line of best fit**that follows (fits) your data’s path.

|  |  |  |
| --- | --- | --- |
| **Sample Data** | | |
| **25˚** | 2 m | 2 m |
| **50˚** | 4 m | 5 m |
| **70˚** | 8 m | 2 m |

Some shots misfire.   
These **outliers** are   
too far from the line   
of best fit to include.



A **stopwatch** works as well,   
(less accurately.)







Using the graph, **predict** which   
angle will hit a **target distance**.   
  
**Launch** with the predicted angle.  
Repeat with more predictions –   
**refine** the line/curve of best fit!  
  
**Use these distances to find other   
variables in the kinematic equations -**how does launch angle affect velocity?

This angle prediction **should** hit the target distance. If it doesn’t   
land after three shots,   
*launch and graph the   
curve of best fit again.*



**Angle˚**

**X-Axis** **=** **Distance in Meters**

1 2 3 4 5 6 7 8 9 10



**Angle Prediction**

**Mark** the target   
distance with   
a **line|**



**Remember!** Each   
time you re-design your launcher, make   
another graph.   
*New designs need   
to test new data.*

