

Ball Bounce

Student Activity

7 8 9 **10** 11 12



TI-Nspire™



Activity



Student



50 min

Introduction

The motion of a bouncing ball can be modelled using a parabola. In this activity you will collect position and time data using a CBR (Calculator Based Ranger) as a ball bounces beneath the detector. Things you need to keep in mind while completing this activity:

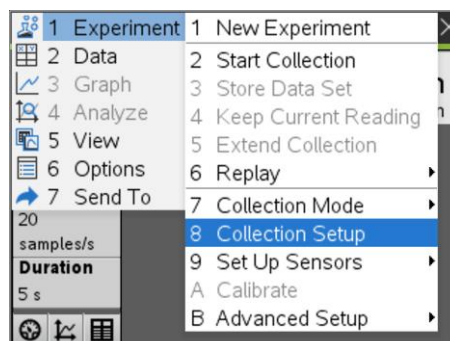
- The data is real, so it won't be 'perfect'. Real world data generally is messy! The idea of developing equations is to model the motion and evaluate the appropriateness and accuracy of the equations.
- It takes less than a minute to collect the data, so if you're not happy with your first data sample, you can quickly and easily generate a new sample.
- Keep the motion sensor at the same height throughout the data collection period. It can be moved side to side to track the bouncing ball, but the height cannot be changed. If the ball has a lot of side-ways movement, consider a more level surface or a smaller bounce height.
- Collect data for approximately 4 to 5 bounces. Images in this activity include sample data.

Set up

Start a new TI-Nspire document, insert a Calculator Application and then connect the CBR to the calculator using the USB cable. The Vernier DataQuest Application will launch automatically.

Once the Application has launched select:

Menu > Experiment > Collection Setup

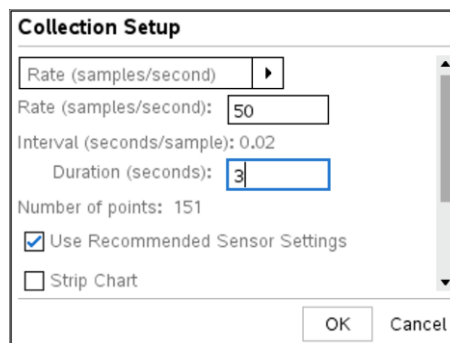


The data collection rate and duration depends on the type of ball and the height from which the ball is released. To begin try the following:

Rate: 50 (Samples per second)

Duration: 3 (Seconds)

Note that this menu also provides the option for remote data collection. If you are collecting the sample by yourself, check the box for remote collection so you can disconnect the CBR and use the Trigger button to initiate data collection.



A basketball bouncing on a smooth and level concrete surface makes an excellent target. If the ball is well inflated and released from a height of approximately 1m, the set up shown will produce between 3 to 5 bounces.

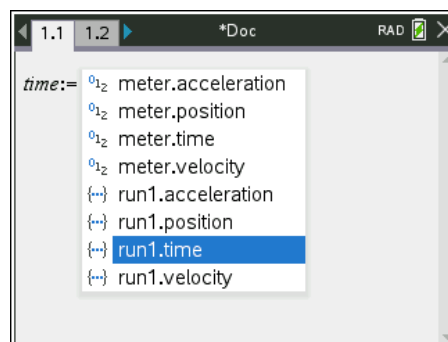
Navigate to the Calculator Application and assign the following variables:

time:= run1.time [Available from the VAR menu]

temp:= run1.position [Available from the VAR menu]

height:= max(temp) - temp

Note: If you collected and saved multiple data sets, select your best run set. run#.time



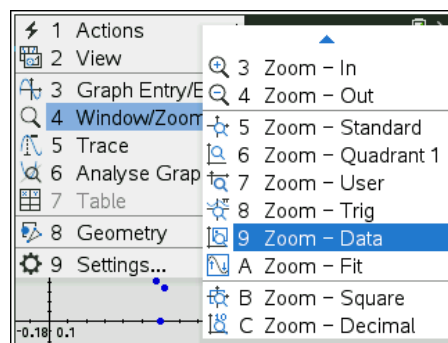
Navigate to the Graph Application and set up a **Scatter Plot** using the **Graph Entry/Edit** menu.

Set Scatter Plot 1 (S1) such that:

x = time

y = height

Use the Window / Zoom option to zoom in on the data. Make sure the x and y axis are visible, if not, adjust the window settings accordingly.



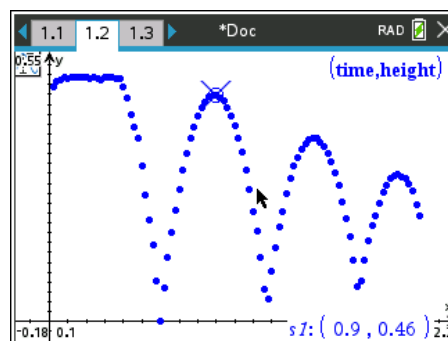
Question: 1

What is the purpose of the instruction: height := max (temp) – temp ?

Modelling Motion using Translations

Use the **Trace > Graph Trace** command to move the cursor across the data set. Navigate to the top of the first ball bounce and record the time and height of the ball.

Note: Each data set is different so do NOT use the values shown opposite.



Question: 2

Use the translational form of a parabola: $f(x) = a(x-h)^2 + k$ to determine appropriate values for the corresponding parameters.

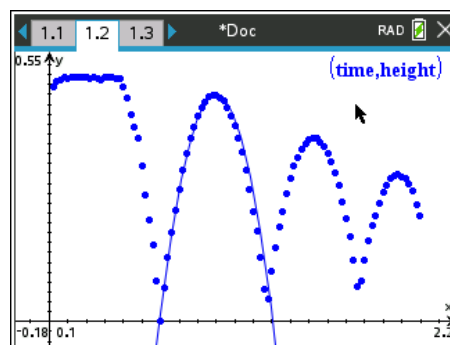
Question: 3

Graph your function and experiment with the value of a so that your parabola models the first ball bounce.

Note: You may choose to also experiment with h and k to create a better fit.

The graph shown opposite illustrates how well the parabola can model the bouncing ball. Further experimentation with the parameters in this equation could produce an even better fit.

Note: The data shown opposite includes a 'flat' spot at the start of the data collection. The flat spot corresponds to the ball being held after the CBR has started data collection.



Question: 4

Model the remaining ball bounces using the Trace command and the technique outlined in Questions 2 and 3.

Question: 5

What similarities do each of the parabolic equations share?

Question: 6

Thinking about transformations and your current equation; suggest and test another way to generate equations for the remaining ball bounces.



**Save your document! The data you have collected is unique, so too are your equations.
The accuracy of your equations is easy to check by using each of the graphs.**

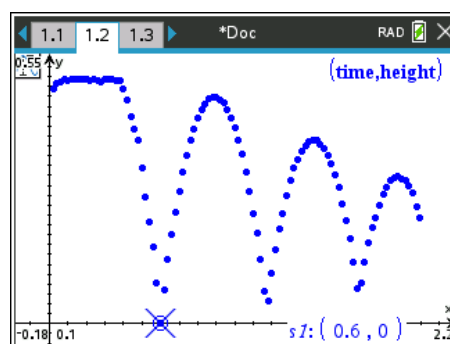
Modelling Motion using Impact Points

Another way to generate an appropriate equation is to consider *when* the ball hit the ground.

Switch the current equations OFF (don't delete them) so that you don't get confused between equations.

Use the **Trace > Graph Trace** commands to locate the closest point to *when* the ball struck the ground.

In the example shown here the first point is actually on the axis, the second point is not. The second point can be estimated using the proximity of neighbouring points.



Question: 7

Use the intercept form of a parabola: $f(x) = a(x - m)(x - n)$, determine appropriate values for the parameters m and n .

Question: 8

Graph your function and experiment with the value of a so that your parabola models the first ball bounce.

Note: You may choose to experiment with m and n to create a better fit.

Question: 9

Explain why changing the value of a has no effect on the axis intercepts.

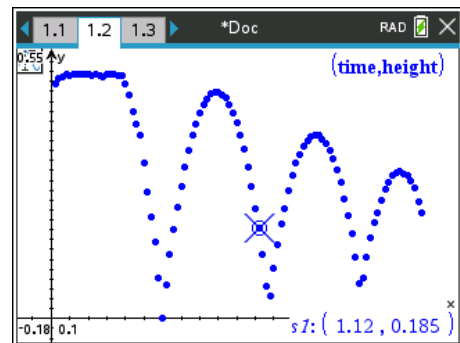
Question: 10

Model the remaining ball bounces using this technique and record your results. [Save your document]

Modelling Motion using Simultaneous Equations

Another way to generate an appropriate equation is to identify appropriately spaced points along the path of the ball bounce and determining the equation that passes through these points.

Switch the current equations OFF (don't delete them) so that you don't get confused between equations. Use the **Trace > Graph Trace** commands to locate three points that you want the parabola to pass through. These points should be well spaced and representative of the path of the ball.



Question: 11

Set up the simultaneous equations using your selected three points. [Include the coordinates of your points]

Question: 12

Determine the equation that passes through your three points.

Question: 13

Determine the remaining equations for each of the ball bounces using this approach.

Extension

The three different approaches include various levels of difficulty, slightly different equations and presumably slightly better or worse models for the data. Which one is best? The purpose of this extension task is to determine which one of the equations provides the best model.

Suppose the equation for the first bounce is modelled by the function: $f(x)$. You can determine *where* the equation predicts the ball to be by considering the corresponding times. Locate the first and last point on the bounce using the trace command as before.

Insert a Spreadsheet Application and enter the list names at the top of each column. Suppose the bounce starts at time = 0.6 and concludes as time = 1.16. Select the cells containing this data, including the height values, (hold down the shift key) and copy (Ctrl + C) these values. Navigate to the top of an empty column (Ctrl + 7 = Top) and paste the data (Ctrl + V).

In a new column generate the corresponding values for the function as shown opposite. (t_1 = Times & h_1 = Heights for bounce one)

	t1	h1	hp	G
=			=f('t1)	
1	0.6	-1.E-16	0.009411	
2	0.62	0.06881	0.068998	
3	0.64	0.130116	0.124358	
4	0.66	0.181806	0.175491	
5	0.68	0.224192	0.222398	

The spreadsheet shown opposite shows hp (height prediction) with the actual values (h_1) for the height of the first ball bounce, notice the accuracy of the predicted height throughout the ball bounce.

Use this approach to determine the 'average' difference between the predicted height and actual height for each ball bounce and its corresponding equations and discuss your findings.