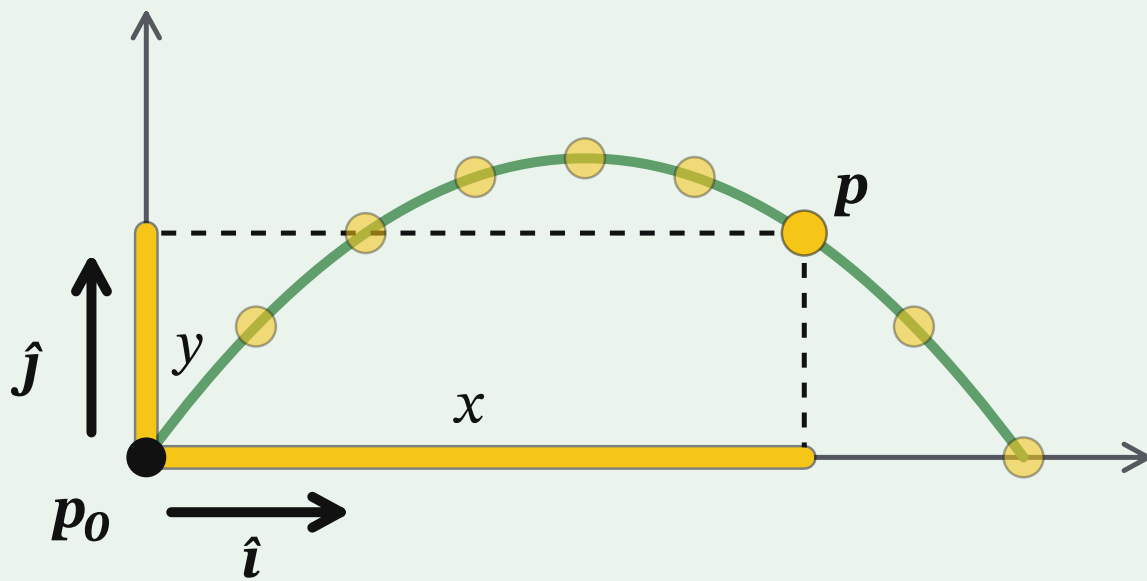


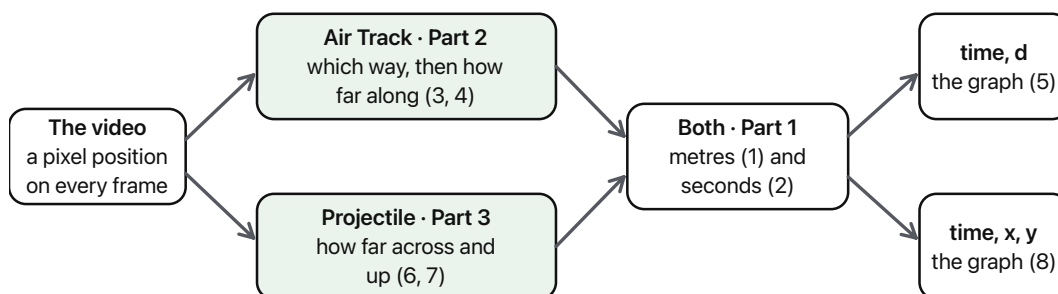
Camera Method

The Mathematics.



Summary

Accelab's Camera method films one of two experiments: a cart on a track (**Air Track**) or an object in flight (**Projectile**). The video gives a pixel position on every frame, but physics needs metres and seconds. This document explains the mathematics in between.



[Figure A] **The route from video to data.** Numbers in brackets are sections.

How the document is organised. [Part 1](#) covers what both share: the scale and the clock. [Part 2](#) is the cart: it keeps to a line, so one distance places it. [Part 3](#) is the projectile: it follows a curve, so it needs two. The [Toolbox](#) holds the mathematics they lean on.

Question	Air Track	Projectile
Which direction?	Fitted from the points: the widest spread (Section 3)	Fixed: across the picture and up it (Section 6), so the phone must be level (Section 7)
How far, in pixels?	One dot product, along the track (Section 4)	Two dot products, across and up (Section 6)
How many metres and seconds?	A known length in the picture (Section 1) and the frames' timestamps (Section 2)	The same
What comes out	Time and distance; a parabola in time (Section 5)	Time, x and y; a line, and a parabola in time and in space (Section 8)

[Table A] **The two experiments side by side.** The third row is shared.

In the code, each experiment is one function in `TrackGeometry.swift`: `makeSplits` and `makePositionSplits`. The worked numbers are illustrative, except in [Section 8.4](#).

Contents

Summary	2
PART 1 · BOTH EXPERIMENTS	4
1 How many metres is a pixel?	5
2 How many seconds?	6
PART 2 · AIR TRACK	7
3 Which way does the track run?	8
3.1 The idea: the widest shadows	8
3.2 Turning "spread" into a number	8
3.3 Finding the angle where V is largest	9
3.4 Check with numbers	9
3.5 Why do it this way?	10
4 How far along it?	11
4.1 Check with numbers	11
5 Why the graph curves	12
PART 3 · PROJECTILE	13
6 Where is the object?	14
6.1 Two fixed directions instead of one fitted one	14
6.2 Check with numbers	15
6.3 Why do it this way?	15
7 Why must the phone be level?	16
7.1 What a tilt does	16
7.2 How much it matters	17
8 Why does the flight curve?	18
8.1 Against time	18
8.2 Against each other: the path	18
8.3 Getting g from the data	19
8.4 Check with a real run	20
TOOLBOX	21
9.1 atan2	21
9.2 Double-angle identities	22
9.3 Rotating the axes	23
9.4 Small angles	23

Part 1

Both experiments

Pixels to metres, frames to seconds.

Whatever is filmed, a video measures in its own units: lengths in pixels and time in frames. Turning those into metres and seconds works in exactly the same way for a cart and for an object in flight, so it comes first. Parts 2 and 3 then show where each experiment's lengths in pixels come from.

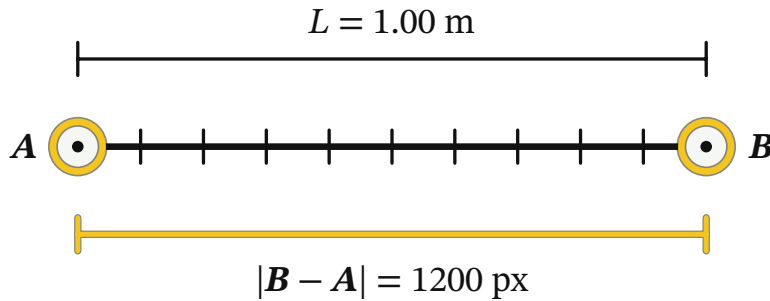
In this part

1 How many metres is a pixel?	5
2 How many seconds?	6

1 How many metres is a pixel?

A pixel has no physical size. The same metre covers few pixels when the camera is far away and many when it is close, so the video needs something of known length: the ruler.

$$k = L / |\mathbf{B} - \mathbf{A}| \quad d = k s$$



[Figure 1.A] **Calibrating the scale.** The ruler's real length L above, and the same span in pixels below.

L is the ruler's real length, and $|\mathbf{B} - \mathbf{A}| = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$ is the pixel distance between its two marked ends. Pythagoras is needed because the ruler may be tilted in the picture.

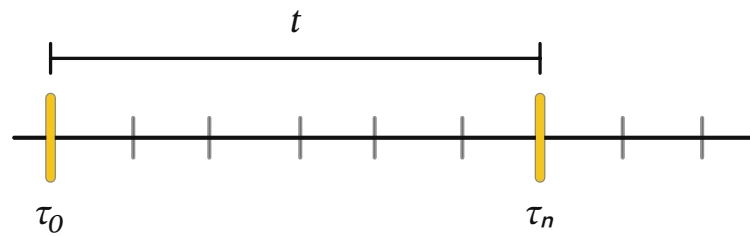
With 1 m spanning 1200 px, $k = 1 / 1200 \text{ m per pixel}$, and a length of $s = 600 \text{ px}$ in the picture becomes $d = 0.50 \text{ m}$.

The hidden assumption. One k is used for the whole picture. That only holds if the camera faces the motion square-on and the ruler is as far from it as the cart or the object in flight, which is what the app's filming checklist protects.

2 How many seconds?

Time comes from each frame's own timestamp in the video file, measured from the first tracked frame:

$$t = \tau_n - \tau_0$$



[Figure 2.A] **Frame timestamps.** Frames are not perfectly evenly spaced; t is the difference between two stamped times.

τ_n is the time stamped on frame n , and τ_0 is the first tracked frame's. Subtracting makes the start $t = 0$, just as every distance is measured from the first tracked point.

Why not frame number $\div 60$? iPhone video frames are not guaranteed to be exactly $1 / 60$ s apart. Counting frames would put that error straight into t .

Timing errors count twice. From $d = \frac{1}{2}at^2$, $a = 2d / t^2$, so a 1% error in t becomes about a 2% error in a .

Part 2

Air Track

A cart on a track: one distance.

A cart keeps to a straight line, so a single number places it: how far along the track it has gone. The track is slanted in the picture, so the app first finds which way it runs, then measures along it. Part 1 turns that distance into metres and its frame into seconds.

In this part

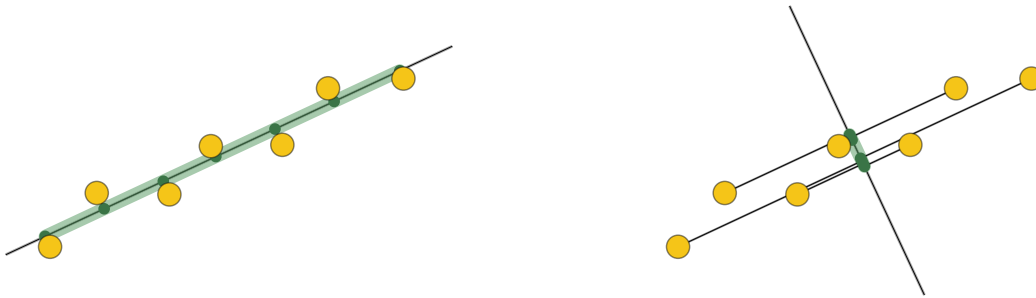
3 Which way does the track run?	8
4 How far along it?	11
5 Why the graph curves	12

3 Which way does the track run?

The camera is never perfectly level and the track is tilted, so on screen the track is a slanted line. Before measuring distance along it, the app has to know which way it points. It does this by finding the direction in which the tracked points are most spread out.

3.1 The idea: the widest shadows

Pick any direction and drop every point straight onto a line in that direction, like casting shadows. If the direction matches the track, the shadows spread out wide. If it runs across the track, they bunch together. So the track's direction is the one where the shadows are most spread out.



[Figure 3.1.A] **Shadows along and across the track.** Left: a line along the track, where the shadows (dark dots) spread wide. Right: a line across the track, where they bunch up.

3.2 Turning "spread" into a number

First subtract the average position, so the points are centred on the origin: $a = x - \bar{x}$ and $b = y - \bar{y}$. Only the direction matters, not where the line sits.

A direction at angle φ is the unit vector $(\cos \varphi, \sin \varphi)$. The shadow of the point (a, b) on it is a dot product:

$$\text{shadow} = a \cos \varphi + b \sin \varphi$$

Shadows can be positive or negative and would cancel if simply added, so square them first, then add. Call the total V :

$$V(\varphi) = \sum (a \cos \varphi + b \sin \varphi)^2 = S_{xx} \cos^2 \varphi + 2S_{xy} \sin \varphi \cos \varphi + S_{yy} \sin^2 \varphi$$

$$S_{xx} = \sum a^2 \quad S_{xy} = \sum ab \quad S_{yy} = \sum b^2$$

The three sums are not a special technique. They are what is left when the bracket is expanded: how far the points spread in x , in y , and how much x and y move together. They are computed once; after that only φ changes.

3.3 Finding the angle where V is largest

The double-angle identities ([Section 9.2](#)) turn the squares and the product into single terms in 2φ :

$$V(\varphi) = \text{constant} + A \cos 2\varphi + B \sin 2\varphi$$

$$A = (S_{xx} - S_{yy})/2 \qquad B = S_{xy}$$

The part that changes is itself a dot product of two vectors:

$$A \cos 2\varphi + B \sin 2\varphi = (A, B) \cdot (\cos 2\varphi, \sin 2\varphi)$$

$(\cos 2\varphi, \sin 2\varphi)$ is a unit vector pointing at angle 2φ . A dot product is largest when the two vectors point the same way, so 2φ must be the angle of the vector (A, B) . Doubling that vector does not change its angle, which gives the formula in the code:

$$\varphi = \frac{1}{2} \text{atan2}(2S_{xy}, S_{xx} - S_{yy}) \qquad \mathbf{u} = (\cos \varphi, \sin \varphi)$$

The 2 and the $\frac{1}{2}$ both come from the double-angle identities. $\mathbf{u}$ has length 1 because $\cos^2 \varphi + \sin^2 \varphi = 1$, which matters in [Section 4](#). The function `atan2` is explained in [Section 9.1](#).

3.4 Check with numbers

Take three points on a line of gradient $\frac{1}{2}$: $(-2, -1)$, $(0, 0)$ and $(2, 1)$. Their average is already $(0, 0)$.

$$S_{xx} = 8 \qquad S_{yy} = 2 \qquad S_{xy} = 4$$

$$\varphi = \frac{1}{2} \text{atan2}(8, 6) = \frac{1}{2} \times 53.13^\circ = 26.57^\circ$$

The line's real angle is $\tan^{-1}(\frac{1}{2}) = 26.57^\circ$. They agree.

3.5 Why do it this way?

Why all the points, not just the first and last? Every tracked point is a few pixels off. With two points, those two errors decide the direction. With all of them, the errors largely cancel.

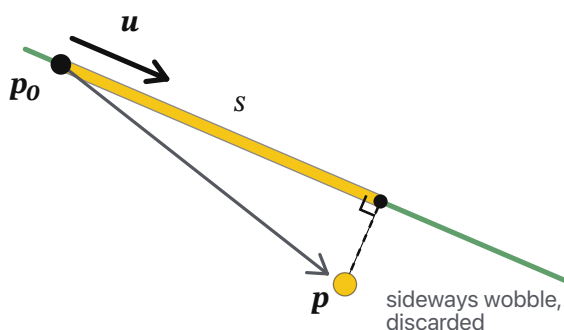
Why not an ordinary line of best fit? The usual $y = mx + c$ fit assumes x is exact and only y has error, and its gradient blows up when the track looks vertical. Here x and y are both uncertain, so the fit uses perpendicular distances and works at any tilt.

Why are widest shadows and smallest perpendicular distances the same thing? For each point, Pythagoras gives $(\text{distance from centre})^2 = (\text{shadow})^2 + (\text{perpendicular distance})^2$. The left side is fixed by where the point is, so making the shadows as large as possible makes the perpendicular distances as small as possible.

4 How far along it?

$\mathbf{p}_0$ is the first tracked point, the start. $\mathbf{p} - \mathbf{p}_0$ is the cart's displacement from the start, so the first point is automatically at zero. Its distance along the track is a dot product with the unit vector from [Section 3](#):

$$s = (\mathbf{p} - \mathbf{p}_0) \cdot \mathbf{u}$$



[Figure 4.A] **Projecting a point onto the track.** s is measured along the track from the start.

That displacement has a part along the track and a small sideways part from tracking error. By definition $\mathbf{v} \cdot \mathbf{u} = |\mathbf{v}| |\mathbf{u}| \cos \alpha$, and because $|\mathbf{u}| = 1$ this is just $|\mathbf{v}| \cos \alpha$: the length of the shadow of $\mathbf{v}$ on the track, already in pixels. The sideways part is perpendicular to $\mathbf{u}$, so its dot product is zero and it drops out.

Why not just the length $|\mathbf{p} - \mathbf{p}_0|$? That would count the sideways wobble as distance, so every value would come out slightly too large.

Which way is positive? A line has two directions, $\mathbf{u}$ and $-\mathbf{u}$, and [Section 3](#) returns one arbitrarily. If the last point comes out negative, every s is multiplied by -1 : the cart only rolls downhill, so the last point marks down-slope.

4.1 Check with numbers

With $\mathbf{u} = (0.8, 0.6)$ and $\mathbf{p} - \mathbf{p}_0 = (468, 376)$:

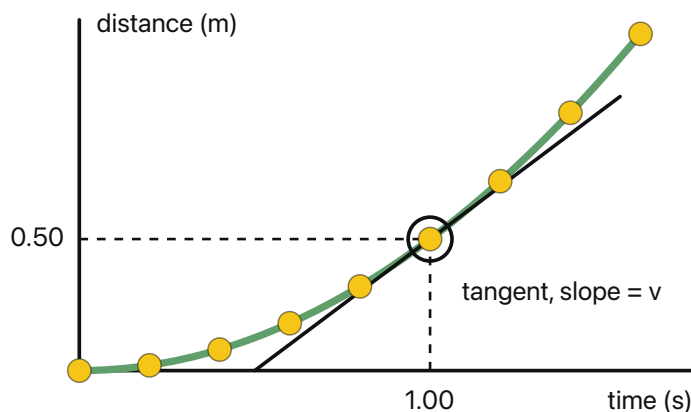
$$s = 468 \times 0.8 + 376 \times 0.6 = 374.4 + 225.6 = 600 \text{ px}$$

The displacement is really $600\mathbf{u} = (480, 360)$ plus a wobble of $(-12, 16)$. The wobble dotted with $\mathbf{u}$ is $-9.6 + 9.6 = 0$, so it has no effect.

5 Why the graph curves

Each point is now one CSV row of time and distance. On a distance–time graph the slope is the speed. The cart starts at rest and speeds up, so the graph starts flat and gets steeper.

$$a = g \sin \theta \qquad v = at \qquad d = \frac{1}{2}at^2$$



[Figure 5.A] **Distance against time.** Points at equal time steps on a parabola. The ringed point is 0.50 m at 1.00 s, and the straight line is the tangent there.

The slope of the graph at any instant is the derivative of distance with respect to time, which is the speed. The acceleration is the derivative of speed in the same way:

$$v = \frac{dd}{dt} = at \qquad a = \frac{dv}{dt}$$

The tangent line in the figure shows this slope at the ringed point. There $t = 1.00$ s, so its slope is $1 \times 1.00 = 1.00$ m/s. Earlier points have flatter tangents and later points steeper ones, which is what a changing speed looks like.

The only unbalanced force along the track is a constant component of gravity, so the acceleration is constant, the speed grows in proportion to time, and the distance grows with time squared: a parabola. The example point fits. With $a = 1 \text{ m/s}^2$, $t = 1.00$ s gives $d = 0.50$ m, the row 1.0000, 0.5000.

To get a from the data, plot v against t (slope a) or d against t^2 (slope $\frac{1}{2}a$), then compare with $g \sin \theta$. Accelab stops at exporting d and t ; that analysis is the student's part of the lab.

Part 3

Projectile

An object in flight: two distances.

An object in flight follows a curve, so there is no direction to fit and one number cannot place it. The app measures how far it has gone across and how far up, along the picture's own two edges, which is why the phone has to be level. Part 1's scale and clock are used unchanged.

In this part

6 Where is the object?	14
7 Why must the phone be level?	16
8 Why does the flight curve?	18

6 Where is the object?

An object in flight does not keep to a line, so [Sections 3](#) and [4](#) cannot be used as they are. There is no track direction to find, and the direction of widest spread of a curved path is only a slanted compromise that means nothing physical. What physics wants instead is two numbers on each frame: how far the object has gone across, x , and how far up, y , both measured from where it started.

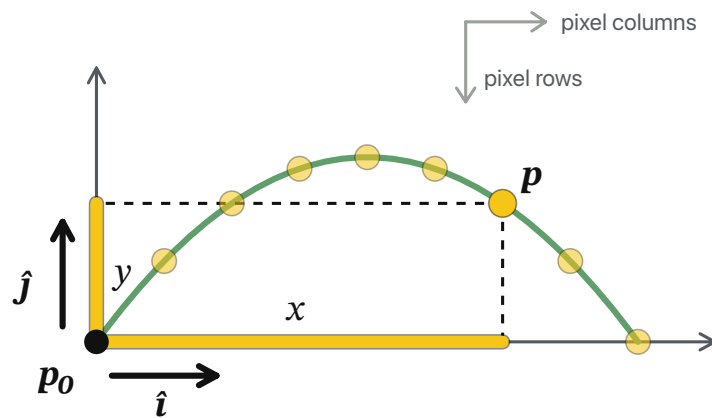
6.1 Two fixed directions instead of one fitted one

The picture has two directions built in: along its width and up its height. If the phone is level, these are the real horizontal and vertical. As unit vectors, where σ is either 1 or -1:

$$\hat{\mathbf{i}} = (\sigma, 0) \quad \hat{\mathbf{j}} = (0, -1)$$

Then each position is the dot product of [Section 4](#), once with each, and the scale k from [Section 1](#) turns pixels into metres:

$$x = k (\mathbf{p} - \mathbf{p}_0) \cdot \hat{\mathbf{i}} \quad y = k (\mathbf{p} - \mathbf{p}_0) \cdot \hat{\mathbf{j}}$$



[Figure 6.1.A] **Measuring across and up.** x and y are the two parts of the displacement from the start $\mathbf{p}_0$ to the object $\mathbf{p}$.

$\hat{\mathbf{i}}$ and $\hat{\mathbf{j}}$ are perpendicular and each has length 1, so x and y are the two **components** of the displacement. Together they hold everything it does. For the cart, the part across the track was thrown away as wobble; here that part is the height, which is half the answer.

Why the -1 in $\hat{\mathbf{j}}$? On a screen, pixel rows are numbered from the top, so the picture's own y grows downwards. Height grows upwards. $\hat{\mathbf{j}}$ points up the picture, which turns “fewer rows down” into “more height”.

What is σ ? It is +1 if the last tracked point is to the right of the first and -1 if it is to the left, so x counts up in the direction the object travels, whichever way it was thrown. This is the cart's rule again: distance counts up down-slope. Height is never given a sign this way. Up is up wherever the object lands; if y took its sign from the motion, an object pushed off a table, which only ever goes down, would come out with gravity pointing upwards.

6.2 Check with numbers

Use the scale from [Section 1](#), $k = 1 / 1200$ m per pixel. An object is thrown to the right, so $\sigma = 1$. Half a second later it is 1200 px to the right of its start and 330 px higher up the picture, which is 330 rows *fewer*: $\mathbf{p} - \mathbf{p}_0 = (1200, -330)$ px.

$$x = (1200 \times 1 + (-330) \times 0) / 1200 = 1.000 \text{ m}$$

$$y = (1200 \times 0 + (-330) \times (-1)) / 1200 = 0.275 \text{ m}$$

Thrown to the left instead, the displacement is $(-1200, -330)$ px and $\sigma = -1$. Then $x = (-1200 \times -1) / 1200 = 1.000$ m again, and y is untouched. The CSV row is 0.5000, 1.00000, 0.27500 either way.

6.3 Why do it this way?

Why is the first point the origin? Subtracting $\mathbf{p}_0$ makes the first row $(0, 0)$ at $t = 0$. The equations of motion then need no starting position, only a starting velocity ([Section 8](#)).

Why one scale for both directions? A pixel is as tall as it is wide, so the k measured along the ruler holds for heights too. It rests on the same hidden assumption as before: the camera square-on, and the ruler as far away as the object.

Why not let the student place the axes? Some video-analysis tools do. Fixing them to the picture means nothing to set up and nothing to get wrong on screen, at the price of one rule while filming: keep the phone level. [Section 7](#) shows what that rule is worth.

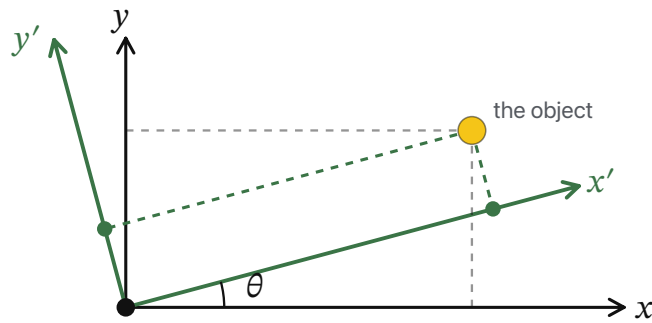
7 Why must the phone be level?

The cart's direction is found from the cart's own points. Tilt the phone and the fitted u tilts with it, so nothing is lost. A projectile has no such protection: its axes are the picture's edges, and if the phone is rolled by an angle θ , those edges are not the real horizontal and vertical.

7.1 What a tilt does

Call the true coordinates (x, y) and what the tilted picture measures (x', y') . Turning the axes by θ mixes the two ([Section 9.3](#)):

$$x' = x \cos \theta + y \sin \theta \qquad y' = y \cos \theta - x \sin \theta$$



[Figure 7.1.A] **A tilted picture.** Its axes (green) are turned by θ from the true ones, so it reads a slightly different x and y for the same object.

Put in the true motion from [Section 8](#), $x = v_x t$ and $y = v_y t - \frac{1}{2} g t^2$:

$$\begin{aligned} y' &= (v_y \cos \theta - v_x \sin \theta) t - \frac{1}{2} (g \cos \theta) t^2 \\ x' &= (v_x \cos \theta + v_y \sin \theta) t - \frac{1}{2} (g \sin \theta) t^2 \end{aligned}$$

Two things have happened. In y' , g has become $g \cos \theta$. In x' , a t^2 term has appeared from nowhere: the object seems to accelerate sideways at $g \sin \theta$.

7.2 How much it matters

The size of each effect depends only on the tilt.

Tilt θ	Measured $g \cos \theta$	Error in g	Sideways $g \sin \theta$
1°	9.809 m/s ²	0.02% low	0.17 m/s ²
2°	9.804 m/s ²	0.06% low	0.34 m/s ²
5°	9.773 m/s ²	0.4% low	0.85 m/s ²
10°	9.661 m/s ²	1.5% low	1.70 m/s ²

[Table 7.2.A] **The effect of a tilted phone**, taking $g = 9.81 \text{ m/s}^2$.

For a small tilt $\cos \theta$ stays very close to 1 while $\sin \theta$ grows at once ([Section 9.4](#)). So g from the y - t graph survives a careless phone, but the x - t graph, which should be a straight line, bends. **A curved x - t graph is the sign of a tilted phone.**

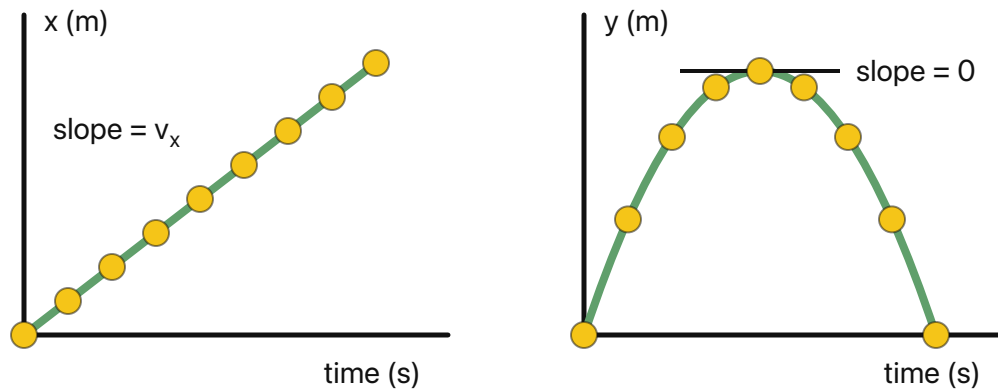
8 Why does the flight curve?

Each point is now one CSV row of time, x and y . Two different graphs come out of it.

8.1 Against time

Once the object has left the hand, the only force on it is gravity, straight down (ignoring the air). Nothing changes its sideways speed v_x , and its upward speed, which starts at v_y , falls by g every second:

$$x = v_x t \qquad y = v_y t - \frac{1}{2}gt^2$$



[Figure 8.1.A] **Against time.** Points at equal time steps. x climbs a straight line; y climbs, stops and falls along a parabola.

The y - t graph is the cart's graph from [Section 5](#) turned upside down, with g in place of $g \sin \theta$: a track tilted to 90° would simply be a drop. Its slope is the upward speed, $dy/dt = v_y - gt$. At the top that slope is zero, which gives the time and height of the peak:

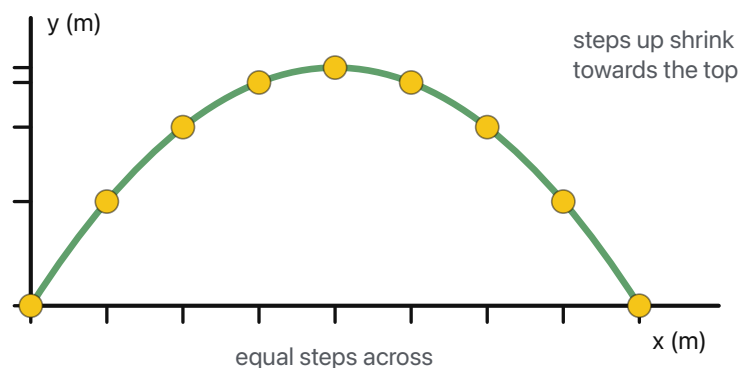
$$t_{\text{top}} = v_y / g \qquad y_{\text{top}} = v_y^2 / 2g$$

8.2 Against each other: the path

From the first equation $t = x / v_x$. Putting that into the second removes time altogether:

$$y = (v_y / v_x) x - (g / 2v_x^2) x^2$$

This is a quadratic in x , so the path through the air is itself a parabola, in space this time rather than in time. It is the yellow trail on the video, and the curve the Analysing screen draws.



[Figure 8.2.A] **Height against distance.** The same points as Figure 8.1.A. That screen uses one scale for both axes, the smaller of width $\div$ range of x and height $\div$ range of y , so the path keeps its true shape.

8.3 Getting g from the data

Accelab stops at exporting t , x and y ; finding g is the student's part of the lab. Three ways, from the same rows:

Fit the parabola. Fit $y = bt - ct^2$ to the y - t graph. Then $g = 2c$.

Plot the speed. Between two rows the upward speed is about $\Delta y / \Delta t$. Plotted against time it is a straight line of slope $-g$.

Difference twice. At equal time steps each Δy is smaller than the one before by $g \Delta t^2$. At 60 frames a second that is $9.81 / 3600 \approx 2.7$ mm, which in most shots is only a pixel or two, about the size of the tracking error itself, so this way is the noisiest. A fit uses every point, and the errors largely cancel, as in [Section 3.5](#).

8.4 Check with a real run

Unlike the other numbers here, these were measured. A generated clip of an object launched at $(2.4, 3.3)$ m/s under $g = 9.81$ m/s², drawn at 600 px per metre and 60 frames a second beside a 1 m ruler, was put through the app: imported, calibrated on the ruler, tracked automatically over 46 frames and exported.

Clip	g from $y(t)$	v_x from $x(t)$	Largest error
Thrown to the right	9.798 m/s ²	2.399 m/s	2.1 mm
Thrown to the left	9.814 m/s ²	2.399 m/s	3.1 mm

[Table 8.4.A] **Recovered from the exported CSV.** The true values are 9.81 m/s² and 2.4 m/s; 1 px is 1.7 mm.

A generated clip is the kindest possible test: a sharp object, a level camera and no perspective. Real footage will do worse, mostly through blur and a phone that is not quite square-on.

9 Toolbox

[Section 3](#) leans on two pieces of mathematics that are worth stating on their own, and Part 3 leans on two more.

9.1 atan2

$\text{atan2}(y, x)$ returns the angle of the vector (x, y) measured from the x -axis. It is a programming function, not a maths symbol, and y comes first.

$\tan^{-1}(y / x)$ has two problems. Dividing throws information away: $(1, 1)$ and $(-1, -1)$ point in opposite directions, but both give $y / x = 1$. And when $x = 0$ the division fails. atan2 takes x and y separately, reads their signs, and covers all four quadrants, from -180° to 180° .

Vector (x, y)	$\tan^{-1}(y / x)$	$\text{atan2}(y, x)$
$(1, 1)$	45°	45°
$(-1, 1)$	-45° , wrong	135°
$(-1, -1)$	45° , wrong	-135°
$(0, 1)$	cannot be computed	90°

[Table 9.1.A] **Inverse tangent against atan2** for four vectors.

Both problems really happen in [Section 3](#). If the points spread more vertically than horizontally, $S_{xx} - S_{yy}$ is negative and $\tan^{-1}$ would pick the narrowest direction instead of the widest. If the track sits at exactly 45° , $S_{xx} = S_{yy}$ and the division is by zero.

9.2 Double-angle identities

These write the sine and cosine of 2θ in terms of θ . They come from the compound-angle identities for $\sin(A + B)$ and $\cos(A + B)$ with $A = B = \theta$.

$$\sin 2\theta = 2 \sin \theta \cos \theta$$

$$\cos 2\theta = \cos^2\theta - \sin^2\theta = 2\cos^2\theta - 1 = 1 - 2\sin^2\theta$$

Rearranging the last two gives the forms used in [Section 3](#):

$$\cos^2\theta = (1 + \cos 2\theta)/2 \qquad \sin^2\theta = (1 - \cos 2\theta)/2$$

A quick check at $\theta = 30^\circ$: $\sin 60^\circ = 0.866$ and $2 \times 0.5 \times 0.866 = 0.866$; $\cos 60^\circ = 0.5$ and $0.75 - 0.25 = 0.5$.

In [Section 3](#) they replace $\cos^2\varphi$, $\sin^2\varphi$ and $\sin \varphi \cos \varphi$ with single terms in 2φ . That is what lets V be read as a dot product, and it is why 2φ and $\frac{1}{2}$ appear in [the final formula](#).

9.3 Rotating the axes

Turn a pair of axes anticlockwise by θ . The new x -axis points along the unit vector $(\cos \theta, \sin \theta)$ and the new y -axis along $(-\sin \theta, \cos \theta)$. A point's new coordinates are its dot products with those two, exactly as in [Section 4](#):

$$\begin{aligned}x' &= x \cos \theta + y \sin \theta \\y' &= -x \sin \theta + y \cos \theta\end{aligned}$$

The point has not moved; only its description has changed. Its distance from the origin is the same either way, because $x'^2 + y'^2 = x^2 + y^2$.

A quick check at $\theta = 30^\circ$ with the point $(2, 1)$: $x' = 2 \times 0.866 + 1 \times 0.5 = 2.232$ and $y' = -2 \times 0.5 + 1 \times 0.866 = -0.134$, and $2.232^2 + 0.134^2 = 5.00 = 2^2 + 1^2$.

In [Section 7](#) the new axes are the edges of a tilted picture, and the true motion is rewritten in them.

9.4 Small angles

Measure a small angle θ in radians (degrees $\times \pi / 180$). Then, to a very good approximation:

$$\sin \theta \approx \theta \qquad \cos \theta \approx 1 - \frac{1}{2}\theta^2$$

θ	θ in radians	$\sin \theta$	$\frac{1}{2}\theta^2$	$1 - \cos \theta$
1°	0.01745	0.01745	0.00015	0.00015
2°	0.03491	0.03490	0.00061	0.00061
5°	0.08727	0.08716	0.00381	0.00381
10°	0.17453	0.17365	0.01523	0.01519

[Table 9.4.A] **The small-angle approximations at four angles.** Each pair of columns agrees.

The sine is in proportion to the angle, but the cosine falls short of 1 only by the angle *squared*, which for a small angle is far smaller. That is the whole of [Section 7.2](#): a slightly tilted phone bends the x - t graph, through $\sin \theta$, long before it spoils g , through $\cos \theta$.