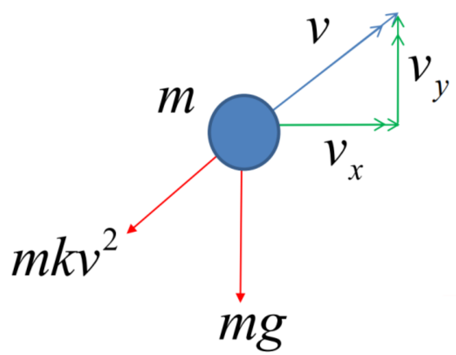
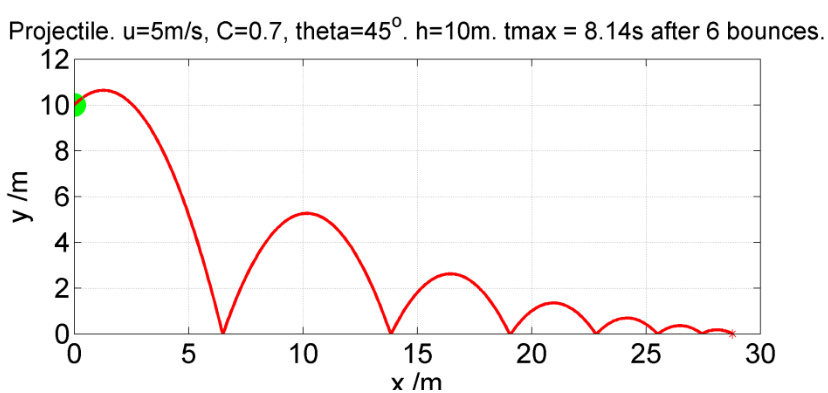
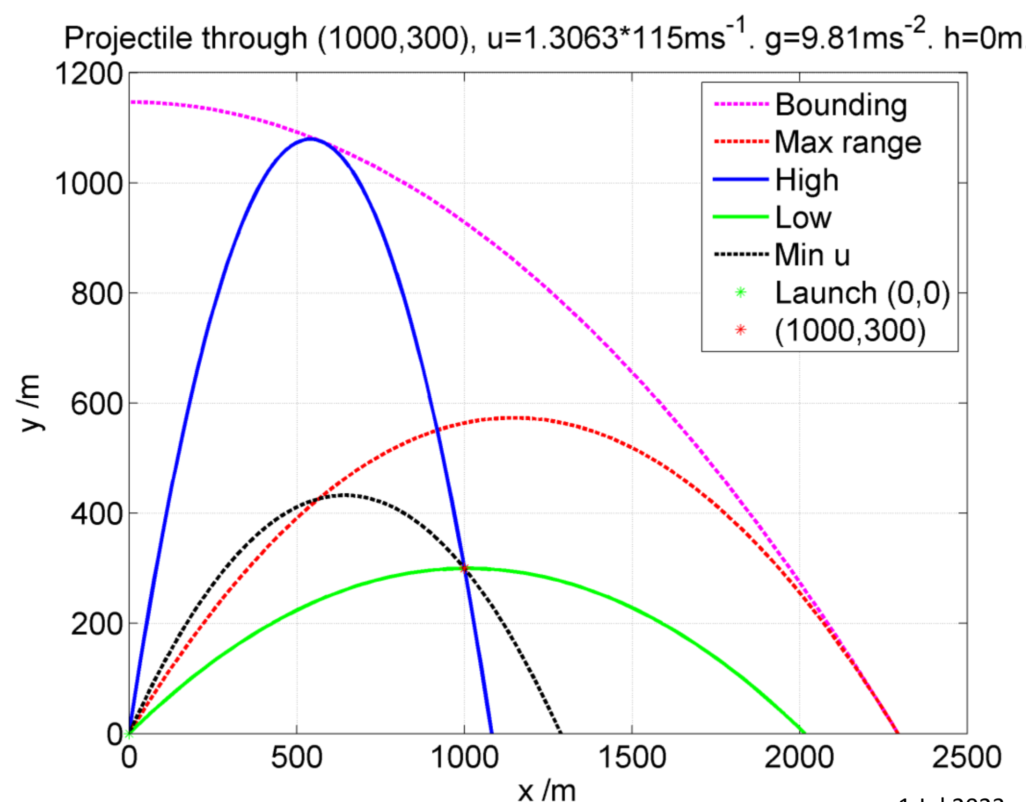



# BPhO Computational Challenge

# 2024 Projectiles




BPhO
British Physics Olympiad

v1 Jul 2023

**Instructions:** Welcome to the **British Physics Olympiad Computational Challenge 2024.** The goal is to build computer models based upon the instructions in this document. Much can be achieved using a *spreadsheet* such as Microsoft Excel, although you are encouraged to use a *programming language* of your choice* for the more sophisticated models and graphical visualizations.

The challenge runs from **Easter 2024 till August 2024.** To submit an entry you will need to fill in a web form. There may be a small administration charge of, payable online as per other BPhO competition entries.

The deliverable of the challenge is to produce a **screencast** of ***maximum length two minutes*** which describes your response to the challenge, i.e. the graphs and the code & spreadsheets and your explanation of these. The videos must be uploaded to **YouTube**, and we recommend you set these as *Unlisted* with *Comments disabled*. **Your entry will comprise a YouTube link.** *Instructions how to do this are at the end of this presentation.* To produce the screencast, we recommend the Google Chrome add-on [Screencastify.]

You can enter the challenge **individually** or in **pairs**. If you opt for the latter, *both* of you must make equal contributions to the screencast.

**Gold**, Silver or **Bronze** e-certificates will be emailed to each complete entry, and the **top five** Golds will be invited to present their work at a special ceremony. You should receive a result by December 2024. Note no additional feedback will be provided, and the decision of the judges is final.

**Bronze:** Initial spreadsheet-based challenge elements completed, some basic coding.
**Silver:** All the spreadsheet-based elements completed, and a commendable attempt at the programming-based elements. Most tasks completed to a reasonable standard.
**Gold:** All tasks completed to a high standard, with possible extension work such as the construction of apps (i.e. programs with graphical user interfaces), significant development of the models, attempt at extension work, short research papers etc.

*__MATLAB__ or **Python** is recommended, although any system that can easily execute code in loops and plot graphs will do. e.g. **Octave, Java, Javascript, C#, C++...** Use what you can access and feel comfortable with. Programming resources

## How to make a screencast using Screencastify and upload this to Youtube

1. Download the Google Chrome web browser
2. Download the Screencastify add-on to Chrome. The free educational version will allow up to 5 minutes of video.
3. When you are ready to make your video (have all the program windows open in advance, and prepare what you are going to say), click on the Screencastify arrow in the corner of your browser. Follow the instructions to record a screen, and a three second countdown will begin.
4. Record your video!
5. Export your video to a **.webm** or **.mp4** file. There is also a direct to YouTube upload option.
6. Upload your video to YouTube (you will need to set up an account first and establish a Channel).
7. Navigate to your video and copy to the clipboard the YouTube weblink. Submit this link in your submission form in the BPhO website.
8. It is recommended that (i) you *don't* have a presenter image in your video (you can turn off this in Screencastify) , i.e. **only have a voice-over**. Also *turn off Comments* in YouTube and make the video *Unlisted.* This means nobody can leave comments, and only those with the link will find your video.

Screencastify

**Exact model (no air resistance) using constant acceleration motion of a particle of mass $m$**

The *only* acceleration is $g$ downwards!

$$x = u_x t$$

$$y = h + u_y t - \tfrac{1}{2} g t^2$$

$$v_x = u_x$$

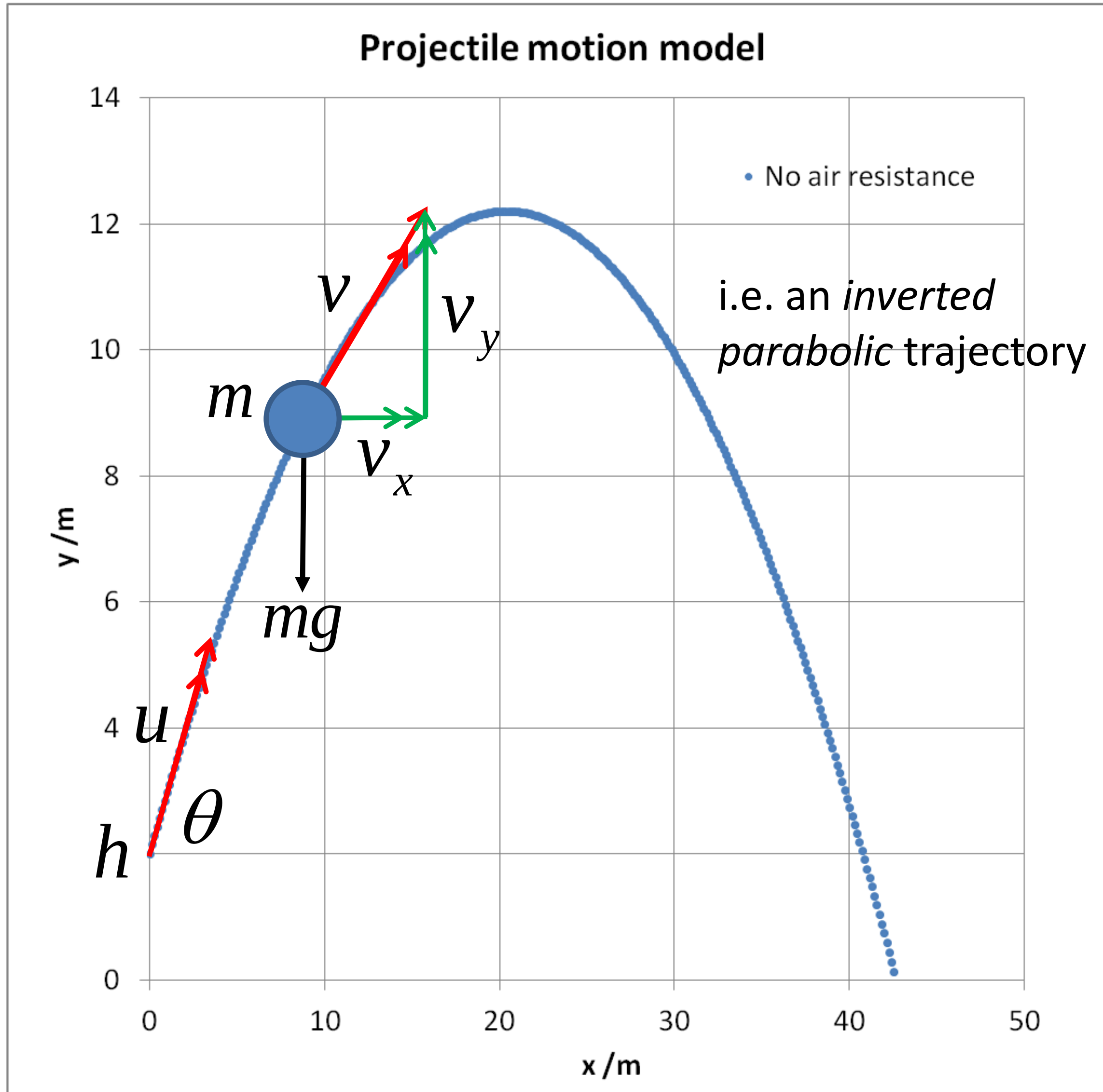
$$v_y = u_y - gt$$

$$v = \sqrt{v_x^2 + v_y^2}$$

Initial $x$ and $y$ velocities

$$u_x = u\cos\theta$$

$$u_y = u\sin\theta$$



**Fixed timestep projectile motion model**

Dr A. French. 14/7/2023

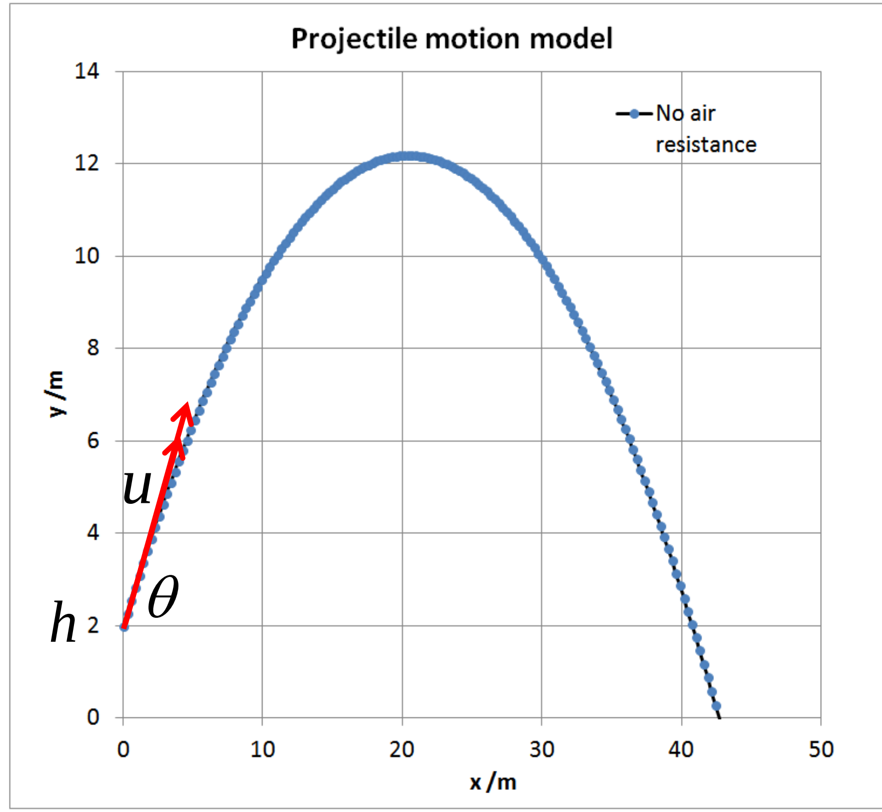
**Inputs**

| | |
|---|---|
| launch angle /deg | 45 |
| launch angle /rad | 0.7854 |
| launch speed /ms^-1 | 20 |
| launch height /m | 2 |
| g /ms^-2 | 9.81 |

| | |
|---|---|
| Time step /s | 0.02 |

| | |
|---|---|
| vx (m/s) | 14.142 |
| initial vy (m/s) | 14.142 |

| | No air resistance model | | | | |
|---|---|---|---|---|---|
| t /s | vx | vy | v | x | y |
| 0 | 14.142 | 14.142 | 20 | 0 | 2 |
| 0.02 | 14.142 | 13.946 | 19.862 | 0.2828 | 2.2809 |
| 0.04 | 14.142 | 13.75 | 19.724 | 0.5657 | 2.5578 |
| 0.06 | 14.142 | 13.554 | 19.588 | 0.8485 | 2.8309 |
| 0.08 | 14.142 | 13.357 | 19.453 | 1.1314 | 3.1 |
| 0.1 | 14.142 | 13.161 | 19.319 | 1.4142 | 3.3652 |
| 0.12 | 14.142 | 12.965 | 19.186 | 1.6971 | 3.6264 |
| 0.14 | 14.142 | 12.769 | 19.054 | 1.9799 | 3.8838 |
| 0.16 | 14.142 | 12.573 | 18.923 | 2.2627 | 4.1372 |
| 0.18 | 14.142 | 12.376 | 18.793 | 2.5456 | 4.3867 |
| 0.2 | 14.142 | 12.18 | 18.664 | 2.8284 | 4.6322 |
| 0.22 | 14.142 | 11.984 | 18.537 | 3.1113 | 4.8739 |
| 0.24 | 14.142 | 11.788 | 18.411 | 3.3941 | 5.1116 |
| 0.26 | 14.142 | 11.592 | 18.286 | 3.677 | 5.3454 |
| 0.28 | 14.142 | 11.395 | 18.162 | 3.9598 | 5.5752 |
| 0.3 | 14.142 | 11.199 | 18.039 | 4.2426 | 5.8012 |
| 0.32 | 14.142 | 11.003 | 17.918 | 4.5255 | 6.0232 |
| 0.34 | 14.142 | 10.807 | 17.798 | 4.8083 | 6.2413 |
| 0.36 | 14.142 | 10.611 | 17.68 | 5.0912 | 6.4555 |
| 0.38 | 14.142 | 10.414 | 17.563 | 5.374 | 6.6657 |
| 0.4 | 14.142 | 10.218 | 17.447 | 5.6569 | 6.8721 |
| 0.42 | 14.142 | 10.022 | 17.333 | 5.9397 | 7.0745 |
| 0.44 | 14.142 | 9.8257 | 17.22 | 6.2225 | 7.2729 |




i.e. ignore air resistance

**Challenge #1: Create a simple model of *drag-free* projectile motion in a spreadsheet or via a programming language. Inputs are: launch angle from horizontal $\theta$, strength of gravity $g$, launch speed $u$ and launch height $h$. Use a fixed increment of time. The graph must automatically update when inputs are changed.**

$$x = u_x t$$

$$y = h + u_y t - \tfrac{1}{2} g t^2$$

$$v_x = u_x \qquad u_x = u\cos\theta$$

$$v_y = u_y - gt \qquad u_y = u\sin\theta$$

$$v = \sqrt{v_x^2 + v_y^2}$$

**PROJECTILE MODEL. Ignore air resistance.**

Dr A. French. 14/07/2023

| | |
|---|---|
| strength of gravity g (N/kg) | 9.81 |
| launch elevation (deg) | 42 |
| launch elevation theta /rad | 0.733 |
| launch speed u (m/s) | 10 |
| launch height h (m) | 1 |
| Range R (m) | 11.15 |
| Apogee xa (m) | 5.07 |
| Apogee ya (m) | 3.28 |
| Time of flight T (s) | 1.50 |

$$x_a = \frac{u^2}{g}\sin\theta\cos\theta$$

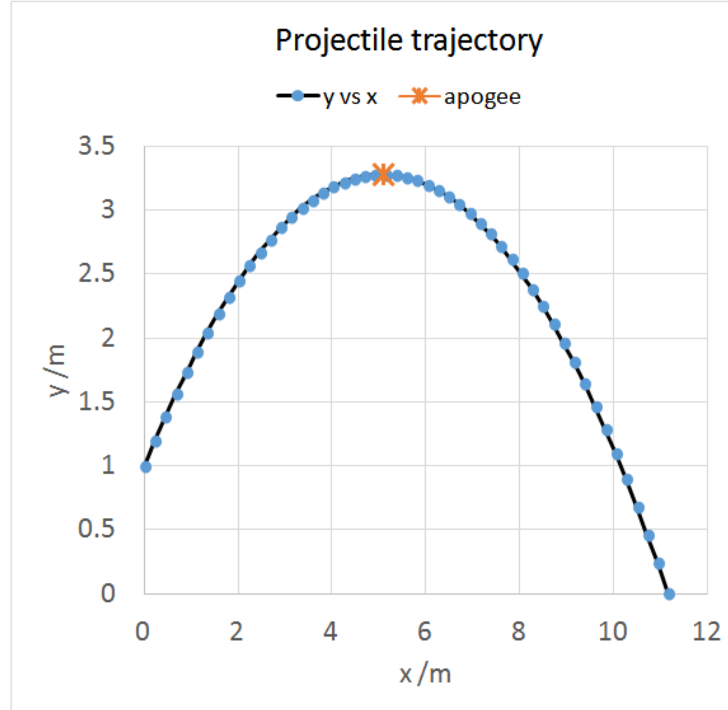
$$T = \frac{R}{u\cos\theta}$$

$$y_a = h + \frac{u^2}{2g}\sin^2\theta$$

$$R = \frac{u^2}{g}\left(\sin\theta\cos\theta + \cos\theta\sqrt{\sin^2\theta + \frac{2gh}{u^2}}\right)$$

fraction of range $R$

| | x /m | y /m |
|---|---|---|
| 0 | 0 | 1 |
| 0.02 | 0.223 | 1.1963 |
| 0.04 | 0.4459 | 1.3838 |
| 0.06 | 0.6689 | 1.5625 |
| 0.08 | 0.8918 | 1.7324 |
| 0.1 | 1.1148 | 1.8934 |
| 0.12 | 1.3377 | 2.0456 |
| 0.14 | 1.5607 | 2.1889 |
| 0.16 | 1.7837 | 2.3234 |
| 0.18 | 2.0066 | 2.4491 |
| 0.2 | 2.2296 | 2.566 |
| 0.22 | 2.4525 | 2.674 |
| 0.24 | 2.6755 | 2.7732 |
| 0.26 | 2.8984 | 2.8636 |
| 0.28 | 3.1214 | 2.9452 |
| 0.3 | 3.3443 | 3.0179 |
| 0.32 | 3.5673 | 3.0818 |
| 0.34 | 3.7903 | 3.1368 |
| 0.36 | 4.0132 | 3.1831 |
| 0.38 | 4.2362 | 3.2204 |
| 0.4 | 4.4591 | 3.249 |
| 0.42 | 4.6821 | 3.2687 |
| 0.44 | 4.905 | 3.2797 |
| 0.46 | 5.128 | 3.2817 |
| 0.48 | 5.351 | 3.275 |
| 0.5 | 5.5739 | 3.2594 |
| 0.52 | 5.7969 | 3.235 |
| 0.54 | 6.0198 | 3.2017 |




$$t = \frac{x}{u\cos\theta}$$

$$y = h + x\tan\theta - \frac{g}{2u^2}(1+\tan^2\theta)x^2$$

**Challenge #2: Create a more sophisticated exact ('analytic') model using equations for the projectile trajectory. In this case define a equally spaced *array* of x coordinate values between 0 and the maximum horizontal range $R$. Plot the trajectory and the apogee.**

**Projectile model**
Dr F. 14/7/23

**Target X,Y in m**

| | | |
|---|---|---|
| X | 1000 | input |
| Y | 300 | input |
| g /ms^-2 | 9.81 | input |

**minumim launch speed /ms^-1**
114.83

**u /ms^-1**
150 input

$$u \geq \sqrt{g}\sqrt{Y + \sqrt{X^2 + Y^2}}$$

**Discriminant**
1.698338

**High ball angle /radians**
1.33

**Low ball angle /radians**
0.54

**High ball angle /deg**
75.94

**Low ball angle /deg**
30.76

**Time of flight /s**
27.44

**Time of flight /s**
7.76

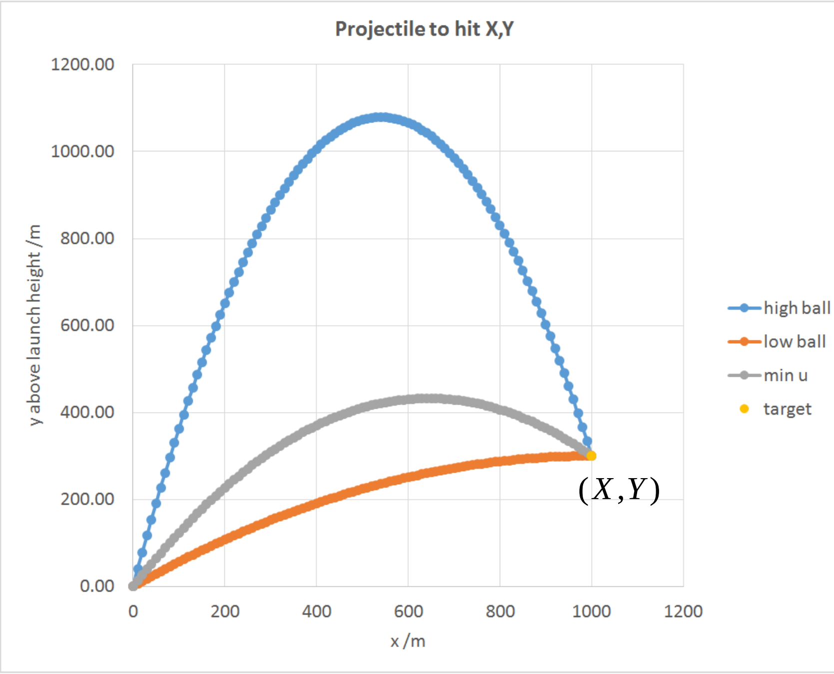
**Minimum speed angle /rad**
0.93

**Minimum speed angle /deg**
53.35

**Time of flight /s**
14.59

$$\theta = \tan^{-1}\left(\frac{Y + \sqrt{X^2 + Y^2}}{X}\right)$$

| f | x | y low ball | y high ball | y min u |
|---|---|---|---|---|
| 0 | 0 | 0.00 | 0.00 | 0.00 |
| 0.01 | 10 | 5.92 | 39.55 | 13.34 |
| 0.02 | 20 | 11.79 | 78.36 | 26.46 |
| 0.03 | 30 | 17.59 | 116.43 | 39.38 |
| 0.04 | 40 | 23.34 | 153.77 | 52.09 |
| 0.05 | 50 | 29.02 | 190.37 | 64.59 |
| 0.06 | 60 | 34.65 | 226.22 | 76.88 |
| 0.07 | 70 | 40.22 | 261.34 | 88.97 |
| 0.08 | 80 | 45.73 | 295.73 | 100.84 |
| 0.09 | 90 | 51.18 | 329.37 | 112.51 |
| [illegible] | [illegible] | 56.57 | 362.27 | 123.96 |
| [illegible] | [illegible] | 61.90 | 394.44 | 135.21 |
| [illegible] | [illegible] | 67.18 | 425.87 | 146.25 |
| [illegible] | [illegible] | 72.39 | 456.56 | 157.08 |
| 0.14 | 140 | 77.55 | 486.51 | 167.70 |
| 0.15 | 150 | 82.64 | 515.72 | 178.11 |
| 0.16 | 160 | 87.68 | 544.19 | 188.32 |
| 0.17 | 170 | 92.66 | 571.93 | 198.31 |
| 0.18 | 180 | 97.58 | 598.93 | 208.10 |
| 0.19 | 190 | 102.44 | 625.19 | 217.68 |
| 0.2 | 200 | 107.24 | 650.71 | 227.04 |
| 0.21 | 210 | 111.98 | 675.49 | 236.20 |
| 0.22 | 220 | 116.66 | 699.53 | 245.16 |
| 0.23 | 230 | 121.29 | 722.84 | 253.90 |
| 0.24 | 240 | 125.85 | 745.41 | 262.43 |
| 0.25 | 250 | 130.36 | 767.23 | 270.76 |
| 0.26 | 260 | 134.80 | 788.32 | 278.87 |
| 0.27 | 270 | 139.19 | 808.68 | 286.78 |
| 0.28 | 280 | 143.52 | [illegible] | [illegible] |
| [illegible] | [illegible] | 147.79 | [illegible] | [illegible] |
| [illegible] | [illegible] | 152.00 | [illegible] | [illegible] |
| [illegible] | [illegible] | 156.15 | [illegible] | [illegible] |
| [illegible] | [illegible] | 160.24 | [illegible] | [illegible] |
| [illegible] | [illegible] | 164.28 | [illegible] | [illegible] |
| [illegible] | [illegible] | 168.25 | [illegible] | [illegible] |
| 0.35 | 350 | 172.17 | [illegible] | [illegible] |




$$Y = h + X\tan\theta - \frac{g}{2u^2}(1+\tan^2\theta)X^2$$

$$\therefore a\tan^2\theta + b\tan\theta + c = 0 \qquad a = \frac{g}{2u^2}X^2 \qquad b = -X \qquad c = Y - h + \frac{gX^2}{2u^2}$$

$$\theta_{\pm} = \tan^{-1}\left(\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}\right)$$

Note in this example I have set $h = 0$. But you don't have to!

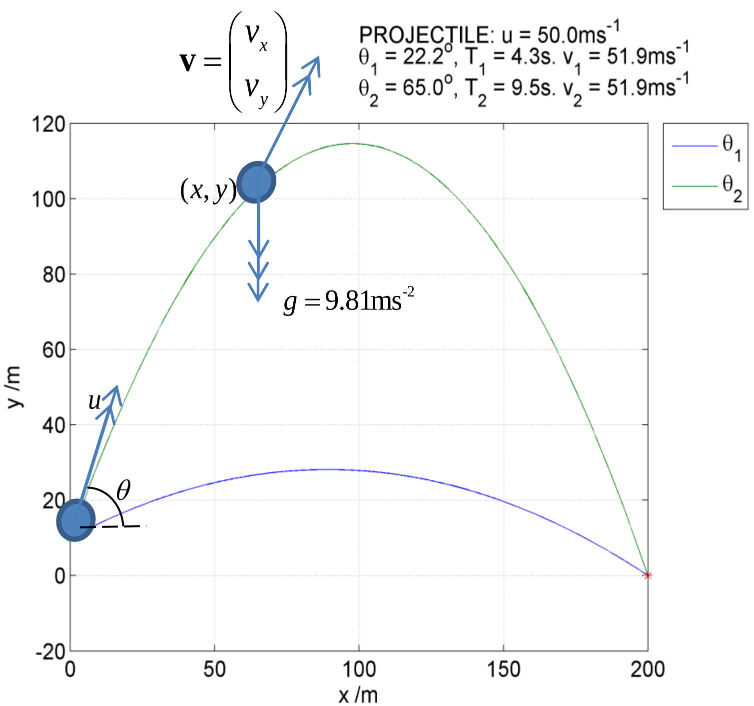
**Challenge #3: Create a new projectile model which is based upon calculating trajectories that are launched from $(0,0)$ and pass through a fixed position $(X,Y)$. Calculate the minimum launch speed to achieve this, and hence determine 'low ball' and 'high ball' trajectories. Derivations of the associated mathematics are on the next few slides.**

**Projectiles** are typically modelled as point masses (i.e. 'particles') falling under gravity. In other words, internal motion and rotation is ignored and only the centre of mass of the projectile is considered. *Air resistance is often ignored* to enable analysis to proceed without a computer. Note this assumption may be significantly invalid for many real projectiles! Hence this system reduces to a *two dimensional kinematics problem, where acceleration is constant.*

Let the coordinates of the projectile be $(x,y)$ on a Cartesian grid. Let the initial velocity be $u$ at an elevation of $\theta$ and let the projectile be launched from $(0,h)$
Since acceleration is constant:

$$\boxed{\begin{aligned} v_x &= u\cos\theta \\ v_y &= u\sin\theta - gt \\ v_y^2 &= u^2\sin^2\theta - 2g(y-h) \\ x &= ut\cos\theta \\ y &= h + ut\sin\theta - \tfrac{1}{2}gt^2 \end{aligned}}$$

Note this means the $x$ direction velocity is *always constant* throughout the motion!




We can therefore combine these equations to find various properties of the projectile's trajectory

$$x = ut\cos\theta$$

$$\therefore \boxed{t = \frac{x}{u\cos\theta}}$$

$$\frac{1}{\cos^2\theta} = 1 + \tan^2\theta$$

$$\therefore \boxed{y = h + x\tan\theta - \frac{g}{2u^2}(1+\tan^2\theta)x^2}$$

i.e. a projectile trajectory is an *inverted parabola*

If the projectile is required to pass through (or collide with!) a particular coordinate $(X,Y)$, we can solve the quadratic trajectory equation to determine the elevation angle, given speed $u$ is known. This calculation relates to models of all ball sports, gunnery (ballistics) etc.

$$Y = h + X\tan\theta - \frac{g}{2u^2}(1+\tan^2\theta)X^2$$

$$a\tan^2\theta + b\tan\theta + c = 0$$

$$\boxed{\begin{aligned} a &= \frac{g}{2u^2}X^2 \\ b &= -X \\ c &= Y - h + \frac{gX^2}{2u^2} \\ \theta &= \tan^{-1}\left(\frac{-b \pm \sqrt{b^2-4ac}}{2a}\right) \end{aligned}}$$

Note *multiple solutions* are possible, depending on the sign of the *discriminant* $b^2 - 4ac$

Elevation angles which give rise to a *zero discriminant* define the *bounding parabola* for the projectile (see next page).

The *apogee* of the trajectory is when $v_y = 0$

$$v_y = u\sin\theta - gt \quad \therefore \; v_y = 0 \Rightarrow \boxed{t_a = \frac{u}{g}\sin\theta}$$

$$v_y^2 = u^2\sin^2\theta - 2g(y-h) \quad \therefore \; v_y = 0 \Rightarrow \boxed{y_a = h + \frac{u^2}{2g}\sin^2\theta}$$

$$x_a = ut_a\cos\theta \quad \therefore \boxed{x_a = \frac{u^2}{g}\sin\theta\cos\theta}$$

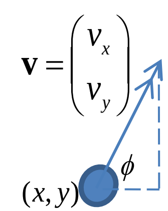
The speed $v$ of the projectile is:

$$v = \sqrt{v_x^2 + v_y^2}$$

$$v = \sqrt{u^2\cos^2\theta + u^2\sin^2\theta - 2g(y-y_0)}$$

$$\boxed{v = \sqrt{u^2 - 2g(y-y_0)}}$$

Compute angle of velocity using: $\boxed{\phi = \tan^{-1}\left(\frac{v_y}{v_x}\right)}$



## Possible values for $u$ and the bounding parabola

$$\boxed{Y = h + X\tan\theta - \frac{g}{2u^2}(1+\tan^2\theta)X^2}$$

**Trajectory equation**

$$a\tan^2\theta + b\tan\theta + c = 0$$

$$a = \frac{g}{2u^2}X^2$$

$$b = -X$$

$$c = Y - h + \frac{gX^2}{2u^2}$$

$$\boxed{\theta = \tan^{-1}\left(\frac{-b \pm \sqrt{b^2-4ac}}{2a}\right)}$$

For real values of $\theta$: $b^2 - 4ac \geq 0$

Without loss of generality, set a coordinate system such that $h = 0$ i.e. vary the target coordinates $X,Y$ instead, by shifting the origin

$$X^2 - 4\left(-\frac{gX^2}{2u^2}\right)\left(-Y - \frac{g}{2u^2}X^2\right) \geq 0$$

$$2u^4X^2 - 2gX^2\left(2Yu^2 + gX^2\right) \geq 0$$

$$u^4 - 2Ygu^2 - g^2X^2 \geq 0$$

$$\left(u^2 - Yg\right)^2 - Y^2g^2 - g^2X^2 \geq 0$$

$$u^2 \geq Yg + g\sqrt{X^2+Y^2}$$

$$u^2 \leq Yg - g\sqrt{X^2+Y^2}$$

Non physical, since $u$ is real and positive

$$\therefore \boxed{u \geq \sqrt{g}\sqrt{Y+\sqrt{X^2+Y^2}}}$$

The **minimum $u$ parabola** is defined by the trajectory corresponding to the minimum velocity required to generate a projectile trajectory which intersects with $(X,Y)$.

$$\boxed{u^2 = g\left(Y + \sqrt{X^2+Y^2}\right)}$$

$$y = x\tan\theta - \frac{g}{2u^2}(1+\tan^2\theta)x^2$$

**Trajectory equation for minimum $u$ parabola**

$$a\tan^2\theta + b\tan\theta + c = 0$$

$$a = \frac{g}{2u^2}X^2, \quad b = -X, \quad c = Y + \frac{g}{2u^2}X^2$$

$$b^2 - 4ac = 0$$

$$\therefore \theta = \tan^{-1}\left(\frac{-b}{2a}\right)$$

$$\theta = \tan^{-1}\left(\frac{X}{\frac{g}{u^2}X^2}\right)$$

$$\theta = \tan^{-1}\left(\frac{u^2}{gX}\right)$$

$$\boxed{\theta = \tan^{-1}\left(\frac{Y+\sqrt{X^2+Y^2}}{X}\right)}$$

**minimum $u$ parabola elevation angle**

$$\therefore \tan\theta = \frac{Y+\sqrt{X^2+Y^2}}{X}$$

$$y = x\tan\theta - \frac{g}{2u^2}(1+\tan^2\theta)x^2$$

$$y = x\left(\frac{Y+\sqrt{X^2+Y^2}}{X}\right) - \frac{g}{2g\left(Y+\sqrt{X^2+Y^2}\right)}\left(1 + \frac{\left(Y+\sqrt{X^2+Y^2}\right)^2}{X^2}\right)x^2$$

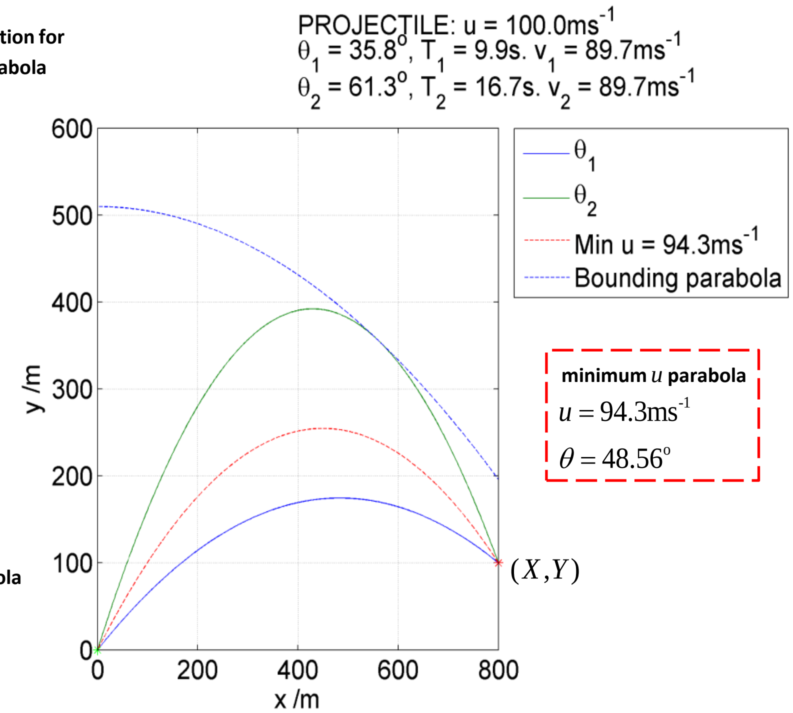
$$y = x\left(\frac{Y+\sqrt{X^2+Y^2}}{X}\right) - \frac{g}{2g\left(Y+\sqrt{X^2+Y^2}\right)}\left(\frac{X^2+Y^2+2Y\sqrt{X^2+Y^2}+X^2+Y^2}{X^2}\right)x^2$$

$$y = x\left(\frac{Y+\sqrt{X^2+Y^2}}{X}\right) - \frac{1}{Y+\sqrt{X^2+Y^2}}\left(\frac{X^2+Y^2+Y\sqrt{X^2+Y^2}}{X^2}\right)x^2$$

$$y = x\left(\frac{Y+\sqrt{X^2+Y^2}}{X}\right) - \frac{\sqrt{X^2+Y^2}}{Y+\sqrt{X^2+Y^2}}\left(\frac{\sqrt{X^2+Y^2}+Y}{X^2}\right)x^2$$

$$\boxed{y = x\left(\frac{Y+\sqrt{X^2+Y^2}}{X}\right) - \frac{\sqrt{X^2+Y^2}}{X^2}x^2}$$

**minimum $u$ parabola. Only *one* value of $\theta$ is possible, since the trajectory equation discriminant is zero.**




**minimum $u$ parabola**

$$u = 94.3\text{ms}^{-1}$$

$$\theta = 48.56^\circ$$

The **bounding parabola** is slightly different – this bounds the possible set of trajectories *given* a value of $u$

$$y = x\tan\theta - \frac{g}{2u^2}(1+\tan^2\theta)x^2$$

$$2u^2y = 2u^2x\tan\theta - gx^2 - gx^2\tan^2\theta$$

$$gx^2\tan^2\theta - 2u^2x\tan\theta + 2u^2y + gx^2 = 0$$

For positive discriminant:

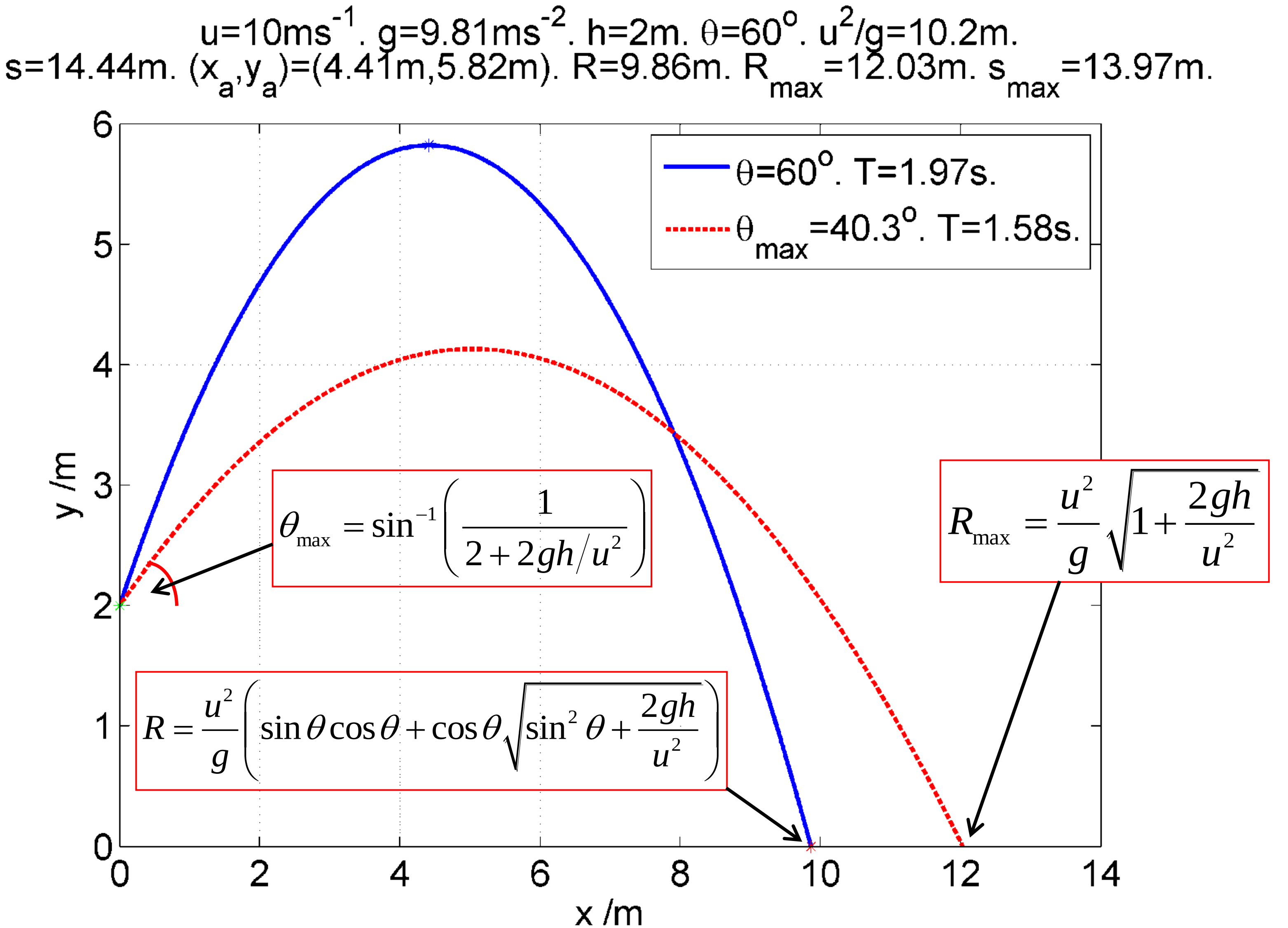
$$4u^4x^2 - 4gx^2\left(2u^2y + gx^2\right) \geq 0$$

$$\frac{u^4}{g} \geq 2u^2y + gx^2$$

$$y \leq \frac{u^2}{2g} - \frac{g}{2u^2}x^2$$

**Bounding parabola**

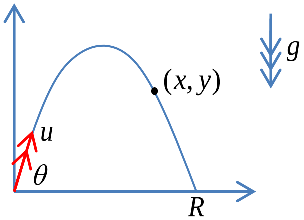
$$\boxed{y = \frac{u^2}{2g} - \frac{g}{2u^2}x^2}$$




**Challenge #4: Create a new projectile model which compares a trajectory to the *trajectory which maximizes horizontal range given the same launch height and launch speed*. Inputs are $u$,$h$,$g$ and $\theta$. For the maximum range trajectory you need to calculate the optimum angle. For $h > 0$ note this is not 45°... Derivation in the next few slides.**

Note there is a nicer way of doing this!

## The maximum range problem




Given a fixed projectile launch speed what angle maximises range?

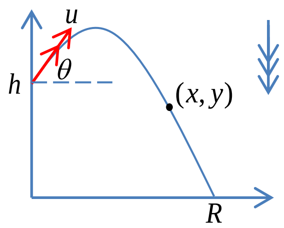
$$x = ut\cos\theta$$
$$y = ut\sin\theta - \tfrac{1}{2}gt^2$$

$$x = R, y = 0$$
$$\therefore 0 = t\left(u\sin\theta - \tfrac{1}{2}gt\right)$$
$$t > 0 \Rightarrow u\sin\theta - \tfrac{1}{2}gt = 0$$
$$\therefore t = \frac{2u\sin\theta}{g}$$
$$R = ut\cos\theta$$
$$\therefore R = \frac{2u^2}{g}\sin\theta\cos\theta$$
$$\boxed{R = \frac{u^2}{g}\sin 2\theta}$$

Hence maximum range is:

$$\boxed{R_{\max} = \frac{u^2}{g}, \quad \theta = 45^\circ}$$

Let us now extend the problem to a starting height which is *not* at ground level.




$$x = ut\cos\theta$$
$$y = ut\sin\theta - \tfrac{1}{2}gt^2 + h$$

$$x = R, y = 0$$
$$\therefore 0 = ut\sin\theta - \tfrac{1}{2}gt^2 + h$$
$$t^2 - \frac{2ut}{g}\sin\theta - \frac{2h}{g} = 0$$
$$\left(t - \frac{u\sin\theta}{g}\right)^2 - \frac{u^2\sin^2\theta}{g^2} - \frac{2gh}{g^2} = 0$$
$$t = \frac{u\sin\theta}{g} + \frac{u}{g}\sqrt{\sin^2\theta + \frac{2gh}{u^2}}$$

positive root since $t > 0$

$$R = ut\cos\theta$$
$$\boxed{R = \frac{u^2}{g}\left(\sin\theta\cos\theta + \cos\theta\sqrt{\sin^2\theta + \frac{2gh}{u^2}}\right)}$$

To maximize $R$ we need to find $\theta$ such that:

$$\frac{d}{d\theta}\left(\frac{Rg}{u^2}\right) = 0$$

For brevity define $\boxed{\alpha = \frac{2gh}{u^2}}$

Note: $\alpha = \frac{2gh}{u^2} = \frac{mgh}{\frac{1}{2}mu^2} = \frac{\text{GPE}}{\text{KE}}$

$$\frac{d}{d\theta}\left(\sin\theta\cos\theta + \cos\theta\sqrt{\sin^2\theta + \alpha}\right) = 0$$
$$\sin\theta(-\sin\theta) + \cos\theta(\cos\theta) + \frac{\frac{1}{2}\cos\theta}{\sqrt{\sin^2\theta + \alpha}}(2\sin\theta\cos\theta) - \sin\theta\sqrt{\sin^2\theta + \alpha} = 0$$
$$-\sin^2\theta + \cos^2\theta = \sin\theta\sqrt{\sin^2\theta + \alpha} - \frac{\sin\theta\cos^2\theta}{\sqrt{\sin^2\theta + \alpha}}$$
$$\sqrt{\sin^2\theta + \alpha}\left(1 - 2\sin^2\theta\right) = \sin\theta\left(\sin^2\theta + \alpha\right) - \sin\theta\left(1 - \sin^2\theta\right) \qquad \cos^2\theta = 1 - \sin^2\theta$$
$$\sqrt{\sin^2\theta + \alpha} = \frac{2\sin^3\theta + (\alpha - 1)\sin\theta}{1 - 2\sin^2\theta}$$
$$\sin^2\theta + \alpha = \frac{4\sin^6\theta + 4(\alpha - 1)\sin^4\theta + (\alpha - 1)^2\sin^2\theta}{1 - 4\sin^2\theta + 4\sin^4\theta}$$
$$\left(1 - 4\sin^2\theta + 4\sin^4\theta\right)\left(\sin^2\theta + \alpha\right) = 4\sin^6\theta + 4(\alpha - 1)\sin^4\theta + (\alpha - 1)^2\sin^2\theta$$
$$\sin^2\theta + \alpha - 4\sin^4\theta - 4\alpha\sin^2\theta + 4\sin^6\theta + 4\alpha\sin^4\theta =$$
$$4\sin^6\theta + 4\alpha\sin^4\theta - 4\sin^4\theta + (\alpha - 1)^2\sin^2\theta$$

$$\alpha + (1 - 4\alpha)\sin^2\theta = (\alpha - 1)^2\sin^2\theta$$
$$\alpha = \left(\alpha^2 - 2\alpha + 1 - 1 + 4\alpha\right)\sin^2\theta$$
$$\alpha = \left(\alpha^2 + 2\alpha\right)\sin^2\theta$$
$$\frac{1}{\alpha + 2} = \sin^2\theta$$
$$\boxed{\sin\theta = \frac{1}{\sqrt{2 + \alpha}}}$$
$$\cos\theta = \sqrt{1 - \tfrac{1}{2+\alpha}}$$
$$\boxed{\cos\theta = \sqrt{\frac{1 + \alpha}{2 + \alpha}}}$$

The range-maximizing angle is therefore:

$$\boxed{\theta = \sin^{-1}\left(\frac{1}{\sqrt{2 + \alpha}}\right)}$$

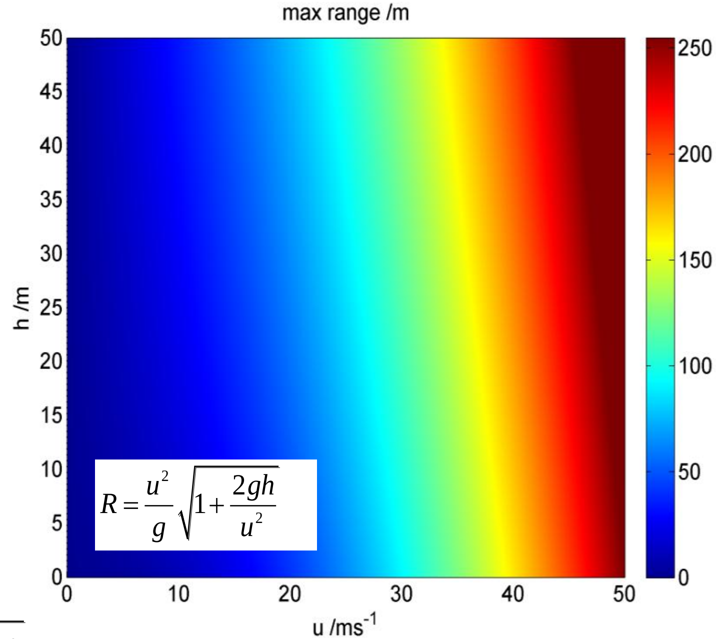
$$\frac{Rg}{u^2} = \sin\theta\cos\theta + \cos\theta\sqrt{\sin^2\theta + \alpha}$$
$$\frac{Rg}{u^2} = \frac{1}{\sqrt{2 + \alpha}}\sqrt{\frac{1 + \alpha}{2 + \alpha}} + \sqrt{\frac{1 + \alpha}{2 + \alpha}}\sqrt{\frac{1}{2 + \alpha} + \alpha}$$
$$\frac{Rg}{u^2} = \frac{\sqrt{1 + \alpha}}{2 + \alpha} + \sqrt{\frac{1 + \alpha}{2 + \alpha}}\sqrt{\frac{1 + 2\alpha + \alpha^2}{2 + \alpha}}$$
$$\frac{Rg}{u^2} = \frac{\sqrt{1 + \alpha}}{2 + \alpha} + \frac{\sqrt{1 + \alpha}}{2 + \alpha}\sqrt{(1 + \alpha)^2}$$
$$\frac{Rg}{u^2} = \frac{\sqrt{1 + \alpha}}{2 + \alpha}(1 + 1 + \alpha) = \frac{\sqrt{1 + \alpha}}{2 + \alpha}(2 + \alpha)$$
$$\frac{Rg}{u^2} = \sqrt{1 + \alpha}$$

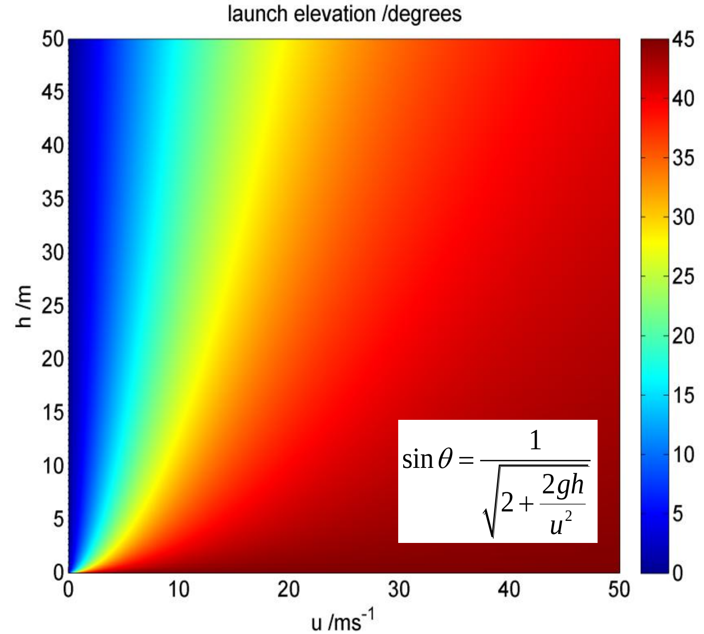
$$\boxed{R = \frac{u^2}{g}\sqrt{1 + \frac{2gh}{u^2}}, \quad \sin\theta = \frac{1}{\sqrt{2 + \frac{2gh}{u^2}}}}$$

Given the maximum range problem involves *two parameters*, to visualize possible solutions we need to plot a *surface* graph.

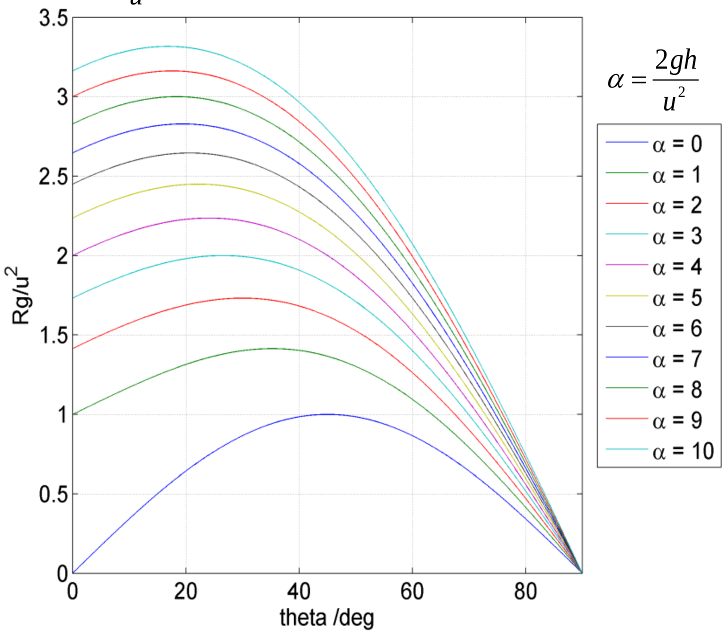
In the example plots, *colour* is used to indicate the height of the surface.

In all examples $g = 9.81\text{ms}^{-2}$



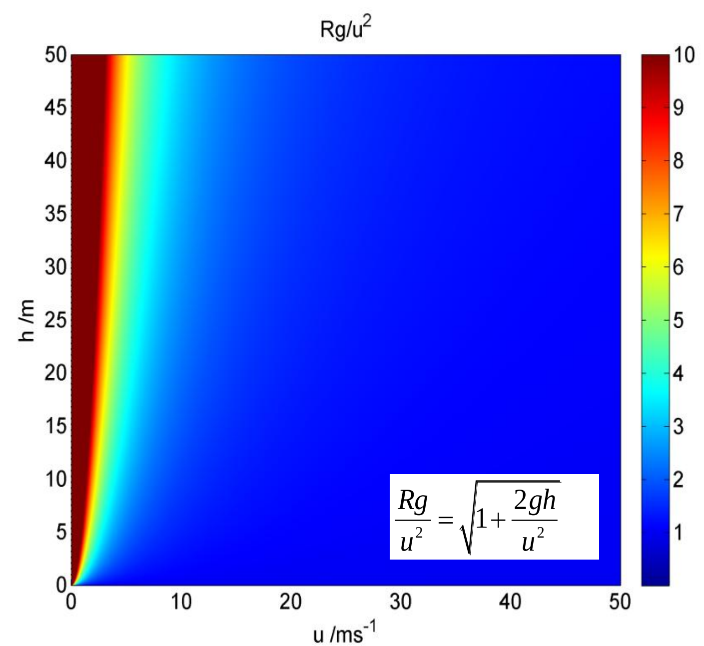

$$\frac{Rg}{u^2} = \sin\theta\cos\theta + \cos\theta\sqrt{\sin^2\theta + \alpha}$$




This graph demonstrates that range has a maximum value as the launch elevation is varied.

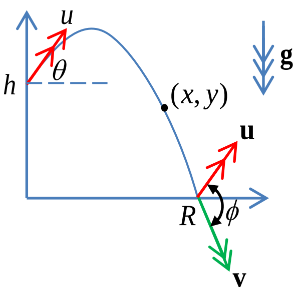
The angle which results in the maximum range is given by

$$\sin\theta = \frac{1}{\sqrt{2+\alpha}}$$



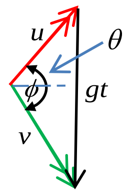
## An elegant solution to the maximum range problem

There is an alternative, more *geometric*, method that arrives at the solution to the maximum range problem without so much trigonometric horror!




The velocity at maximum range $R$ is given by the vector equation:

$$\boxed{\mathbf{v}=\mathbf{u}+\mathbf{g}t}$$




The area $A$ of the vector triangle can be computed in *two* different ways:

$$A=\tfrac{1}{2}uv\sin\phi$$
$$A=\tfrac{1}{2}gt\times u\cos\theta$$
$$\therefore uv\sin\phi=gut\cos\theta$$

Since the projectile moves at constant speed horizontally: $\boxed{R=ut\cos\theta}$

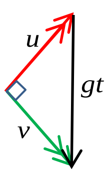
By conservation of energy: $mgh+\tfrac{1}{2}mu^2=\tfrac{1}{2}mv^2 \quad \therefore \boxed{v=\sqrt{2gh+u^2}}$

Hence: $uv\sin\phi=gut\cos\theta \;\Rightarrow \dfrac{u}{g}\sin\phi\sqrt{2gh+u^2}=R$

$$\therefore R=\frac{u^2}{g}\sqrt{1+\frac{2gh}{u^2}}\sin\phi$$

The largest $R$ possible corresponds to $\sin\phi=1\Rightarrow\boxed{\phi=90^\circ}$

$$\boxed{R=\frac{u^2}{g}\sqrt{1+\frac{2gh}{u^2}}}$$




At maximum range the velocity triangle is *right angled*, so using Pythagoras' theorem we can calculate the time of flight corresponding to the maximum range

$$g^2t^2=u^2+v^2 \quad \therefore g^2t^2=u^2+2gh+u^2$$

$$\therefore \boxed{t=\frac{u}{g}\sqrt{2+\frac{2gh}{u^2}}}$$

We can use this result, combined with the expression for $R$, to find the required elevation angle to result in maximum range.

$$R=ut\cos\theta$$

$$\therefore \frac{u^2}{g}\sqrt{1+\frac{2gh}{u^2}}=u\frac{u}{g}\sqrt{2+\frac{2gh}{u^2}}\cos\theta$$

$$\therefore \cos\theta=\frac{\sqrt{1+\frac{2gh}{u^2}}}{\sqrt{2+\frac{2gh}{u^2}}}$$

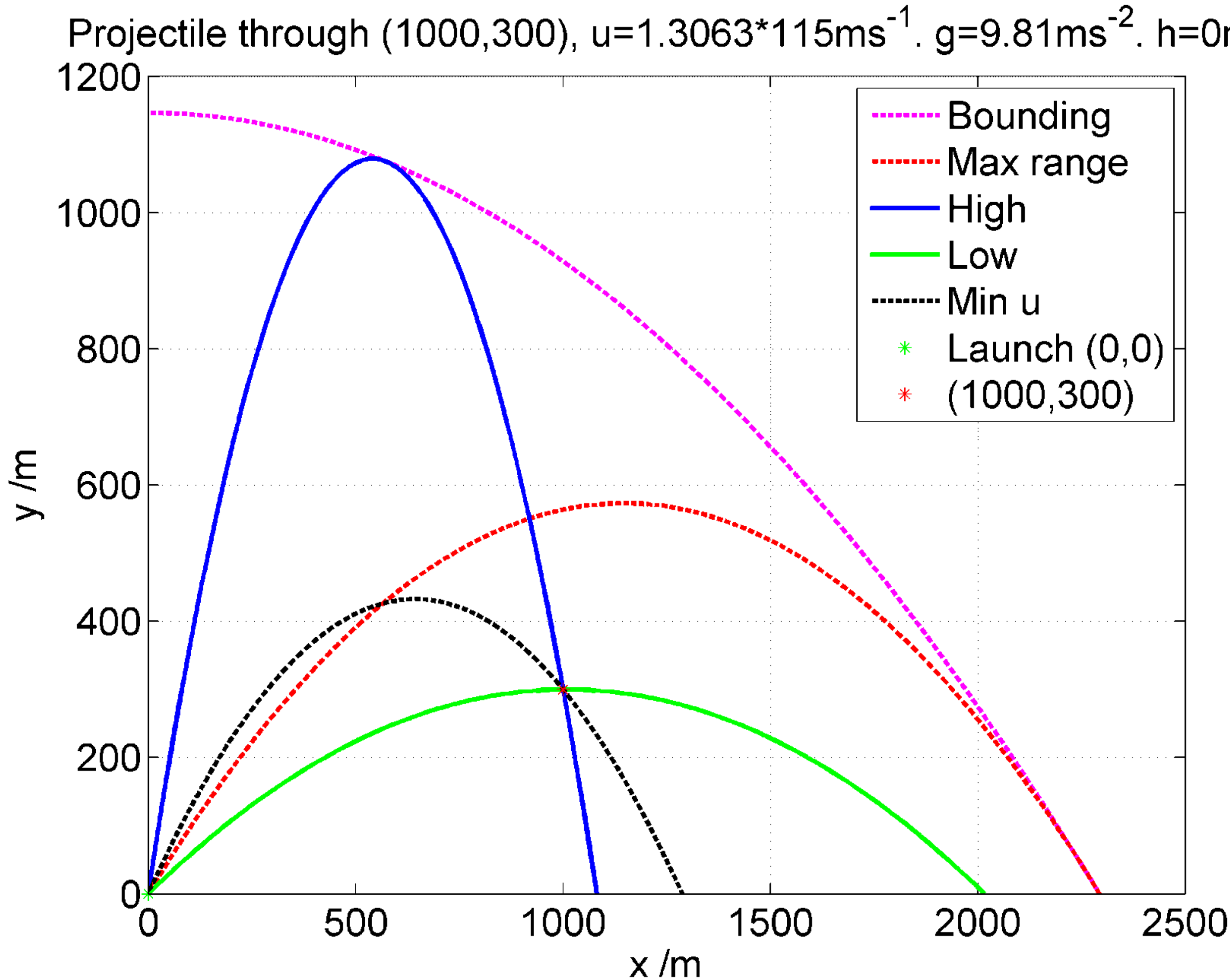
$$\sin^2\theta=1-\cos^2\theta$$

$$\therefore \sin^2\theta=1-\frac{1+\frac{2gh}{u^2}}{2+\frac{2gh}{u^2}}$$

$$\therefore \sin^2\theta=\frac{2+\frac{2gh}{u^2}-1-\frac{2gh}{u^2}}{2+\frac{2gh}{u^2}}$$

$$\therefore \sin^2\theta=\frac{1}{2+\frac{2gh}{u^2}}$$

$$\therefore \boxed{\theta=\sin^{-1}\left(\frac{1}{2+2gh/u^2}\right)}$$



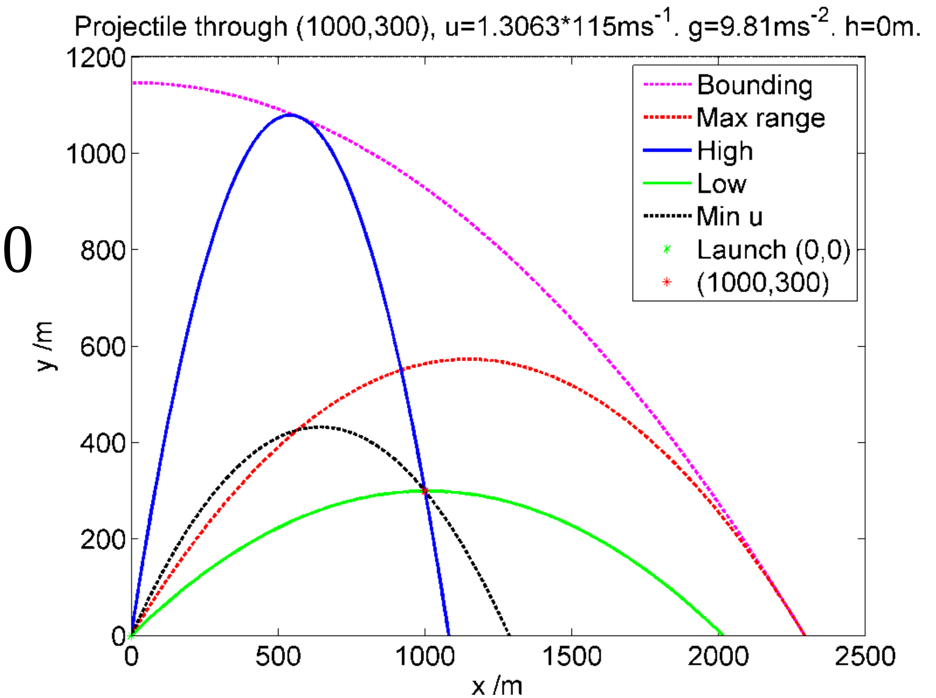

**Challenge #5: Update your projectile model of a trajectory which passes through $(X,Y)$ with the *bounding parabola*, in addition to minimum speed, max range and high and low ball curves. The bounding parabola marks the region where possible $(X,Y)$ coordinates could be reached given $u,h,g$ inputs.**

The **bounding parabola** sets the limit of the possible set of trajectories *given* a value of $u$

$$y = x\tan\theta - \frac{g}{2u^2}(1+\tan^2\theta)x^2$$

$$2u^2 y = 2u^2 x\tan\theta - gx^2 - gx^2\tan^2\theta$$

$$gx^2\tan^2\theta - 2u^2 x\tan\theta + 2u^2 y + gx^2 = 0$$




For positive discriminant of this quadratic:

$$4u^4x^2 - 4gx^2\left(2u^2y + gx^2\right) \geq 0$$

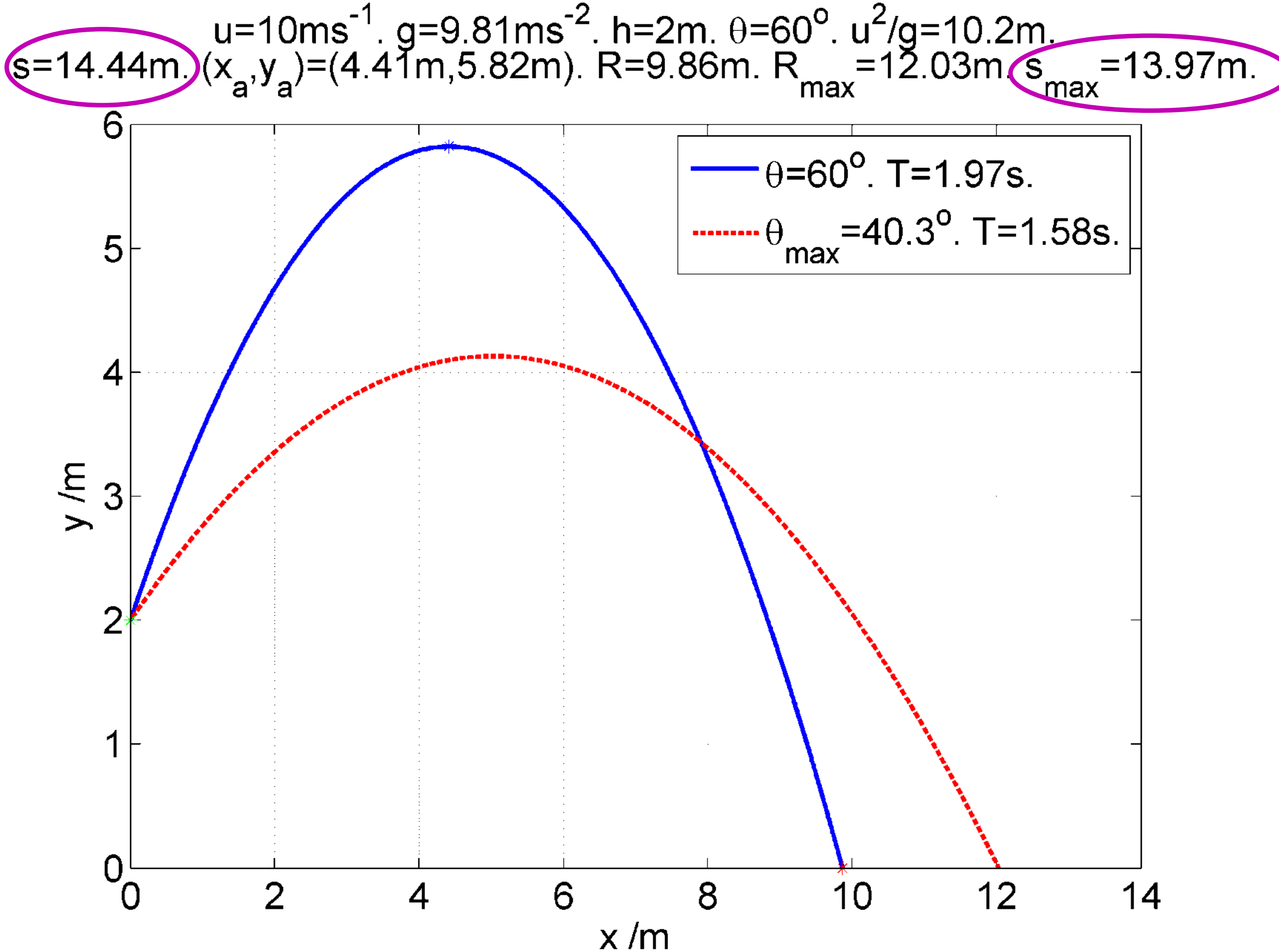
$$\frac{u^4}{g} \geq 2u^2y + gx^2$$

$$y \leq \frac{u^2}{2g} - \frac{g}{2u^2}x^2$$

**Bounding parabola**

$$\boxed{y = \frac{u^2}{2g} - \frac{g}{2u^2}x^2}$$

Shift $y$ coordinates by $h$ if launching a projectile from $(0,h)$



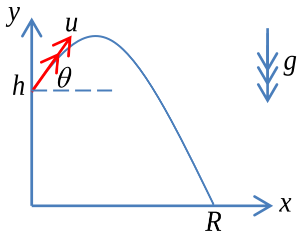

**Challenge #6: Now update your projectile model with a calculation of the *distance travelled by the projectile* i.e. the length of the inverted parabolic arc. The calculus for this is on the next slide, and example MATLAB code follows.**

**Projectile distance travelled**

The distance travelled by a particle undergoing projectile motion from $(0,h)$ is given by:

$$s = \int_0^X \sqrt{(dx)^2 + (dy)^2}$$

$$s = \int_0^X \sqrt{1 + \left(\frac{dy}{dx}\right)^2}\, dx$$




Now trajectory equation is:

$$\boxed{y = h + x\tan\theta - \frac{g}{2u^2}(1+\tan^2\theta)x^2}$$

$$R = \frac{u^2}{g}\left(\sin\theta\cos\theta + \cos\theta\sqrt{\sin^2\theta + \frac{2gh}{u^2}}\right)$$

$$\therefore \frac{dy}{dx} = \tan\theta - \frac{gx}{u^2}(1+\tan^2\theta)$$

$$\therefore s = \int_0^X \sqrt{1 + \left(\tan\theta - \frac{gx}{u^2}(1+\tan^2\theta)\right)^2}\, dx$$

Consider a substitution:

$$z = \tan\theta - \frac{gx}{u^2}(1+\tan^2\theta) \quad \therefore dz = -\frac{g}{u^2}(1+\tan^2\theta)dx$$

$$\therefore s = -\frac{u^2}{g(1+\tan^2\theta)}\int_{\tan\theta}^{\tan\theta - \frac{gX}{u^2}(1+\tan^2\theta)} \sqrt{1+z^2}\,dz$$

Note standard integral:

$$\int \sqrt{1+z^2}\,dz = \tfrac{1}{2}\ln\left|\sqrt{1+z^2} + x\right| + \tfrac{1}{2}z\sqrt{1+z^2} + c$$

$$\boxed{\therefore s = \frac{u^2}{g(1+\tan^2\theta)}\left[\tfrac{1}{2}\ln\left|\sqrt{1+z^2} + z\right| + \tfrac{1}{2}z\sqrt{1+z^2}\right]_{\tan\theta - \frac{gX}{u^2}(1+\tan^2\theta)}^{\tan\theta}}$$

Which can be calculated easily using MATLAB/Python/Excel etc, and checked with a numeric approximate calculation using a small discrete value of $\Delta x$.

Consider a **special case when projectile is launched from the origin** (i.e. $h = 0$), and $X = R = \frac{2u^2}{g}\sin\theta\cos\theta$ i.e. when the inverted parabolic trajectory crosses the horizontal axis after launch.

$$\therefore \tan\theta - \frac{gX}{u^2}(1+\tan^2\theta) = \tan\theta - 2\sin\theta\cos\theta(1+\tan^2\theta)$$

$$= \tan\theta - \frac{2\sin\theta\cos\theta}{\cos^2\theta} = -\tan\theta$$

$$\therefore s = \frac{u^2}{g(1+\tan^2\theta)}\left[\tfrac{1}{2}\ln\left|\sqrt{1+z^2} + z\right| + \tfrac{1}{2}z\sqrt{1+z^2}\right]_{-\tan\theta}^{\tan\theta}$$

$$= \tfrac{1}{2}\frac{u^2}{g(1+\tan^2\theta)}\left(\ln\left|\sqrt{1+\tan^2\theta} + \tan\theta\right| + \tan\theta\sqrt{1+\tan^2\theta} - \ln\left|\sqrt{1+\tan^2\theta} - \tan\theta\right| + \tan\theta\sqrt{1+\tan^2\theta}\right)$$

$$= \frac{u^2}{g(1+\tan^2\theta)}\left(\tfrac{1}{2}\ln\left|\frac{\sqrt{1+\tan^2\theta} + \tan\theta}{\sqrt{1+\tan^2\theta} - \tan\theta}\right| + \tan\theta\sqrt{1+\tan^2\theta}\right)$$

$$= \frac{u^2\cos^2\theta}{g}\left(\tfrac{1}{2}\ln\left|\frac{\frac{1}{\cos\theta} + \frac{\sin\theta}{\cos\theta}}{\frac{1}{\cos\theta} - \frac{\sin\theta}{\cos\theta}}\right| + \frac{\sin\theta}{\cos\theta}\frac{1}{\cos\theta}\right) = \frac{u^2\cos^2\theta}{g}\left(\tfrac{1}{2}\ln\left|\frac{1+\sin\theta}{1-\sin\theta}\right| + \frac{\sin\theta}{\cos^2\theta}\right)$$

$$\frac{1+\sin\theta}{1-\sin\theta} = \frac{(1+\sin\theta)^2}{1-\sin^2\theta} = \frac{(1+\sin\theta)^2}{\cos^2\theta}$$

$$\boxed{\therefore s = \frac{u^2}{g}\left(\ln\left(\frac{1+\sin\theta}{\cos\theta}\right)\cos^2\theta + \sin\theta\right)}$$

When $R$ is maximized: $\theta = \frac{\pi}{4}$, $\sin\theta = \cos\theta = \frac{1}{\sqrt{2}} = \frac{\sqrt{2}}{2}$, $R = \frac{2u^2}{g}$,

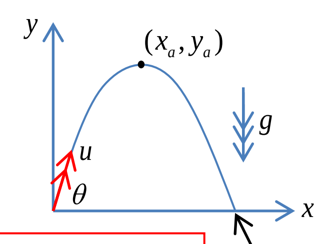
$$\therefore s = \frac{u^2}{g}\left(\ln\left|\frac{1+\frac{1}{\sqrt{2}}}{\frac{1}{\sqrt{2}}}\right| \times \tfrac{1}{2} + \tfrac{\sqrt{2}}{2}\right)$$

$$\therefore s = \tfrac{1}{2}\frac{u^2}{g}\left(\ln\left(1+\sqrt{2}\right) + \sqrt{2}\right)$$

$$\boxed{\therefore s \approx 1.15\frac{u^2}{g}}$$

$$\ln\left(1+\sqrt{2}\right) + \sqrt{2} \approx 2.296$$

*Universal parabola constant*




$$\boxed{x_a = \tfrac{1}{2}R, \quad y_a = \frac{u^2}{2g}\sin^2\theta}$$

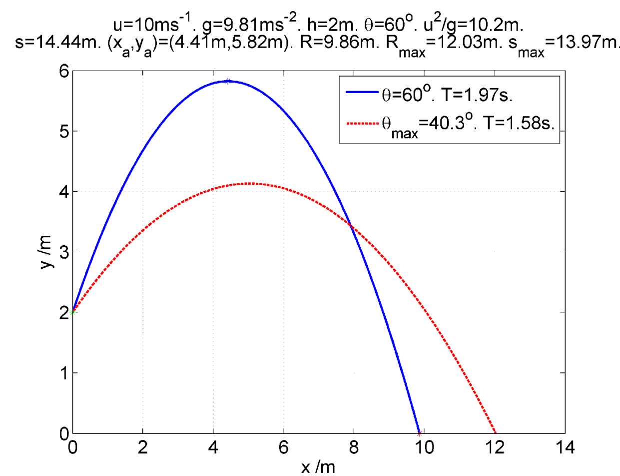
$$\boxed{s = \frac{u^2}{g}\left(\ln\left(\frac{1+\sin\theta}{\cos\theta}\right)\cos^2\theta + \sin\theta\right) \quad t = \frac{x}{u\cos\theta}, \quad T = \frac{R}{u\cos\theta}}$$

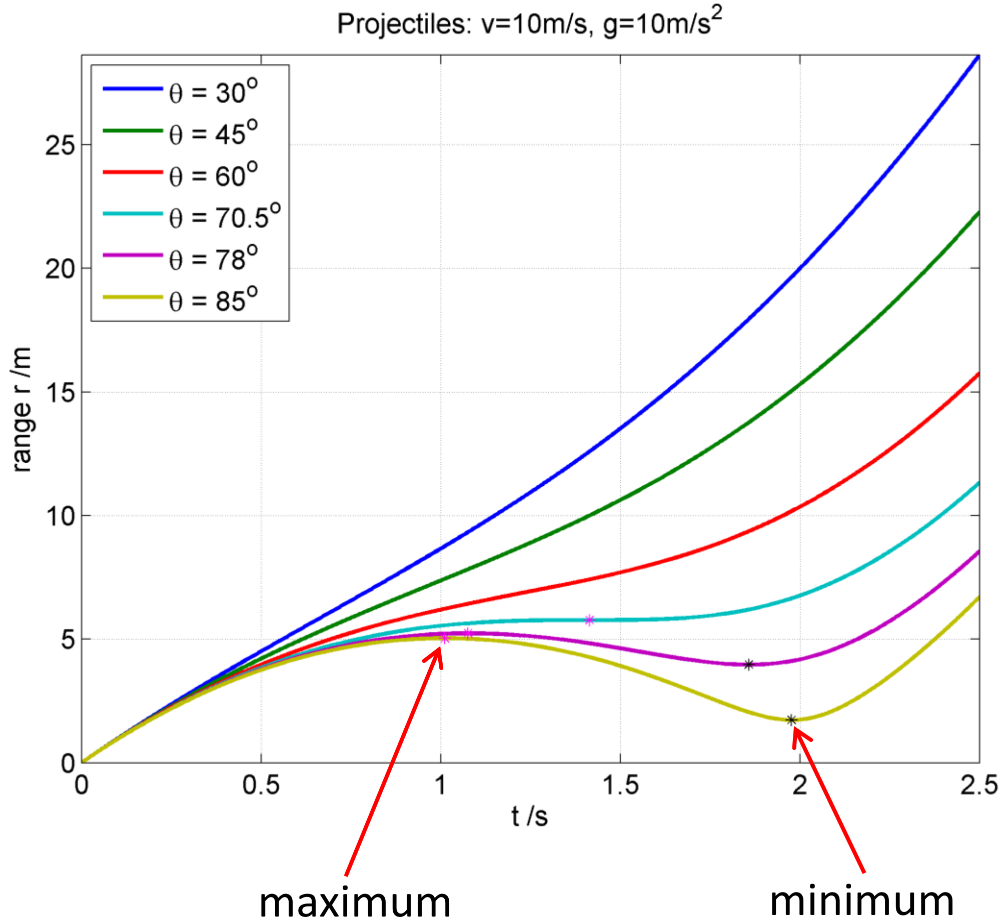
$$\boxed{y = x\tan\theta - \frac{g}{2u^2}(1+\tan^2\theta)x^2}$$

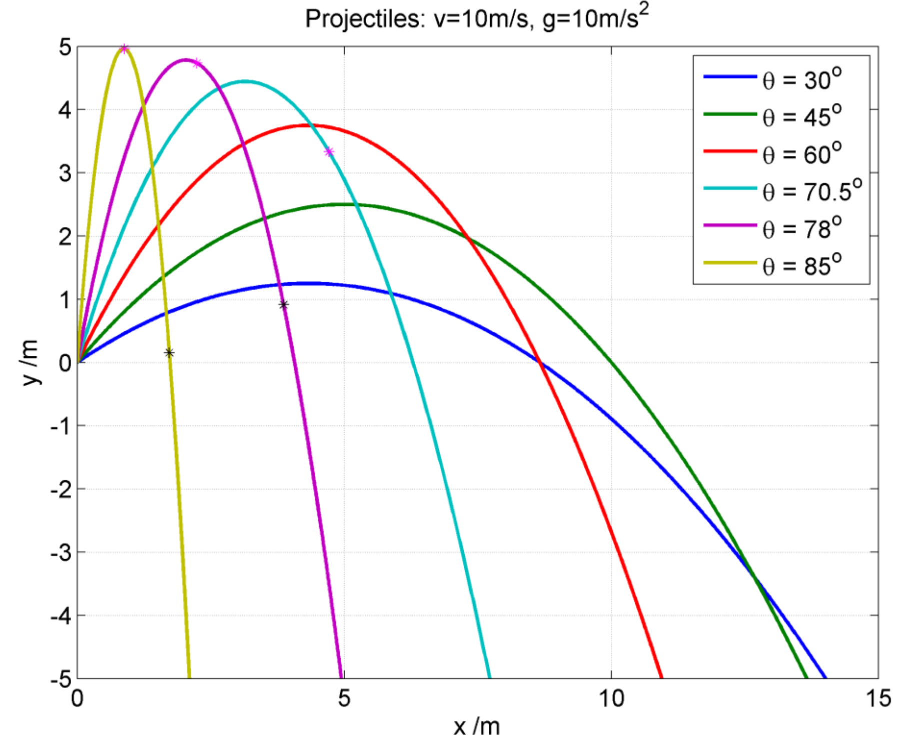
$$\boxed{R = \frac{u^2}{g}\sin 2\theta}$$

# MATLAB code to calculate a projectile trajectory

```
%Projectile trajectory calculator (no air resistance)
function p = pcalc( theta, u, g, h, N )

%Range /m
p.R = ((u^2)/g)*( sin(theta)*cos(theta) +...
   cos(theta)*sqrt( sin(theta)^2 + 2*g*h/(u^2) ) );

%x /m
p.x = linspace(0,p.R,N);

%t /s
p.t = p.x/(u*cos(theta));

%Time of flight /s
p.T = p.R/(u*cos(theta));

%y /m
p.y = h + p.x*tan(theta) - ( g/(2*u^2) )*( p.x.^2 )*( 1 + tan(theta)^2 );

%Apogee (xa,ya in m, ta in s)
p.ta = u*sin(theta)/g;
p.xa = (u^2)*sin(2*theta)/(2*g);
p.ya = h + ( (u^2)/(2*g) )*sin(theta)^2;

%x,y velocities in m/s
p.vx = u*cos(theta)*ones(1,N);
p.vy = u*sin(theta) - g*p.t;

%Projectile speed /ms-1
p.v = sqrt( p.vx.^2 + p.vy.^2 );

%Velocity angle /rad anticlockwise from horizontal
p.phi = atan2( p.vy,p.vx );

%Compute length of trajectory /m
a = (u^2)/( g * (1 + ( tan(theta) )^2 ) );
b = tan(theta);
c = tan(theta) - g*p.R*( 1 + (tan(theta))^2 )/(u^2);
p.s = a * ( z_func(b) - z_func(c) );

%Trajectory length /m (numeric calculation)
dx = diff( p.x ); dy = diff( p.y );
p.s_numeric = sum( sqrt( dx.^2 + dy.^2 ) );

%Max range parabola given h,u,g
p.theta_m = asin( sqrt( 1/( 2 + 2*g*h/(u^2) ) ) );
p.T_m = (u/g)*sqrt( 2 + 2*g*h/(u^2) );
p.R_m = ((u^2)/g)*sqrt( 1 + 2*g*h/(u^2) );

%%

%Projectile trajectory length function
function y = z_func(z)
y = 0.5*log( abs( sqrt(1+z.^2) + z ) ) + 0.5*z.*sqrt( 1+ z.^2 ) ;
```






$$t_{\pm} = \frac{3u}{2g}\left(\sin\theta \pm \sqrt{\sin^2\theta - \tfrac{8}{9}}\right)$$

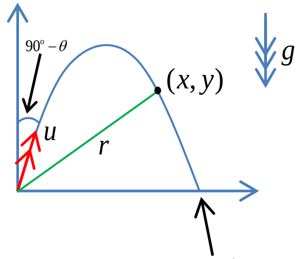
$$\theta \geq \sin^{-1}\left(\tfrac{2\sqrt{2}}{3}\right) \approx 70.5^{\circ}$$

**Challenge #7: A curious fact is that the *range* of a projectile from the launch point (let's set this to be (0,0) for convenience) plotted against time can actually pass through a local maximum and then a minimum, before increasing with increasing gradient. Use the derivations on the next slide to recreate the above graphs. Work out the times, $x$, $y$, and $r$ values for these maxima and minima and plot these via a marker such as a *.**

## Projectile range

The distance $r$ of a particle undergoing projectile motion from $(0,0)$ is given by:

$$r^2 = x^2 + y^2$$
$$y = ut\sin\theta - \tfrac{1}{2}gt^2$$
$$x = ut\cos\theta$$




$$R = \frac{u^2}{g}\sin 2\theta$$

Hence:

$$r^2 = u^2t^2\cos^2\theta + \left(ut\sin\theta - \tfrac{1}{2}gt^2\right)^2$$
$$r^2 = u^2t^2\cos^2\theta + u^2t^2\sin^2\theta - gt^2ut\sin\theta + \tfrac{1}{4}g^2t^4$$
$$r^2 = u^2t^2\left(\cos^2\theta + \sin^2\theta\right) - gt^3u\sin\theta + \tfrac{1}{4}g^2t^4$$
$$r^2 = u^2t^2 - gt^3u\sin\theta + \tfrac{1}{4}g^2t^4$$

$$\therefore \boxed{r = \sqrt{u^2t^2 - gt^3u\sin\theta + \tfrac{1}{4}g^2t^4}}$$

Is it possible to have a maximum or minimum in a graph of $r$ vs $t$ (and hence, since they are proportional) $x$? Ignore 'obvious' minimum when $t = 0$.

$$\frac{dr^2}{dt} = 2r\frac{dr}{dt} \quad \therefore \text{ if } r>0 \text{ then } \frac{dr}{dt}=0 \text{ if } \frac{dr^2}{dt}=0$$
$$r^2 = u^2t^2 - gt^3u\sin\theta + \tfrac{1}{4}g^2t^4$$
$$\therefore \frac{dr^2}{dt} = 2u^2t - 3gt^2u\sin\theta + g^2t^3$$
$$\therefore \frac{dr^2}{dt} = 0 \Rightarrow 2u^2t - 3gt^2u\sin\theta + g^2t^3 = 0$$
$$\therefore t\left(2u^2 - 3gtu\sin\theta + g^2t^2\right) = 0$$

Since $t>0$: $\quad 2u^2 - 3gtu\sin\theta + g^2t^2 = 0$

$$\therefore t^2 - \frac{3u}{g}\sin\theta\, t + \frac{2u^2}{g^2} = 0$$
$$\therefore \left(t - \frac{3u}{2g}\sin\theta\right)^2 - \frac{9u^2}{4g^2}\sin^2\theta + \frac{2u^2}{g^2} = 0$$
$$\therefore \boxed{t_\pm = \frac{3u}{2g}\sin\theta \pm \sqrt{\frac{9u^2}{4g^2}\sin^2\theta - \frac{2u^2}{g^2}}}$$

$$\therefore \boxed{t_\pm = \frac{3u}{2g}\left(\sin\theta \pm \sqrt{\sin^2\theta - \tfrac{8}{9}}\right)}$$

Real roots (i.e. there are times when the graph of $r$ vs $t$ is indeed at a maxima or minima) occur when:

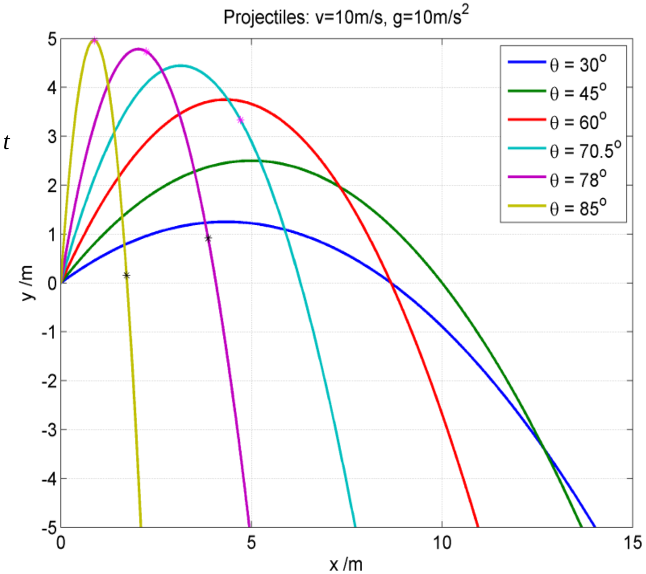
$$\sin^2\theta > \tfrac{8}{9} \quad \therefore \sin\theta > \tfrac{2\sqrt{2}}{3} \approx 70.5^\circ \quad \text{since} \quad 0 \le \theta \le 90^\circ$$

The critical angle for stationary points of $r$ vs $t$ is when the above equality holds.

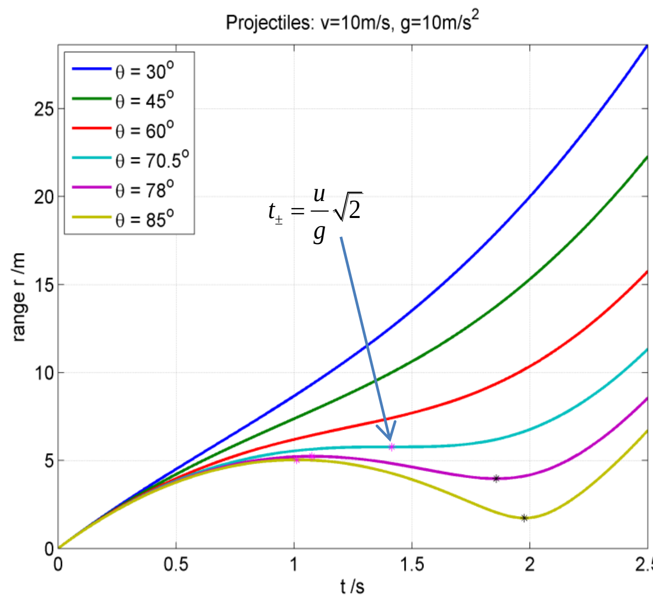
$$\sin\theta = \tfrac{2\sqrt{2}}{3} \Rightarrow \theta \approx 70.5^\circ$$
$$\therefore t_\pm = \frac{3u}{2g}\sin\theta = \frac{3u}{2g}\tfrac{2\sqrt{2}}{3}$$
$$\therefore \boxed{t_\pm = \frac{u}{g}\sqrt{2}}$$

which is a nice result, since the maximum horizontal range when $\theta = 45^\circ$ is:

$$\boxed{R_{\max} = \frac{u}{g}}$$

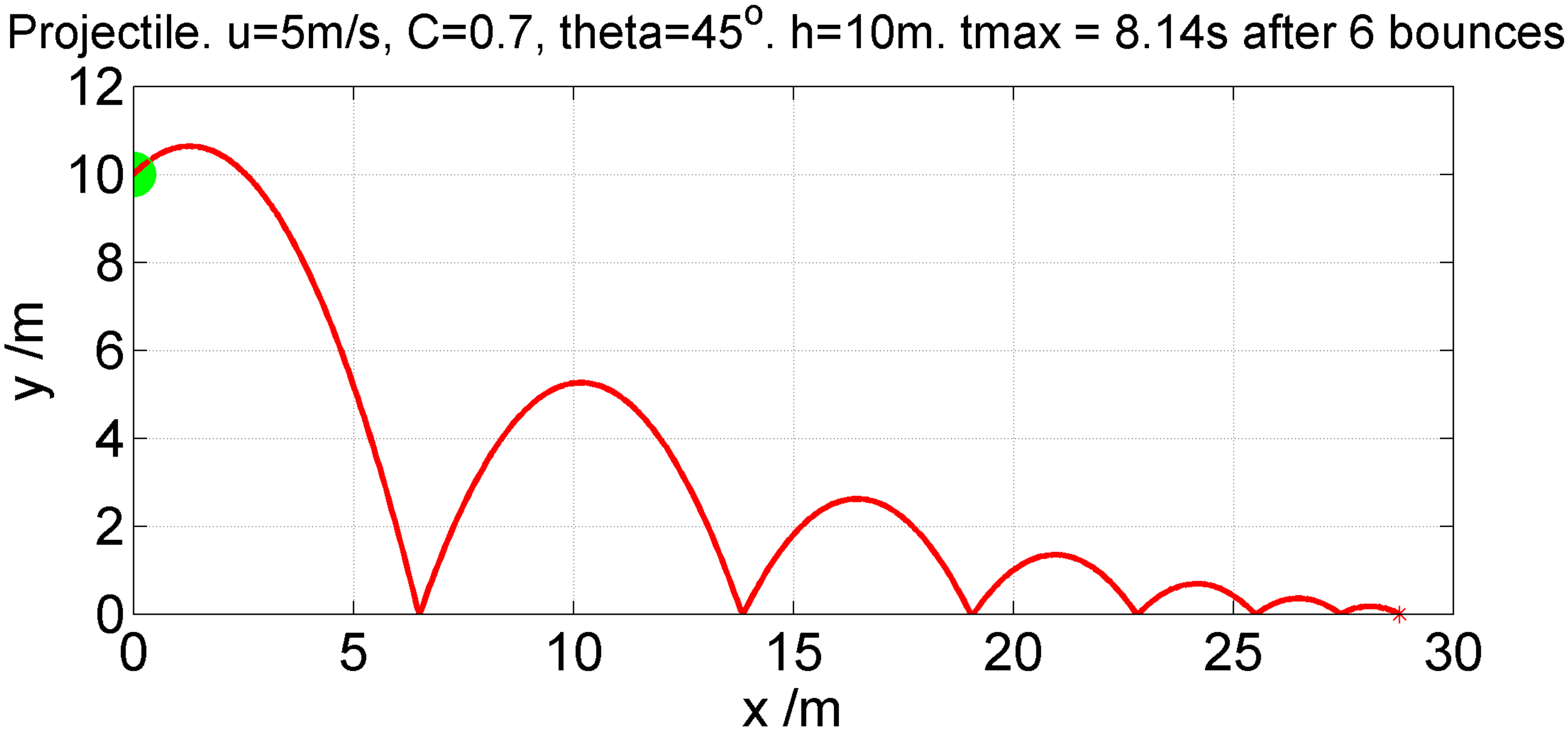



* a maxima in $r$ vs $t$     * a minima in $r$ vs $t$



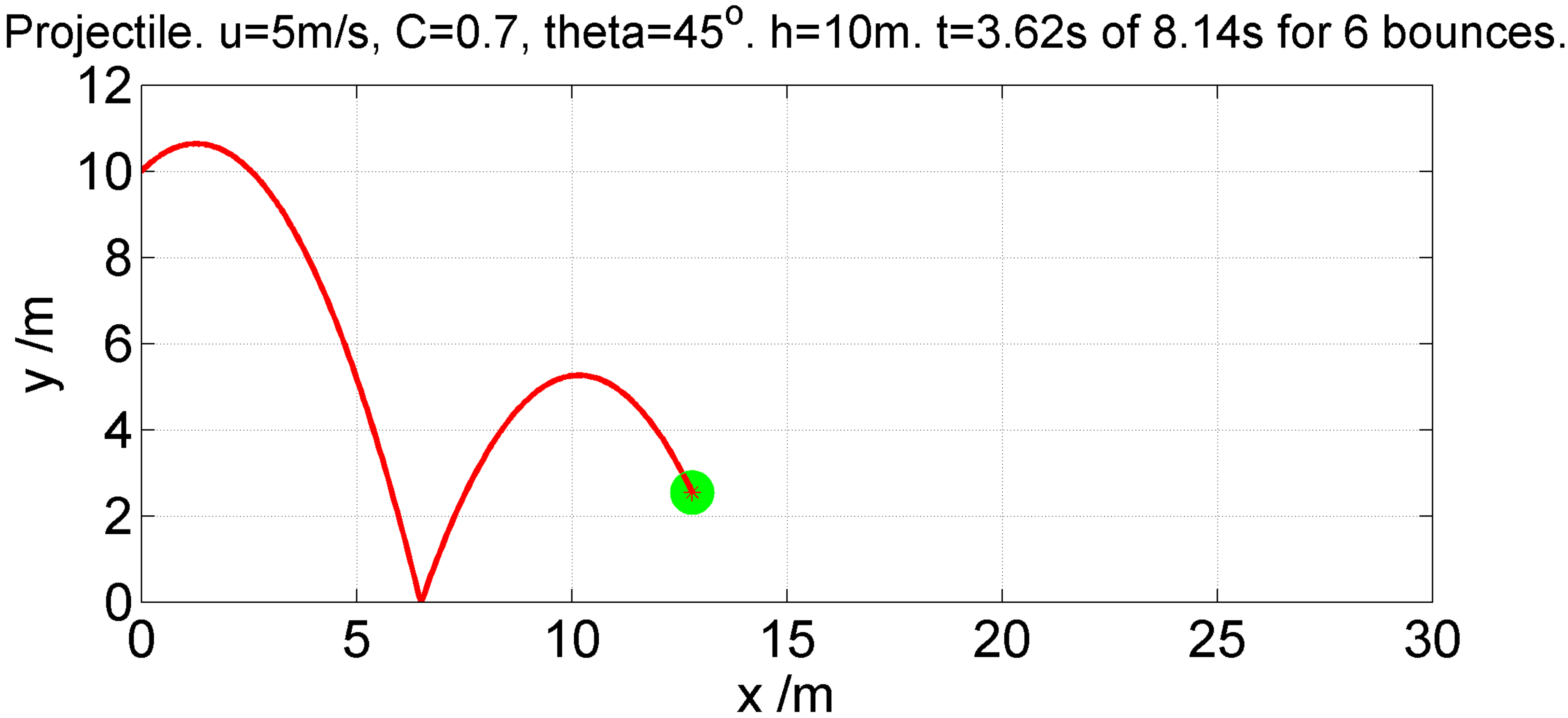

$$\boxed{t_\pm = \frac{3u}{2g}\left(\sin\theta \pm \sqrt{\sin^2\theta - \tfrac{8}{9}}\right) \qquad \theta \ge \sin^{-1}\left(\tfrac{2\sqrt{2}}{3}\right)}$$

You can clearly see a maximum and minimum in a graph of $r$ vs $t$ for elevation angles greater than 70.5°.




**Challenge #8: Use a numerical method assuming constant acceleration motion between small, discrete timesteps (e.g. the 'Verlet' method) to compute a projectile trajectory which includes the possibility of a bounce. Define the coefficient of restitution to be the vertical speed of separation / vertical speed of approach. Assume a constant horizontal speed, and stop the simulation after N bounces.**

***Extension:*** **Modify your code to *animate* the trajectory, and ideally, create a video file for efficient future playback.**

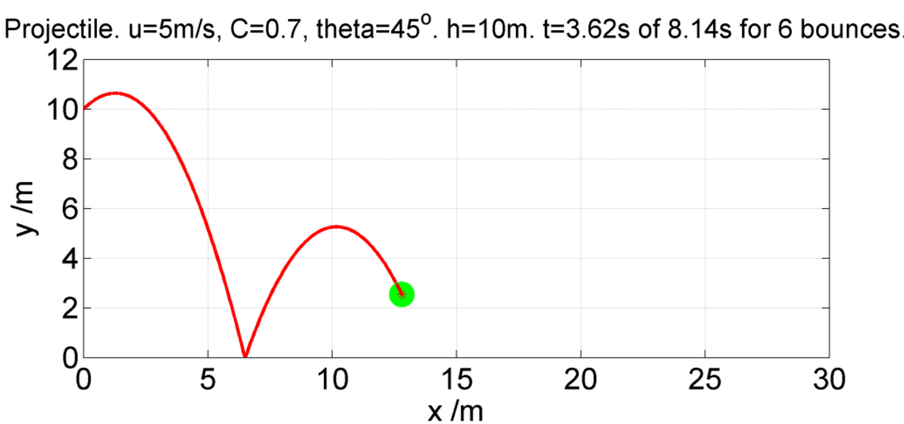



**Challenge #8 Extension*:***

**Modify your code to *animate* the trajectory, and ideally, create a video file for efficient future playback. A nice feature could be for the trajectory to be revealed as a projectile object bounces.**

```
%Verlet trajectory solver
function [t,x,y,vx,vy] = verlet_trajectory_solver(
N,C,g,dt,h,theta,u )

%Initial conditions
theta = theta*pi/180; nbounce = 0; n=1;
t = 0; x = 0; y = h; vy = u*sin(theta); vx =
u*cos(theta);
```

## MATLAB implementation of bouncing projectile using *Verlet* 'constant acceleration-between-timesteps' method




```
%Determine trajectory
while nbounce <= N

    %Acceleration
    ax = 0; ay = -g;

    %Update position
    x(n+1) = x(n) + vx(n)*dt + 0.5*ax*dt^2;
    y(n+1) = y(n) + vy(n)*dt + 0.5*ay*dt^2;

    %Update acceleration (this could involve x,y potentially)
    aax = 0; aay = -g;

    %Update velocity
    vx(n+1) = vx(n) + 0.5*( ax + aax )*dt;
    vy(n+1) = vy(n) + 0.5*( ay + aay )*dt;

    %Update time
    t(n+1) = t(n) + dt;

    %Check if ball has bounced. If so, modify vy accordingly
    if y(n+1) < 0
        y(n+1) = 0;
        vy(n+1) = -C*vy(n+1);
        nbounce = nbounce + 1;
    end

    %Increment counter
    n = n+1;
end
```

**Fixed timestep projectile motion model including air resistance**

Dr A. French. 14/7/2023

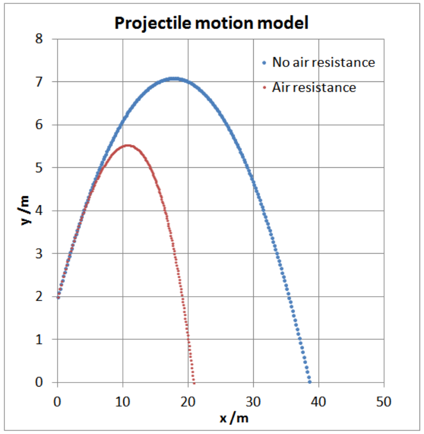
**Inputs**

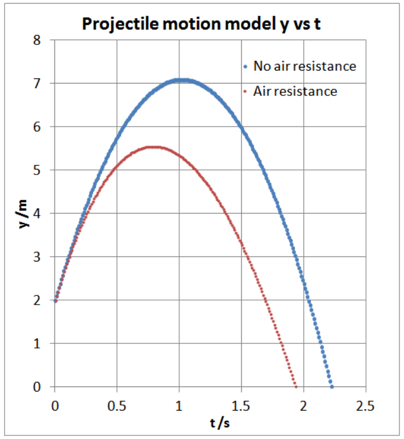
| | |
|---|---|
| launch angle (deg) | 30 |
| launch angle (rad) | 0.5236 |
| launch speed (m/s) | 20 |
| launch height (m) | 2 |
| g /ms^-2 | 9.81 |
| drag coefficient cD | 1 |
| cross sectional area (m^2) | 0.0079 |
| air density (kgm^-3) | 1 |
| object mass /kg | 0.1 |

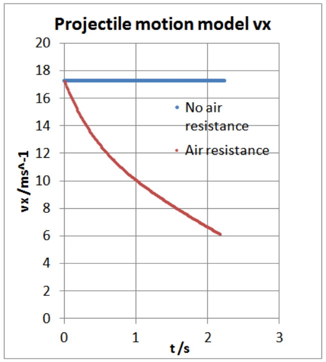
| | |
|---|---|
| air resistance factor k | 0.0393 |

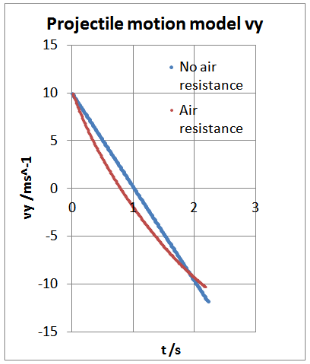
| | |
|---|---|
| Time step /s | 0.01 |

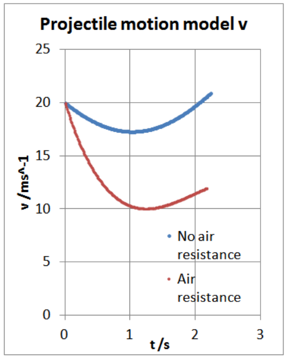
| | No air resistance model | | | | | | Air resistance model | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| t /s | vx | vy | v | x | y | | ax | ay | vx | vy | v | x | y |
| 0 | 17.321 | 10 | 20 | 0 | 2 | | -13.6 | -17.66 | 17.321 | 10 | 20 | 0 | 2 |
| 0.01 | 17.321 | 9.9019 | 19.951 | 0.1732 | 2.0995 | | -13.36 | -17.45 | 17.184 | 9.8234 | 19.794 | 0.1725 | 2.0991 |
| 0.02 | 17.321 | 9.8038 | 19.903 | 0.3464 | 2.198 | | -13.12 | -17.23 | 17.051 | 9.6489 | 19.592 | 0.3437 | 2.1965 |
| 0.03 | 17.321 | 9.7057 | 19.854 | 0.5196 | 2.2956 | | -12.89 | -17.03 | 16.92 | 9.4766 | 19.393 | 0.5136 | 2.2921 |
| 0.04 | 17.321 | 9.6076 | 19.807 | 0.6928 | 2.3922 | | -12.66 | -16.83 | 16.791 | 9.3063 | 19.197 | 0.6821 | 2.386 |
| 0.05 | 17.321 | 9.5095 | 19.759 | 0.866 | 2.4877 | | -12.44 | -16.63 | 16.664 | 9.138 | 19.005 | 0.8494 | 2.4782 |
| 0.06 | 17.321 | 9.4114 | 19.712 | 1.0392 | 2.5823 | | -12.22 | -16.44 | 16.54 | 8.9717 | 18.816 | 1.0154 | 2.5688 |
| 0.07 | 17.321 | 9.3133 | 19.666 | 1.2124 | 2.676 | | -12.01 | -16.25 | 16.418 | 8.8073 | 18.631 | 1.1802 | 2.6577 |
| 0.08 | 17.321 | 9.2152 | 19.619 | 1.3856 | 2.7686 | | -11.81 | -16.07 | 16.298 | 8.6448 | 18.448 | 1.3438 | 2.7449 |
| 0.09 | 17.321 | 9.1171 | 19.573 | 1.5588 | 2.8603 | | -11.61 | -15.9 | 16.18 | 8.4841 | 18.269 | 1.5062 | 2.8306 |
| 0.1 | 17.321 | 9.019 | 19.528 | 1.7321 | 2.951 | | -11.41 | -15.72 | 16.063 | 8.3251 | 18.093 | 1.6674 | 2.9146 |
| 0.11 | 17.321 | 8.9209 | 19.483 | 1.9053 | 3.0406 | | -11.22 | -15.56 | 15.949 | 8.1679 | 17.919 | 1.8274 | 2.9971 |
| 0.12 | 17.321 | 8.8228 | 19.438 | 2.0785 | 3.1294 | | -11.04 | -15.39 | 15.837 | 8.0123 | 17.749 | 1.9864 | 3.078 |
| 0.13 | 17.321 | 8.7247 | 19.394 | 2.2517 | 3.2171 | | -10.86 | -15.24 | 15.727 | 7.8583 | 17.581 | 2.1442 | 3.1574 |
| 0.14 | 17.321 | 8.6266 | 19.35 | 2.4249 | 3.3039 | | -10.68 | -15.08 | 15.618 | 7.706 | 17.416 | 2.3009 | 3.2352 |
| 0.15 | 17.321 | 8.5285 | 19.306 | 2.5981 | 3.3896 | | -10.51 | -14.93 | 15.511 | 7.5552 | 17.253 | 2.4566 | 3.3115 |
| 0.16 | 17.321 | 8.4304 | 19.263 | 2.7713 | 3.4744 | | -10.34 | -14.78 | 15.406 | 7.4059 | 17.094 | 2.6111 | 3.3863 |
| 0.17 | 17.321 | 8.3323 | 19.22 | 2.9445 | 3.5582 | | -10.18 | -14.64 | 15.303 | 7.2581 | 16.937 | 2.7647 | 3.4596 |
| 0.18 | 17.321 | 8.2342 | 19.178 | 3.1177 | 3.6411 | | -10.02 | -14.5 | 15.201 | 7.1117 | 16.782 | 2.9172 | 3.5315 |
| 0.19 | 17.321 | 8.1361 | 19.136 | 3.2909 | 3.7229 | | -9.862 | -14.36 | 15.101 | 6.9667 | 16.63 | 3.0687 | 3.6018 |
| 0.2 | 17.321 | 8.038 | 19.095 | 3.4641 | 3.8038 | | -9.709 | -14.23 | 15.002 | 6.8231 | 16.481 | 3.2192 | 3.6708 |
| 0.21 | 17.321 | 7.9399 | 19.054 | 3.6373 | 3.8837 | | -9.561 | -14.1 | 14.905 | 6.6809 | 16.334 | 3.3688 | 3.7383 |
| 0.22 | 17.321 | 7.8418 | 19.013 | 3.8105 | 3.9626 | | -9.415 | -13.97 | 14.809 | 6.5399 | 16.189 | 3.5173 | 3.8044 |
| 0.23 | 17.321 | 7.7437 | 18.973 | 3.9837 | 4.0405 | | -9.273 | -13.84 | 14.715 | 6.4003 | 16.047 | 3.665 | 3.8691 |
| 0.24 | 17.321 | 7.6456 | 18.933 | 4.1569 | 4.1175 | | -9.134 | -13.72 | 14.623 | 6.2618 | 15.907 | 3.8117 | 3.9324 |
| 0.25 | 17.321 | 7.5475 | 18.894 | 4.3301 | 4.1934 | | -8.999 | -13.6 | 14.531 | 6.1246 | 15.769 | 3.9574 | 3.9944 |
| 0.26 | 17.321 | 7.4494 | 18.855 | 4.5033 | 4.2684 | | -8.866 | -13.49 | 14.441 | 5.9886 | 15.634 | 4.1023 | 4.0549 |
| 0.27 | 17.321 | 7.3513 | 18.816 | 4.6765 | 4.3424 | | -8.736 | -13.37 | 14.353 | 5.8537 | 15.5 | 4.2463 | 4.1141 |
| 0.28 | 17.321 | 7.2532 | 18.778 | 4.8497 | 4.4154 | | -8.61 | -13.26 | 14.265 | 5.72 | 15.369 | 4.3893 | 4.172 |
| 0.29 | 17.321 | 7.1551 | 18.74 | 5.0229 | 4.4875 | | -8.486 | -13.15 | 14.179 | 5.5874 | 15.24 | 4.5316 | 4.2285 |
| 0.3 | 17.321 | 7.057 | 18.703 | 5.1962 | 4.5586 | | -8.365 | -13.05 | 14.094 | 5.4558 | 15.113 | 4.6729 | 4.2838 |
| 0.31 | 17.321 | 6.9589 | 18.666 | 5.3694 | 4.6286 | | -8.247 | -12.94 | 14.011 | 5.3253 | 14.989 | 4.8135 | 4.3377 |
| 0.32 | 17.321 | 6.8608 | 18.63 | 5.5426 | 4.6977 | | -8.131 | -12.84 | 13.928 | 5.1959 | 14.866 | 4.9532 | 4.3903 |
| 0.33 | 17.321 | 6.7627 | 18.594 | 5.7158 | 4.7658 | | -8.018 | -12.74 | 13.847 | 5.0675 | 14.745 | 5.092 | 4.4416 |
| 0.34 | 17.321 | 6.6646 | 18.558 | 5.889 | 4.833 | | -7.907 | -12.65 | 13.767 | 4.94 | 14.626 | 5.2301 | 4.4916 |
| 0.35 | 17.321 | 6.5665 | 18.523 | 6.0622 | 4.8991 | | -7.799 | -12.55 | 13.688 | 4.8135 | 14.509 | 5.3674 | 4.5404 |
| 0.36 | 17.321 | 6.4684 | 18.489 | 6.2354 | 4.9643 | | -7.693 | -12.46 | 13.61 | 4.688 | 14.394 | 5.5039 | 4.5879 |



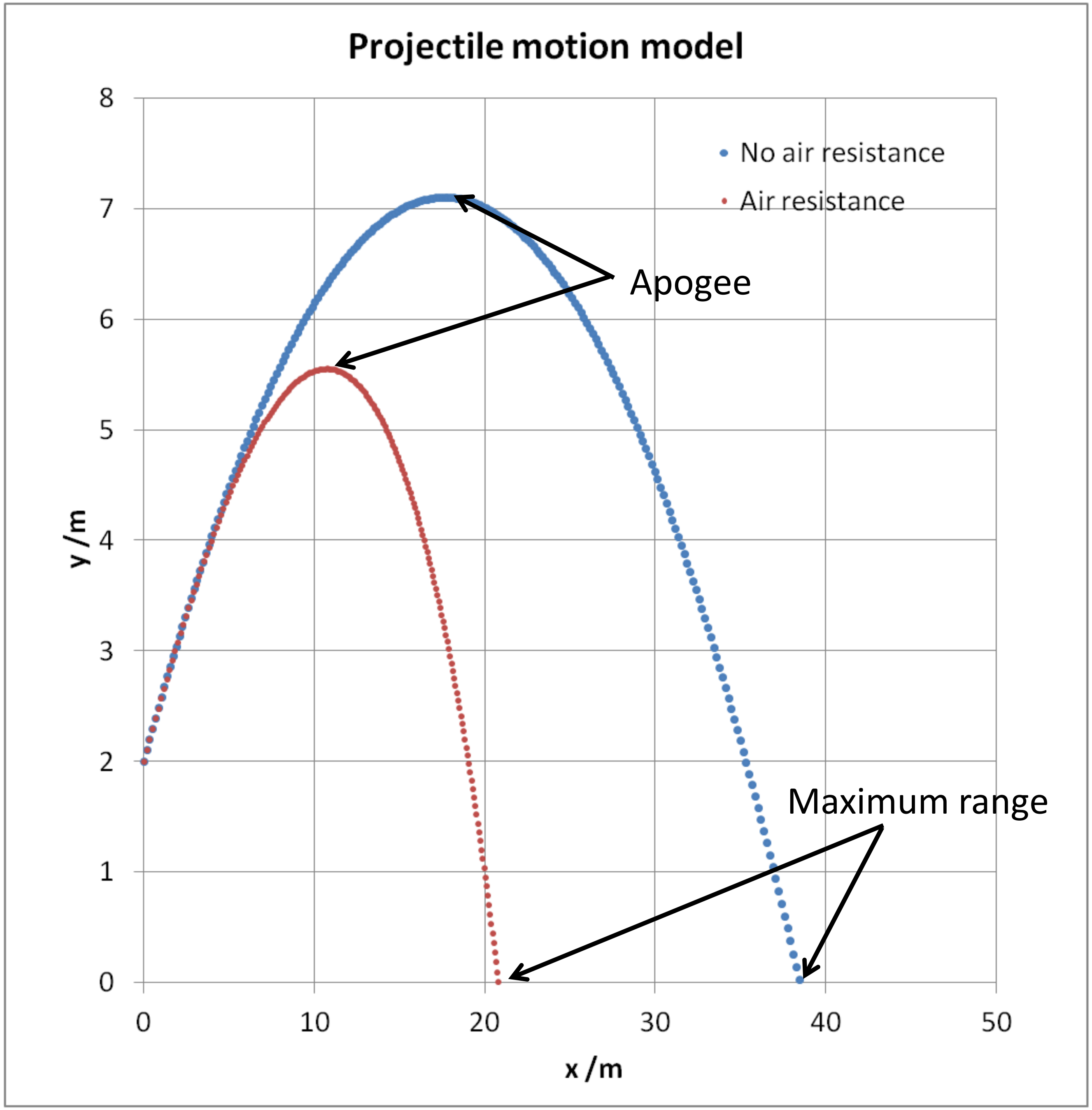



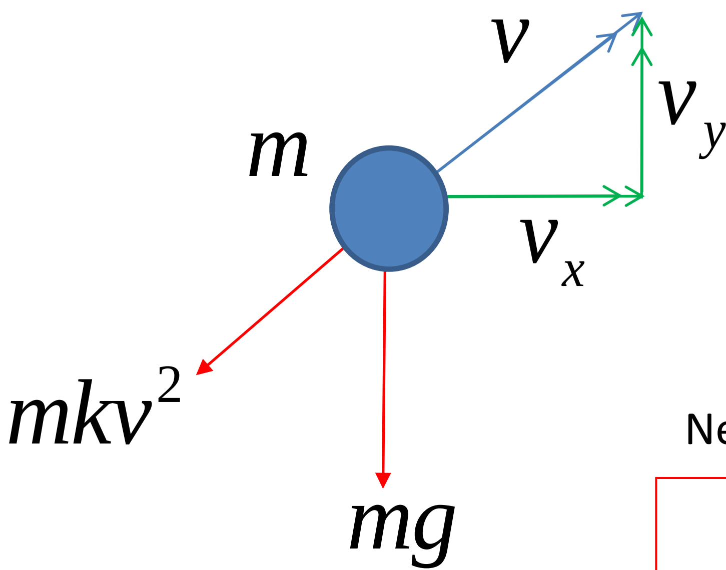

**Challenge #9: Write a new projectile model which compares a drag-free model (use what you have already done in previous challenges) with a model incorporating the effect of air resistance. Use a Verlet method to solve the air-resistance case. It is possible to solve motion under drag which varies with the square of velocity analytically in 1D (see here) but in 2D projectile motion drag always opposes the velocity vector – which makes the maths much harder. So write a numerical recipe! Mathematical details in the next few slides.**

| Inputs | |
|---|---|
| launch angle /deg | 30 |
| launch speed /ms^-1 | 20 |
| launch height /m | 2 |
| g /ms^-2 | 9.81 |
| drag coefficient cD | 0.1 |
| cross sectional area /m^2 | 0.007854 |
| air density /kgm^-3 | 1 |
| object mass /kg | 0.1 |
| air resistance factor k | 0.003927 |
| Time step /s | 0.01 |

Investigate the effect of air resistance using the model.



**Model which incorporates air resistance**




Cross sectional area

$$k = \frac{\frac{1}{2} c_D \rho A}{m}$$

Drag coefficient

Mass

Air density

Newton II

$$x: \qquad ma_x = -\frac{v_x}{v} mkv^2$$

$$y: \qquad ma_y = -mg - \frac{v_y}{v} mkv^2$$

Air resistance always *opposes* the direction of velocity

**Model which incorporates air resistance**

$$a_x = -\frac{v_x}{v} k v^2$$

$$a_y = -g - \frac{v_y}{v} k v^2$$

x and y accelerations

Cross sectional area

$$k = \frac{\frac{1}{2} c_D \rho A}{m}$$

Drag coefficient

Mass

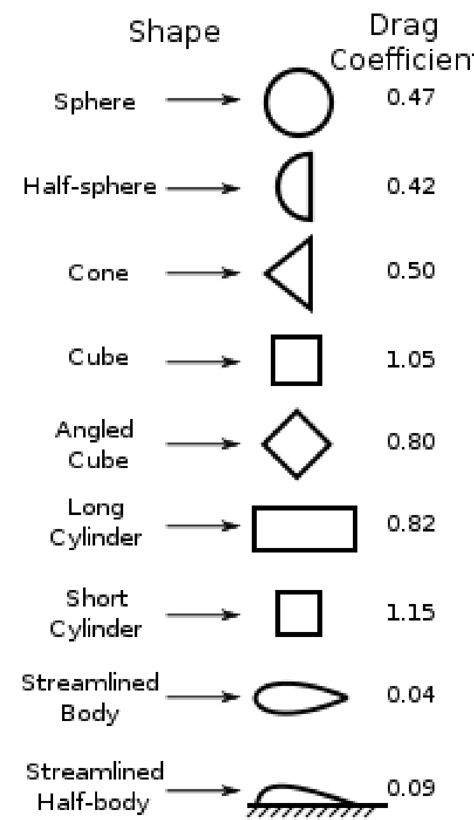
Air density

$$\frac{\Delta v_x}{\Delta t} = a_x, \quad \frac{\Delta v_y}{\Delta t} = a_y$$

$x$ and $y$ accelerations

$$\frac{\Delta x}{\Delta t} = v_x, \quad \frac{\Delta y}{\Delta t} = v_y$$

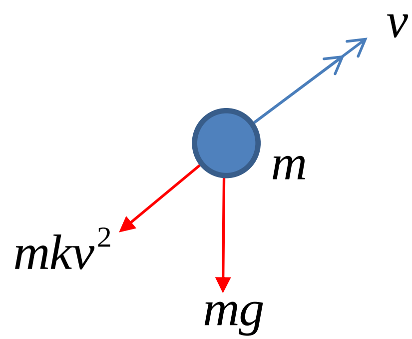
$x$ and $y$ velocities



Measured Drag Coefficients

For *no* air resistance: $a_x = 0 \quad a_y = -g$

## Model which incorporates air resistance




$$k = \frac{\frac{1}{2}c_D \rho A}{m}$$

Air resistance factor

$$t = 0$$
$$u_x = u\cos\theta$$
$$u_y = u\cos\theta$$
$$x = 0$$
$$y = h$$

Initial conditions

$$t_{n+1} = t_n + \Delta t$$

Finite time step (e.g. 0.01s)

$$a_x = -\frac{v_x}{v}kv^2$$

$x$ Acceleration

$$a_y = -g - \frac{v_y}{v}kv^2$$

$y$ Acceleration

i.e. how $x$,$y$, $v_x$, $v_y$ change between time steps

$$x_{n+1} = x_n + v_x \Delta t + \tfrac{1}{2}a_x \Delta t^2$$
$$y_{n+1} = y_n + v_y \Delta t + \tfrac{1}{2}a_y \Delta t^2$$

Constant acceleration motion between the time steps (a "Verlet" method)

$$v_x^{(n+1)} = v_x^{(n)} + a_x \Delta t$$
$$v_y^{(n+1)} = v_y^{(n)} + a_y \Delta t$$
$$v = \sqrt{v_x^2 + v_y^2}$$

## Extension opportunities:

• Consider projectile motion in an atmosphere with a model of air density that diminishes with altitude. See the **2022 BPhO Computational Challenge** for details!

• Consider projectile motion launched from a spherical planet, which rotates about a fixed axis. Work out the latitude and longitude where the projectile lands, and animate the motion. Texture-map a planet surface e.g. Earth, Mars, the Moon....

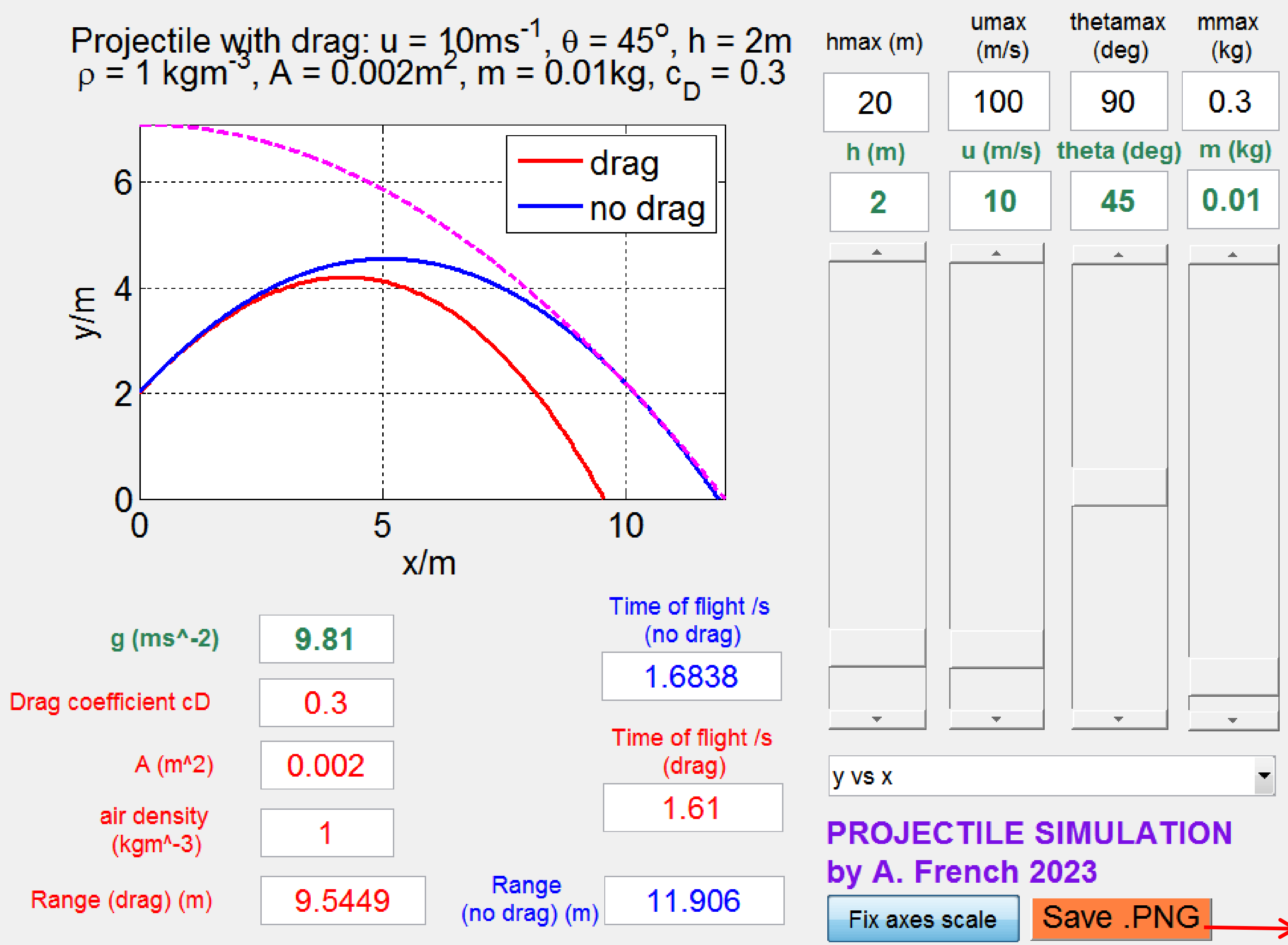
•Write a **graphical user interface** (GUI) for the projectile model and encode this as an 'app'. Coding up an iOS/Android smartphone app will particularly impress the judges.

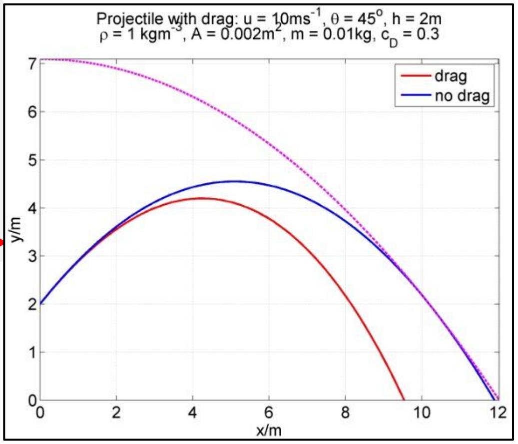
• Write up your model as a **short paper.** (Aim for about 10 sides of A4, two columns). If you have never written a paper before, download a few from the *Physics Education* journal. *The Epidemiology of Eyam* might be a good start... A good opportunity to learn LaTeX – which is the typesetting language used to write most technical papers and books in the physical sciences. Including *Science by Simulation* *

Don't forget to include any extension projects in your video, as this is the only way you will gain credit for your work in the BPhO Computational Challenge.
I'm afraid we cannot accept any other files. **Submit only the YouTube link to your two-minute screencast.**

* *ScibySim* was created in Scientific Word. There are lots of other LaTeX-based tools available. Find one that works for you!

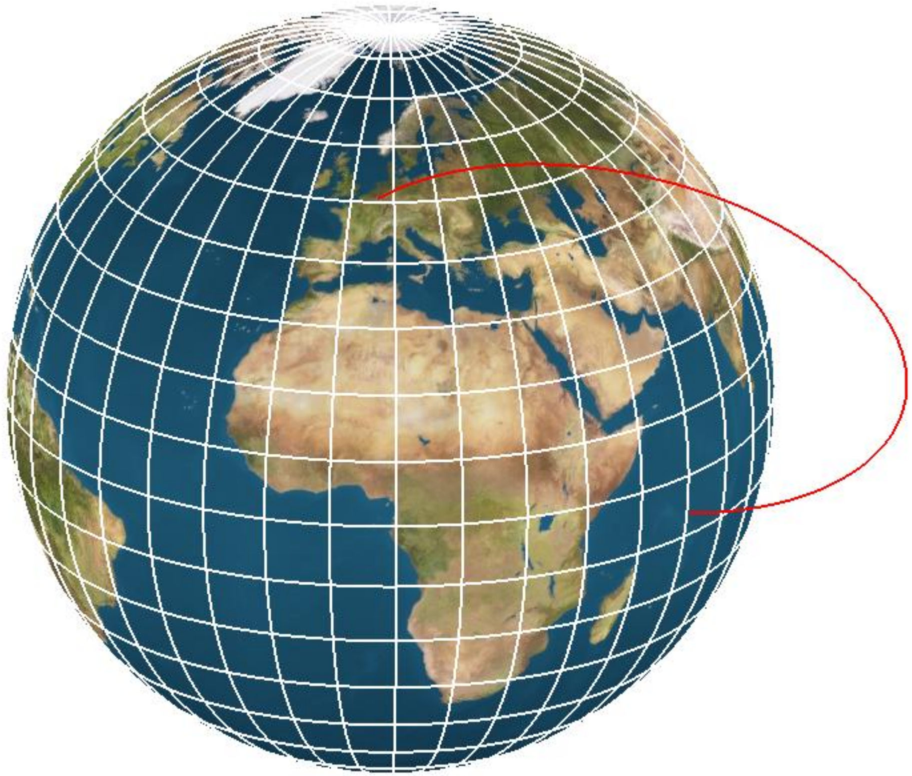
**Extension idea**




Example of a projectile motion model implemented in a **Graphical User Interface (GUI).** The sliders change the $h, u, \theta$ and particle mass $m$ inputs, and the trajectory curves are automatically updated. This simulation will also output high quality PNG files for incorporation into reports etc.

Extension idea



**Newton's law of Universal Gravitation**

$$g = \frac{GM}{r^2}$$

Planet mass

Distance from centre of planet

$$G = 6.67 \times 10^{-11}\,\mathrm{m}^3\mathrm{kg}^{-1}\mathrm{s}^{-2}$$

i.e. you will need to modify the strength of gravity as the projectile gains altitude.

Note also the planet will *rotate under the projectile.* Don't forget the velocity of the rotating planetary surface at launch.

**TOP TIP:** Use a 3D *x,y,z* coordinate system with origin of the centre of the earth. Work out the projectile trajectory AND the (rotating) coordinates of the planet surface.

Consider projectile motion launched from a spherical planet, which *rotates about a fixed axis*. Work out the latitude and longitude where the projectile lands, and animate the motion. Texture-map a planet surface e.g. Earth, Mars, the Moon....